



**CENATAV**

Centro de Aplicaciones de  
Tecnologías de Avanzada  
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2142  
ISSN 2072-6287  
Versión Digital

REPORTE TÉCNICO  
**Reconocimiento  
de Patrones**

SERIE AZUL

**Combinación de agrupamientos:  
un estado del arte**

Lic. Sandro Vega-Pons,  
Dr. C. José Ruiz-Shulcloper

**RT\_029**

**enero 2010**





**CENATAV**

Centro de Aplicaciones de  
Tecnologías de Avanzada  
MINISTERIO DE LA INDUSTRIA BÁSICA

RNPS No. 2142  
ISSN 2072-6287  
Versión Digital

**SERIE AZUL**

REPORTE TÉCNICO  
**Reconocimiento  
de Patrones**

**Combinación de agrupamientos:  
un estado del arte**

Lic. Sandro Vega-Pons,  
Dr. C. José Ruiz-Shulcloper

**RT\_029**

**enero 2010**



# Combinación de agrupamientos: un estado del arte

Lic. Sandro Vega-Pons, Dr. C. José Ruiz-Shulcloper

Centro de Aplicaciones de Tecnología de Avanzada, 7a #21812 e/ 218 y 222, Siboney, Playa, Ciudad de La Habana, Cuba

[svega@cenatav.co.cu](mailto:svega@cenatav.co.cu)

RT\_029 CENATAV

Fecha del camera ready: 27 de noviembre de 2009

**Resumen:** En este trabajo exponemos los principales esquemas de combinación de los resultados de clasificadores no supervisados que se han empleado con el propósito de lograr estructuraciones de “mejor calidad”. El problema de mejorar los resultados de algoritmos de clasificación no supervisados y el sentido que tienen tales mejoras son analizados en este trabajo. Aquí no se pretende hacer un análisis exhaustivo de otras problemáticas muy relacionadas con los esquemas de combinación de resultados como son por ejemplo, los índices de validación de agrupamientos y las formas de medir la similaridad entre agrupamientos. Sin embargo, nos es imposible no referirnos también a estas problemáticas y abordarlas según nuestras concepciones. Los esquemas de combinación de resultados (*clustering ensembles* en inglés) son analizados a partir de una descripción general de sus componentes fundamentales: los mecanismos de generación y las funciones de consenso. En este trabajo se expone también uno de los resultados recientes de los autores en esta temática.

**Palabras clave:** clasificación no supervisada, combinación de agrupamientos, función de similaridad, función de consenso, índices de validación

**Abstract:** In this work we present the main cluster ensemble approaches that have been used in order to achieve “better” organization of the data. The problem of improving the clustering results and the meaning of these improvements are analyzed in this work. We do not attempt to do an in-depth analysis of other problems very related with the cluster ensemble techniques like: the similarity measures between partitions and the clustering validity indexes. However, it is impossible not to refer to these problems and get on them according to our conceptions. The different clustering ensemble methods are analyzed doing a general description of its fundamental components: Generation's mechanisms and the consensus functions. In this work, we also present one of our recent results on this subject.

**Keywords:** Cluster Ensemble, Cluster Analysis, Similarity Measure, Consensus Function, validity index.

## 1 Introducción

La clasificación no supervisada, estructuración o agrupamientos (cluster analysis) es una técnica esencial en cualquier campo de investigación que involucre el análisis o procesamiento de datos, tales como la taxonomía, la minería de datos, la recuperación de documentos, la segmentación de imágenes, etc. Su objetivo fundamental es encontrar la estructuración *subyacente* de un conjunto de datos a partir fundamentalmente de ciertos criterios de agrupamientos, propiedades de los datos y formas de comparación de los mismos [1] ¿Qué es una estructuración subyacente, intrínseca, original? Estos son cuestiones medulares en esta problemática que en la actualidad, según nuestros criterios, no han recibido respuestas adecuadas.

En la bibliografía existe una gran variedad de familias (modelos) de algoritmos de agrupamiento dirigidas a la solución de este problema, entre los que se encuentran: k-Means [2], EM [3], Mean Shift [4], algoritmos basados en teoría espectral de grafos [5, 6], estructuración de

datos mezclados e incompletos [7, 8], estructuraciones difusas como Fuzzy c-Means [9], etc. (ver resumen en [10]). Sin embargo, como es conocido, no existe ningún clasificador no supervisado capaz de trabajar *correctamente* con cualquier tipo de datos y obtener los *mejores* resultados posibles, ya que todos estos modelos de agrupamiento imponen una organización sobre los datos, la cual, puede estar muy distante de la *óptima* para un problema en particular. ¿Qué significa y cómo se obtiene una *estructuración óptima*? Este es también un complejo problema. Toda esta diversidad y complejidad de estos problemas obedecen en gran medida a la naturaleza de los datos y del problema, a los diferentes tamaños, densidades y formas en que se pueden presentar las estructuras internas de los datos (clusters) para cada problema y a la propia naturaleza no supervisada del proceso. Por tanto, para un problema en particular, diferentes soluciones, obtenidas mediante diferentes clasificadores no supervisados, pueden ser igualmente plausibles si no se posee ningún conocimiento previo acerca de los datos, de su distribución subyacente, de las formas de comparación entre los mismos, etc. Cada modelo de algoritmos implícitamente o explícitamente asume propiedades y naturaleza de los datos así como cierta estructuración de los mismos y por lo tanto puede producir resultados erróneos o sin valor alguno, cuando estas asunciones no son satisfechas. Por ejemplo, la familia de algoritmos k-Means, por su forma de construir los agrupamientos, producirá estructuraciones que en el espacio n-dimensional son necesariamente hiperesferas (algunas modificaciones se han realizado para lograr hiperelipsoides de revolución). Por lo que el empleo de clasificadores de esta familia dará como resultado agrupamientos que tendrán, a priori, esa forma. Lo que puede ser o no conveniente para la solución de un problema práctico concreto (aquellos en los que realmente lo conveniente, lo que subyace, es un conjunto de hiperesferas).

Como pudo apreciarse en los párrafos anteriores, en nuestro trabajo estamos diferenciando, para lograr una mayor precisión en nuestro lenguaje, entre familia o modelo de algoritmos, algoritmos, y clasificadores.

¿Qué es un modelo de algoritmos de agrupamientos?

Por modelo de algoritmos de agrupamientos estamos entendiendo una descripción general de un proceso de estructuración de espacios de representación de los objetos sujetos a estudio, que depende de parámetros generales. Por ejemplo, la familia de algoritmos k-Means, tiene implícita el parámetro k (cantidad de agrupamientos a formar), la función distancia para la comparación de las representaciones de los objetos en dicho espacio, un funcional de optimización, que apoya la decisión de detener el proceso iterativo que rige el proceder de esta familia de algoritmos, una manera de seleccionar los centroides y conformar los agrupamientos. A partir de esta familia, podemos obtener diferentes algoritmos de agrupamientos en la medida que dichos parámetros se fijan.

¿Qué es un algoritmo de agrupamientos?

En esta familia de algoritmos, para cada valor de k tenemos un algoritmo cuyo funcionamiento es análogo al del resto de la familia pero que tiene especificado algunos de sus parámetros. En particular, fijar la cantidad de agrupamientos a obtener, la función distancia a emplear produce, por ejemplo, el algoritmo 3-Means. Una vez predeterminado el algoritmo a emplear podemos obtener diferentes clasificadores no supervisados si por ejemplo variamos la función distancia.

¿Qué es un clasificador no supervisado?

Es una instancia particular de un algoritmo al que no resta por definirle parámetro alguno. Cualquier variación de los parámetros de un algoritmo produce otro clasificador del mismo algoritmo y por supuesto de la misma familia. Por ejemplo, el 3-Means, con una distancia euclidiana, es un clasificador diferente al 3-Means con la distancia de Mahalanobis y ambos

proviene del mismo algoritmo y la misma familia y por supuesto pueden producir estructuraciones diferentes en el mismo conjunto de representaciones de objetos.

Usualmente, para la solución de un problema particular, no se sabe qué modelo de algoritmos de agrupamiento escoger, y dentro de un modelo cuál algoritmo y cuál clasificador será el adecuado seleccionar con el objetivo de identificar correctamente la estructura subyacente o producir una que posea un sentido adecuado para el área de conocimientos en particular. Es aquí donde, como solución alternativa o complementaria a escoger algún clasificador no supervisado en particular, se utiliza la idea de combinar soluciones provenientes de diferentes clasificadores (cluster ensemble) para obtener una estructuración final (de consenso). La motivación de utilizar esta filosofía de combinación tiene como precedente el éxito obtenido por la combinación de clasificadores supervisados (combining or fusion classifiers también classifier ensemble). Su extensión a este campo no ha resultado nada trivial y presenta complejidades no encontradas en su antecesor supervisado. Por otra parte, la casi totalidad de las estructuraciones a las que se hace referencia en la literatura son particiones (duras en su mayoría [11, 12] aunque hay resultados también con algunos tipos de particiones difusas [13, 14]).

El propósito que se persigue con esta idea es que al combinar los resultados de diferentes clasificadores no supervisados se mejore la calidad de los resultados alcanzados por clasificadores individuales, ya que esa fusión puede, en principio, compensar posibles errores cometidos por un clasificador simple. Es por esto, que de manera general, la estructuración de consenso debe ser más robusta y más ajustada al problema en particular, que la obtenida por algún clasificador individual. Los términos robusto y ajustado serán objeto de análisis en este trabajo.

Obsérvese cómo incluso en el propio planteamiento del problema subyacen conceptos que deben ser esclarecidos previamente para enmarcar adecuadamente las soluciones alcanzadas y conocer lo que aún falta por hacer. Debemos señalar que en la literatura actual muchos de estos conceptos no son analizados, en algunos casos se asumen como conocidos, como si existiera un consenso al respecto. Es criterio de los autores de este trabajo que ese pretendido conocimiento tácito no existe.

Por lo antes expuesto, en este trabajo mostraremos de manera muy sucinta las bases conceptuales sobre las que descansan estos problemas y plantaremos los presupuestos en los que nos basamos para enfocar la solución de los mismos.

En el segundo epígrafe comenzaremos con el concepto de espacio de representación de los objetos. En ese epígrafe analizaremos las diferentes formas de representar los objetos en problemas de reconocimiento de patrones y las propiedades de esos espacios de representación como condicionante fundamental en la selección y aplicación de los distintos modelos de clasificadores. En los epígrafes 3 y 4 discutiremos los conceptos fundamentales de la clasificación no supervisada (unsupervised classification, clustering analysis) y el importante problema de los índices de validación como instrumento para medir la calidad de los resultados alcanzados por dichos clasificadores. Un concepto medular en reconocimiento de patrones y que en la problemática de la fusión de los resultados de los clasificadores desempeña un papel esencial es el de función de similitud y en particular la forma de medir el parecido entre diferentes estructuraciones de datos. A esto dedicaremos el epígrafe 5.

Finalmente, a partir de un esquema general de la forma en que pueden realizarse la fusión de los resultados de clasificadores no supervisados, expondremos los principales esquemas desarrollados hasta la actualidad e incluiremos un resultado alcanzado por los autores en este contexto. Sobre la base de este análisis, en las conclusiones de este trabajo expondremos las principales líneas de trabajo futuro a las que consideramos es importante dedicar nuestros esfuerzos.

## 2 Espacio de representación

El problema de la representación de los objetos es un asunto esencial para el reconocimiento de patrones [15]. En esta disciplina existen dos formas fundamentales de representar los objetos: a partir de un alfabeto de símbolos, que constituyen partes primitivas del objeto, típico del enfoque sintáctico estructural; o en términos de un conjunto de valores de variables (características, propiedades, rasgos), característico de los enfoques estadístico y lógico combinatorio. Este proceso es el encargado de codificar los objetos de la vida real mediante cierta descripción matemática, de manera que se puedan realizar comparaciones e incluso operaciones entre las representaciones de dichos objetos.

Formalmente esto puede expresarse de la siguiente manera. Sea  $O = \{O_1, O_2, \dots, O_n\}$  un conjunto de  $n$  objetos,  $R = \{r_1, r_2, \dots, r_l\}$  un conjunto de  $l$  rasgos en términos de los cuales serán representados los objetos. Entenderemos por una representación del objeto  $i$ -ésimo  $O_i$  el  $l$ -uplo  $X_i = (r_1(O_i), \dots, r_l(O_i)) = (x_{i1}, \dots, x_{il}) \in \Omega = \Omega_1 \times \dots \times \Omega_l$  donde  $r_j(O_i) = x_{ij} \in \Omega_j$  es el valor que toma el rasgo  $r_j$  en el objeto  $O_i$ . El conjunto  $\Omega_j$  es el conjunto de valores admisibles del rasgo  $r_j$  (dominio de definición de la variable).

Denotaremos  $X = \{X_1, X_2, \dots, X_n\}$  como una representación del conjunto de objetos  $O$ , donde  $X_i$  es una representación de  $O_i$ . Se puede notar que es posible tener diferentes representaciones de un mismo objeto, las cuales pueden responder a espacios de representación totalmente diferentes.

El tipo de espacio de representación de los objetos va a estar determinado por el tipo de rasgo que se emplee en la descripción de los objetos, el conjunto de sus valores, los criterios de comparación de dichos valores y propiedades de los mismos en ese espacio. En muchos casos, todos estos rasgos toman valores en el conjunto de los números reales, quedando la representación de cada objeto, como un vector de  $\mathbb{R}^l$ , lo que puede permitir la utilización de una gran variedad de herramientas matemáticas definidas sobre los espacios vectoriales. Sin embargo, en ocasiones estas herramientas son aplicadas sin una interpretación adecuada en el contexto del problema real. En otros casos, todas las variables son nominales, pudiendo tomar valores en un conjunto de valores veritativos de una cierta lógica (booleana, polivalente) donde la representación de los objetos es una tupla  $X_i \in \Omega_1 \times \Omega_2 \times \dots \times \Omega_l$ , siendo en este caso, por ejemplo,  $\Omega_i = \{0,1\}$ . Sobre este tipo de datos han sido desarrolladas familias de algoritmos de agrupamiento, por ejemplo [16-18]. Sin embargo, en ocasiones, una buena modelación del problema a resolver, obliga a usar variables numéricas mezcladas con variables no numéricas (nominales, ordinales), e incluso se hace necesario el empleo de un símbolo especial para denotar la ausencia de información (missing value). En este caso la representación de los objetos es solamente una tupla de tamaño  $l$  donde ninguna estructura algebraica, topológica o lógica es asumida sobre los datos; es decir, sobre el espacio de representación de los objetos solamente se asume que es un producto cartesiano de dimensión  $l$  donde es conocido el dominio de definición de cada rasgo [8, 19].

Los algoritmos de agrupamiento son aplicados sobre alguna representación de los datos, sin embargo, teniendo un conjunto de objetos sobre cierto espacio de representación, solamente sobre éstos pueden ser aplicados algoritmos de estructuración que cumplan con las propiedades del espacio (por ejemplo, sobre un espacio que solamente se conoce que es un producto cartesiano no se deben aplicar algoritmos de estructuración que exijan la existencia de una distancia entre los objetos lo que requeriría de una estructura de espacio métrico).

Como ya habíamos mencionado es posible tener varias representaciones de un mismo objeto en diferentes espacios de representación, lo cual, en ocasiones permite abordar un problema

utilizando herramientas matemáticas muy diferentes (por ejemplo, un problema de segmentación de imágenes puede ser abordado usando la teoría de grafos si cada imagen es representada mediante un grafo, o puede ser tratado mediante métodos estadísticos si las imágenes son representadas mediante la matriz de los valores de sus píxeles). Sin embargo, en una buena modelación de un problema se debe buscar la representación de los objetos que mejor caracterice la naturaleza de los datos, su semántica en el contexto del problema real y no buscar un espacio de representación que resulte cómodo para trabajar (por ejemplo  $\mathbb{R}^1$ ) simplificando la estructura de los objetos. En el epígrafe 6.1 veremos cómo la utilización de diferentes representaciones de los objetos es una alternativa para la generación de una combinación de agrupamientos.

### 3 Clasificación no supervisada

La clasificación no supervisada es el proceso que permite establecer una relación entre las descripciones de los objetos, con el propósito de agruparlos de modo tal que cada objeto está en un agrupamiento porque el criterio basado en relaciones de similaridad que se emplee para agruparlos se satisface en un grado mayor con objetos de ese agrupamiento que con objetos de cualquier otro agrupamiento. Esta estructuración puede realizarse de dos modos diferentes: preestableciendo el número de agrupamientos (clusters) a formar (*clasificación no supervisada restringida*) o siendo la determinación de este número uno de los problemas a resolver (*clasificación no supervisada libre*).

Aunque no existe una definición mundialmente aceptada para este concepto [20-22], la mayoría de los investigadores plantean que su objetivo es encontrar una estructuración de los datos de manera que los objetos en una misma agrupación sean más parecidos entre sí, que con los que se encuentran en otras agrupaciones [23, 24].

Existen varios puntos de vista para examinar las distintas técnicas de agrupamiento (ver taxonomía en [10]), particularmente éstos pueden ser particionales o jerárquicos. Dado un conjunto de objetos  $O = \{O_1, O_2, \dots, O_n\}$  y una representación de los mismos  $X = \{x_1, x_2, \dots, x_n\}$

- Un algoritmo de agrupamiento particional busca una partición de  $X$ ,  $P = \{C_1, C_2, \dots, C_k\}$ , con  $k \leq n$ , donde cada  $C_i$  es un agrupamiento de la partición, de manera que:
  1.  $C_i \neq \emptyset, i = 1, \dots, k$ ;
  2.  $\bigcup_{i=1}^k C_i = X$ ;
  3.  $C_i \cap C_j = \emptyset, i, j = 1, \dots, k$  con  $i \neq j$ ;
  4. No existe  $j \in \{1, \dots, k\} C_j \subseteq \bigcup_{i \neq j} C_i$
- Un algoritmo de agrupamiento jerárquico tiene como objetivo construir una estructura anidada de estructuraciones  $H = \{H_1, \dots, H_q\}$  ( $q < n$ ), tal que  $C_i \in H_t, C_j \in H_l$  con  $t > l$  implica que  $C_i \subseteq C_j$  o  $C_i \cap C_j = \emptyset$  para todo  $i, j, j \neq i, t, l = 1, \dots, q$ .

En la Fig. 1 se muestra la interrelación existente entre la representación de los objetos, los algoritmos de agrupamiento y los índices de validación en el proceso de la clasificación no supervisada de un conjunto de objetos. Una representación apropiada de los datos, la selección de un algoritmo de agrupamiento capaz de obtener una estructuración de los objetos que cumpla, en cierto grado, con un conjunto de requisitos necesarios para un problema (medidos a partir de uno o más índices de validación) debe aportar la información o conocimiento necesario para la solución del problema en cuestión. Sin embargo, sin un conocimiento previo de la estructura de los objetos es prácticamente imposible saber qué algoritmo utilizar, que parámetros y con qué

tipo de índice de validación verificar los resultados para obtener información realmente relevante que ayude a dar una solución correcta a un problema real.

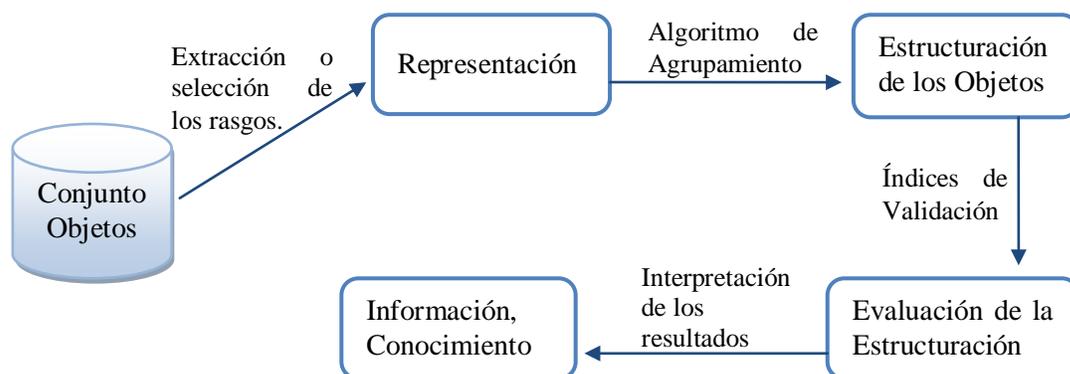


Fig. 1. Proceso de clasificación no supervisada de un conjunto de objetos

La propia formulación del problema de la clasificación no supervisada, en el intento de agrupar conjuntamente los objetos que sean *similares*, necesita una definición de medida de similaridad (o disimilaridad), entre objetos. Intuitivamente las similaridades son medidas aplicables a pares de objetos, de tal manera que objetos muy parecidos tomen grandes valores y objetos diferentes tomen valores pequeños. En este trabajo no se hará un análisis profundo del concepto y propiedades que debe cumplir una medida de similaridad (no existe un consenso mundial acerca de este concepto), sin embargo en el epígrafe 5 se dará una breve descripción de las medidas de similaridad para particiones reportadas en la literatura, debido a su importancia en la combinación de agrupamientos.

#### 4 Evaluación de los agrupamientos (Clustering Validation)

Debido a que para un mismo conjunto de objetos, si aplicamos diferentes algoritmos de agrupamiento se pueden obtener resultados muy diferentes, surge la necesidad de evaluar las estructuraciones, para darle al usuario una idea del grado de eficacia de los resultados obtenidos. De estas medidas de evaluación (*índices*) se espera que sean objetivas y no tengan ninguna preferencia sobre algún algoritmo en particular. Sin embargo, teniendo en cuenta la variedad y la propia ambigüedad que existe en la definición de algoritmo de agrupamiento, surgen preguntas al respecto como son:

¿Podrá un índice de validación decir cuándo una estructuración de los datos es correcta, o es mejor que otra?, ¿Se podrá confeccionar un índice de validación capaz de trabajar de manera imparcial con cualquier tipo de estructuración de los datos?, ¿Qué es lo que realmente se le puede exigir a un índice de validación?

En los últimos años una gran cantidad de índices de validación han sido propuestos; éstos pueden ser duros o difusos, dependiendo del tipo de estructuración para la cual fueron concebidos. En este epígrafe abordaremos muy brevemente los distintos tipos de índices de validación duros, sus objetivos fundamentales y nuestra valoración al respecto.

Existen tres categorías de índices de validación: índices externos, índices relativos e índices internos [25] (Fig.2):

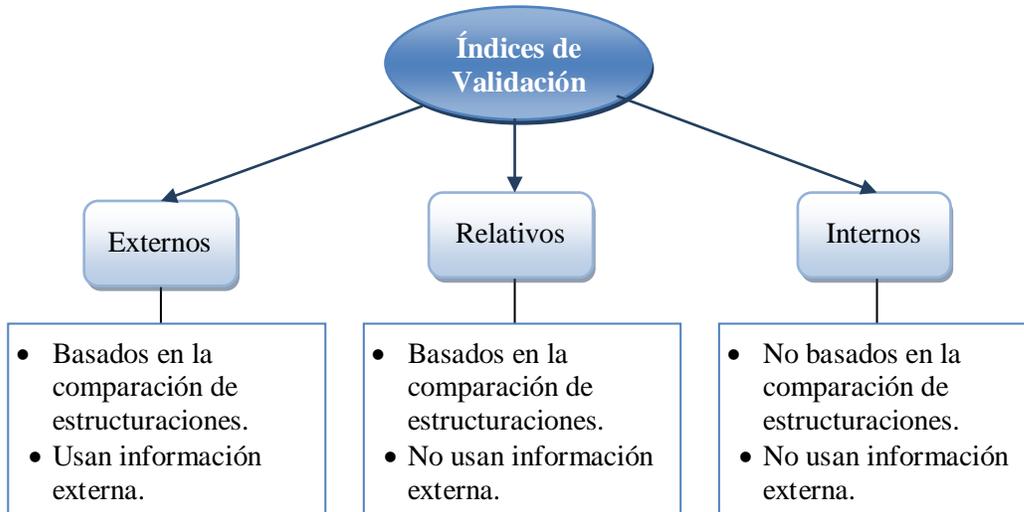


Fig. 2. Diferentes tipos de índices de validación y características generales

Los índices externos usan como patrón para compararse, una estructuración específica, la cual es obtenida a partir de una información previa acerca de los datos, donde este patrón es visto como la estructuración *real* o *verdadera* (*ground-truth*). Un agrupamiento obtenido por un clasificador es mejor en la medida que éste se parece más a dicho patrón. Entre los índices externos más usados se encuentran la medida F (*F-measure*) [26], el índice de Rand [27, 28], el índice de Rand Ajustado [28], el coeficiente de Jaccard [29] y el índice de Minkowski [30]. Sin embargo, un índice de validación externo puede ser cualquier medida de similaridad entre estructuraciones (ver epígrafe 5), siempre calculando la similaridad existente entre la estructuración obtenida por cierto clasificador, respecto a una estructuración conocida, la cual es asumida como *correcta* o *natural* para el conjunto de objetos analizados.

Los índices relativos no dependen de información externa, estos están basados en el cálculo de la consistencia de los clasificadores, comparando las estructuraciones obtenidas a partir del mismo algoritmo bajo condiciones diferentes, con el objetivo de decidir cuál revela mejor las características reales de los objetos. Entre estas medidas podemos encontrar al llamado Figura de Mérito (*Figure of Merit FOM*) [31] y el índice de Estabilidad [32, 33].

Los índices internos se basan solamente en la información del propio agrupamiento, no necesitan información adicional ni la repetición del proceso de agrupamiento. Un gran número de estos índices asumen que las estructuraciones obtenidas por los clasificadores deben cumplir que los objetos en un mismo agrupamiento deben estar más cerca que los objetos de agrupamientos diferentes, por tanto evalúan la compacidad u homogeneidad de cada agrupamiento y la separación de cada uno de los agrupamientos en la estructuración, por ejemplo: el índice de validación SD [29], los basados en teoría de grafos [34], el índice de Davies-Bouldin y el Ancho de la Silueta [35]. Otros índices evalúan qué tanto cumple una estructuración con el concepto de conectividad [36, 37], por ejemplo contando cuántas violaciones a la relación del vecino más cercano hay en una estructuración dada. De manera general existe una gran cantidad de estos índices, cada uno de los cuales aplica un criterio de evaluación diferente o la combinación de un conjunto de éstos, ya sea compacidad, separación, conectividad, etc.

A modo de resumen podemos decir que los índices externos necesitan de una estructuración patrón para compararse, con la cual no se cuenta en la mayoría de los casos, incluso ésta no tiene por qué ser única. Para un mismo problema dos o más estructuraciones de los datos pueden ser correctas dependiendo de los objetivos que se persigan en cada uno de los casos.

Los índices relativos muchas veces son tratados como índices internos, debido a que no usan información externa, sin embargo necesitan de la aplicación repetida del algoritmo de agrupamiento para comparar los resultados obtenidos. En estos casos, como no se conoce la estructura del resultado esperado, esta comparación puede hacerse de acuerdo al cumplimiento de alguna propiedad, es decir, a partir de éstos se puede obtener la partición de los datos que mejor cumple cierta propiedad, la cual es implícitamente o explícitamente medida por este índice.

Los índices internos funcionan de manera similar, un índice puede ser visto como una función que evalúa el cumplimiento de una propiedad o la combinación de un conjunto de éstas, en una partición.

Por tanto para nosotros, respondiendo a las preguntas planteadas al inicio del epígrafe, un índice no puede decir cuándo una estructuración es *correcta* o *mejor* que otra, esto depende en gran medida del problema particular a resolver. Un índice puede decir cuánto se cumple una o un conjunto de propiedades en una partición y a partir de esta información seleccionar la mejor partición respecto a estos criterios.

Tampoco se puede crear un índice de validación capaz de trabajar de manera imparcial con cualquier tipo de estructuración de los objetos. Un clasificador impone una estructura sobre los datos, si la propiedad que mide cierto índice no está acorde con la estructura impuesta por el clasificador, el índice va a valorar la clasificación como mala, incluso cuando dicha clasificación tenga sentido y brinde información relevante para un problema en particular. Por tanto se podría decir que dado un índice de validación, siempre se puede buscar un clasificador para el cual este índice no va a funcionar bien.

Precisamente esta imposibilidad de decidir con toda certeza, a partir de un conjunto de estructuraciones, cuál es la mejor, es uno de los motivos por el cual surge la necesidad de combinar estructuraciones como una variante de solución al problema de la clasificación no supervisada, en lugar de generar múltiples particiones y evaluarlas utilizando un índice y quedarnos con la mejor.

## 5 Medidas de similitud entre estructuraciones

De manera análoga a como ocurre con los algoritmos de agrupamiento, que necesitan de una medida de similitud (o disimilitud) para comparar los objetos con el fin de poderlos agrupar atendiendo a su similitud, una combinación de agrupamientos puede necesitar de una medida de similitud (disimilitud) entre estructuraciones con el objetivo de poder compararlas para obtener la *partición de consenso*. Esta necesidad se hace mayor, en los enfoques de combinación de agrupamientos que tienen como función objetivo encontrar de alguna manera la *partición media* del conjunto de estructuraciones [12, 38-40], es decir, la partición que mayor *información* comparta o más se *parezca* al conjunto de particiones de la combinación de agrupamientos.

La principal dificultad para encontrar una medida de similitud entre particiones es el hecho de que las etiquetas que son otorgadas a cada elemento después de aplicar un algoritmo de agrupamiento son simbólicas, lo cual puede ser un problema si se pretenden comparar directamente las etiquetas de cada partición. Por ejemplo, las dos particiones siguientes hechas

sobre un conjunto de nueve objetos, son iguales y sin embargo la comparación directa entre las etiquetas de las agrupaciones no coincide [41].

Objetos	a	b	c	d	e	f	g	h	i
<b>Partición 1</b>	1	1	1	2	2	2	3	3	3
<b>Partición 2</b>	3	3	3	1	1	1	2	2	2

En la literatura se han propuesto varias medidas de similaridad entre particiones duras y difusas (ver resumen en Kuncheva [41]). En este epígrafe trataremos algunas de las medidas duras que serán utilizadas por los diferentes modelos de combinación de agrupamientos expuestos en el epígrafe 6. Sea  $O = \{O_1, O_2, \dots, O_n\}$  un conjunto de objetos, se pueden definir las siguientes medidas de similaridad sobre particiones de  $O$ :

### 5.1 Índice Rand

El índice de Rand [27] es una medida de similaridad simple entre dos particiones  $P_1$  y  $P_2$ . Denotemos por  $n_{11}$  el número de pares de objetos de  $O$ , que están juntos en un mismo agrupamiento tanto en la partición  $P_1$  como en  $P_2$ ,  $n_{00}$  el número de pares de objetos que están en diferentes agrupamientos en  $P_1$  y  $P_2$ . Consecuentemente se define  $n_{01}$  y  $n_{10}$  como las cantidades de pares de objetos que pertenecen a un mismo cluster en una partición y a agrupamientos diferentes en la otra. El índice de Rand es:

$$r(A, B) = \frac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{01} + n_{10}}$$

Como existen  $n(n-1)/2$  combinaciones de pares de objetos en  $O$ . El índice Rand puede ser reescrito como:

$$r(P_1, P_2) = \frac{2 * (n_{00} + n_{11})}{n(n-1)}$$

El índice de Rand toma valor 1 cuando las particiones son iguales, sin importar el valor de la etiqueta asignada a cada agrupamiento.

### 5.2 Índice de Jaccard

Usando la misma notación usada en el índice de Rand, se define el índice de Jaccard [42] entre dos particiones  $P_1$  y  $P_2$  como:

$$J(P_1, P_2) = \frac{n_{11}}{n_{11} + n_{01} + n_{10}}$$

Esta medida toma valores en el intervalo  $(0, +\infty)$  y es muy similar al índice de Rand, solo que no toma en cuenta el caso en que los objetos pertenecen a diferentes agrupamientos en ambas particiones.

### 5.3 Información Mutua Normalizada

Otro concepto empleado con el propósito de evaluar la calidad de los agrupamientos es el de *información mutua*. En teoría de la información [43], la información mutua es una función simétrica que cuantifica la información compartida entre dos distribuciones. Sean  $P^{(a)} = \{C_1, C_2, \dots, C_{k^a}\}$  y  $P^{(b)} = \{D_1, D_2, \dots, D_{k^b}\}$  dos particiones de  $O$ , siendo  $k^a$  el número de agrupamientos en  $P^{(a)}$  y  $k^b$  el número de agrupamientos en  $P^{(b)}$ . Sea  $n_{i^*}$  el número de objetos en el agrupamiento  $i$ -ésimo de la partición  $P^{(a)}$ ,  $n_{*j}$  el número de objetos en el agrupamiento  $j$ -ésimo de la partición  $P^{(b)}$  y  $n_{ij}$  el número de objetos que están juntos tanto en el agrupamiento  $i$ -ésimo de la partición  $P^{(a)}$  como en el cluster  $j$ -ésimo de la partición  $P^{(b)}$ . La información mutua [43] entre  $P^{(a)}$  y  $P^{(b)}$  se expresa de la siguiente manera:

$$MI(P^{(a)}, P^{(b)}) = \sum_{i=1}^{k^a} \sum_{j=1}^{k^b} \frac{n_{ij}}{n} \log \left( \frac{n_{ij} n}{n_{i^*} n_{*j}} \right)$$

En el caso que nos ocupa, las particiones son tratadas como variables aleatorias, donde para calcular la similaridad entre las particiones se calcula la *información mutua normalizada* [12, 44] entre ellas, la cual se define de la siguiente manera para dos particiones  $P^{(a)}$  y  $P^{(b)}$ :

$$NMI(P^{(a)}, P^{(b)}) = \frac{-2 \sum_{i=1}^{k^a} \sum_{j=1}^{k^b} \frac{n_{ij}}{n} \log \left( \frac{n_{ij} n}{n_{i^*} n_{*j}} \right)}{\sum_{i=1}^{k^a} n_{i^*} \log \left( \frac{n_{i^*}}{n} \right) + \sum_{j=1}^{k^b} n_{*j} \log \left( \frac{n_{*j}}{n} \right)}$$

Si  $P^{(a)}$  y  $P^{(b)}$  son idénticas, entonces la información mutua normalizada toma su máximo valor, igual a 1. Si  $P^{(a)}$  y  $P^{(b)}$  son independientes entonces:

$$\frac{n_{ij} n}{n_{i^*} n_{*j}} \rightarrow 1, \text{ por tanto } NMI(P^{(a)}, P^{(b)}) \rightarrow 0, \text{ tomando el menor valor posible.}$$

### 5.4 Similaridad basada en kernel para grafos

En [40] se propone una medida de similaridad entre particiones usando una función kernel [45] para grafos. Dadas dos particiones de un conjunto de objetos  $O$ , primeramente cada partición es representada mediante un grafo de la siguiente manera: Sea  $P$  una partición, se define un grafo  $G_P = (V, E)$ , donde  $V = \{v_1, v_2, \dots, v_n\}$  y  $v_i$  es el nodo asociado al objeto  $O_i$ . En este grafo va a existir una arista entre los nodos  $v_i$  y  $v_j$  si y sólo si los objetos  $O_i$  y  $O_j$  están en el mismo agrupamiento en  $P$ . De esta manera todos los grafos obtenidos a partir de particiones, son grafos cuyas componentes conexas son grafos completos. Si denotamos como  $\mathbb{P}_O$  al conjunto de todas las particiones posibles de los objetos en  $O$  y como  $\mathbb{G}_O$  al conjunto de todos los grafos formados a partir de una partición en  $\mathbb{P}_O$ . Se puede ver que la función  $g: \mathbb{P}_O \rightarrow \mathbb{G}_O$  que a cada partición le asigna el grafo asociado, es una función biyectiva. Es decir, a cada partición solamente se le va a asignar un grafo y cada grafo de este tipo va a estar asociado a una sola partición.

Por tanto el cálculo de la similaridad entre dos particiones  $P_1, P_2$  es equivalente al cálculo de la similaridad entre los grafos formados a partir de ellas  $G_{P_1}, G_{P_2}$ .

Para esto se define un kernel para grafos basado en la comparación de todos los caminos del grafo (*path kernel*). El cálculo de la similaridad entre dos grafos usando este tipo de kernel es

usualmente muy costosa (exponencial) [46, 47], sin embargo en este caso particular, debido a la estructura del grafo con que se trabaja, es posible realizar los cálculos para comparar todos los caminos entre dos grafos de manera muy eficiente, usando una estructura de datos adecuada puede hacerse en orden lineal  $O(n)$ .

Sea  $\Sigma(G)$  el conjunto de todos los posible caminos  $h$  sobre un grafo  $G = (V, E)$ ,  $h \in V^s$  es una secuencia de nodos  $h_1 h_2 \dots h_s$ , donde existe una arista entre cada par de nodos consecutivos  $h_i, h_{i+1}$  y  $\forall i \neq j \ h_i \neq h_j$ , donde  $s$  es la longitud del camino.

Basado en la idea del Kernel Marginalizado [48] se define el kernel  $k$  como:

$$k(G, G') = \sum_{h \in \Sigma(G)} \sum_{h' \in \Sigma(G')} k_\delta(h = h') p(h/G) p(h'/G')$$

donde  $k_\delta(h = h')$  es el kernel de macheo o delta kernel, el cual toma valor 1 si  $h = h'$  y de lo contrario toma valor 0, y  $p(h/G)$  está dado por:

$$p(h/G) = p_{ini}(h_1) \prod_{i=2}^s p_t(h_i/h_{i-1}) p_{fin}(h_s)$$

siendo  $p_{ini}(h_1)$  la probabilidad de que el camino  $h$  comience en el vértice  $h_1$ ,  $p_t(h_i/h_{i-1})$  la probabilidad de que estando en el vértice  $h_{i-1}$  el próximo vértice del camino sea  $h_i$  y  $p_{fin}(h_s)$  es la probabilidad de que el camino termine en el vértice  $h_s$ .

En este caso, dadas las características del grafo (cada componente conexa es un grafo completo) se toma  $p_{ini}(h_1) = \frac{1}{n}$  ya que un camino cualquiera podría empezar con la misma probabilidad por cualquier nodo del grafo,  $p_t(h_i/h_{i-1}) = \frac{1}{|CC_h|}$  donde  $CC_h$  es la componente conexa que contiene a  $h$  y  $|CC_h|$  su cantidad de elementos,  $p_t(h_i/h_{i-1})$  toma este valor, ya que en una misma componente conexa de este tipo de grafo, todos los nodos están conectados con todos, por tanto, estando en un vértice en particular, es posible moverse hacia cualquier otro vértice de la componente conexa con la misma probabilidad. Finalmente  $p_{fin}(h_s) = \frac{1}{|CC_h|}$  ya que estando en cualquier nodo, existen  $n$  posibilidades,  $n-1$  de ir a nodos vecinos y 1 de quedarse en el nodo que se está y terminar el camino, asumiendo que estas opciones son igualmente probables, terminar el camino en un nodo tiene la misma probabilidad que pasar a un nodo en particular y esta probabilidad es igual a  $\frac{1}{|CC_h|}$ . Por tanto  $p(h/G) = \frac{1}{n|CC_h|^{l-1}}$ .

La cantidad de caminos diferentes en un grafo es exponencial respecto a la cantidad de nodos en el mismo, por tanto la comparación entre todos los nodos de dos grafos sería de esta misma complejidad, lo cual es intratable computacionalmente. Sin embargo,  $p(h/G)$  toma el mismo valor para todos los caminos del mismo tamaño en una misma componente conexa y por tanto no es necesario comparar todos los caminos existentes en los dos grafos, sino sólo conjuntos de caminos que tienen la misma probabilidad  $p(h/G)$  asociada, reduciendo drásticamente el costo computacional.

Esta medida de similaridad al ser un kernel, ofrece grandes ventajas ya que permite establecer la similaridad entre las particiones a partir del cálculo de un producto escalar en un espacio de Hilbert, permitiendo el trabajo con los datos de manera geométrica y utilizar algoritmos que hacen uso del algebra lineal y la geometría analítica. [45].

En [40] se utiliza esta medida de similaridad para llegar a la partición de consenso, calculando el valor promedio de las representaciones de las particiones en el espacio de Hilbert asociado a este kernel.

## 6 Combinación de agrupamientos (*Cluster Ensemble*)

Utilizando la misma notación, el problema de la combinación de resultados de agrupamientos puede ser formalizado de la siguiente manera:

Siendo  $O = \{O_1, O_2, \dots, O_n\}$  el conjunto de objetos, una combinación de agrupamientos es un conjunto  $\mathbb{P} = \{P_1, P_2, \dots, P_m\}$ , donde  $P_i$  es una partición de los objetos  $O$ , para todo  $i = 1, 2, \dots, m$ . Donde el objetivo principal es encontrar una nueva partición  $P^*$  de los datos a partir de las particiones en  $\mathbb{P}$ . Para encontrar esta partición es necesaria una *función de consenso*, la cual es la encargada de combinar toda la información existente en el conjunto de las particiones  $\mathbb{P}$  en una partición final  $P^*$ .

Una buena estrategia de combinación, debe permitir encontrar nuevas estructuraciones más *consistentes* que las existentes, entendiendo por consistente a una estructuración que comparte gran cantidad de información o es muy similar al conjunto de particiones. Esta estructuración de consenso debe ser además, lo más invariante posible a pequeñas variaciones en los datos, es decir, debe ser suficientemente robusta ante información ruidosa. Éstas, entre otras propiedades, son planteadas por los autores de algoritmos de combinación de resultados de clasificadores no supervisados, sin embargo no existe un criterio común ni una formalización rigurosa de las características que debe tener la estructuración de consenso, más bien, cada autor propone las propiedades que cree que debe cumplir un buen mecanismo de combinación.

En el diagrama en la Fig.3 se muestran los pasos principales de la combinación de resultados de clasificadores no supervisados, el mecanismo de generación y la función de consenso.

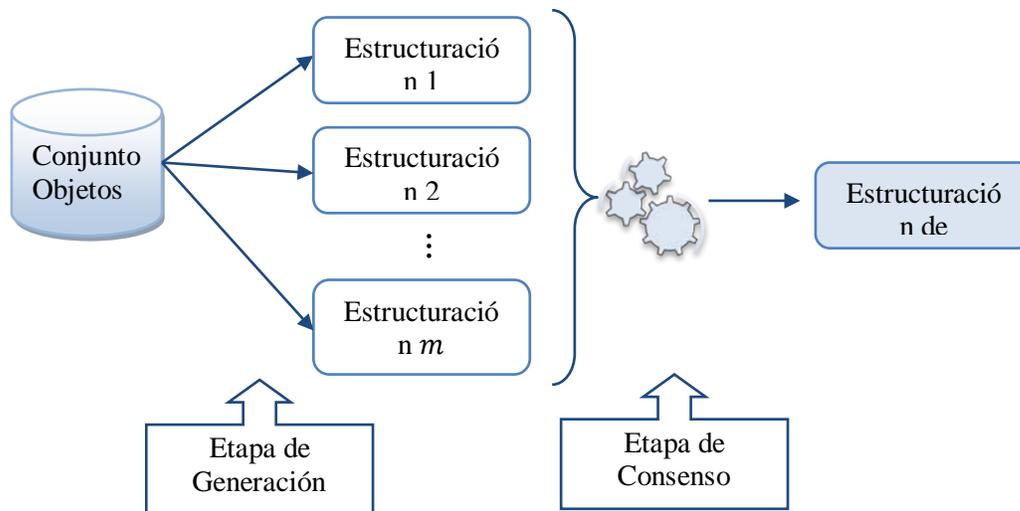
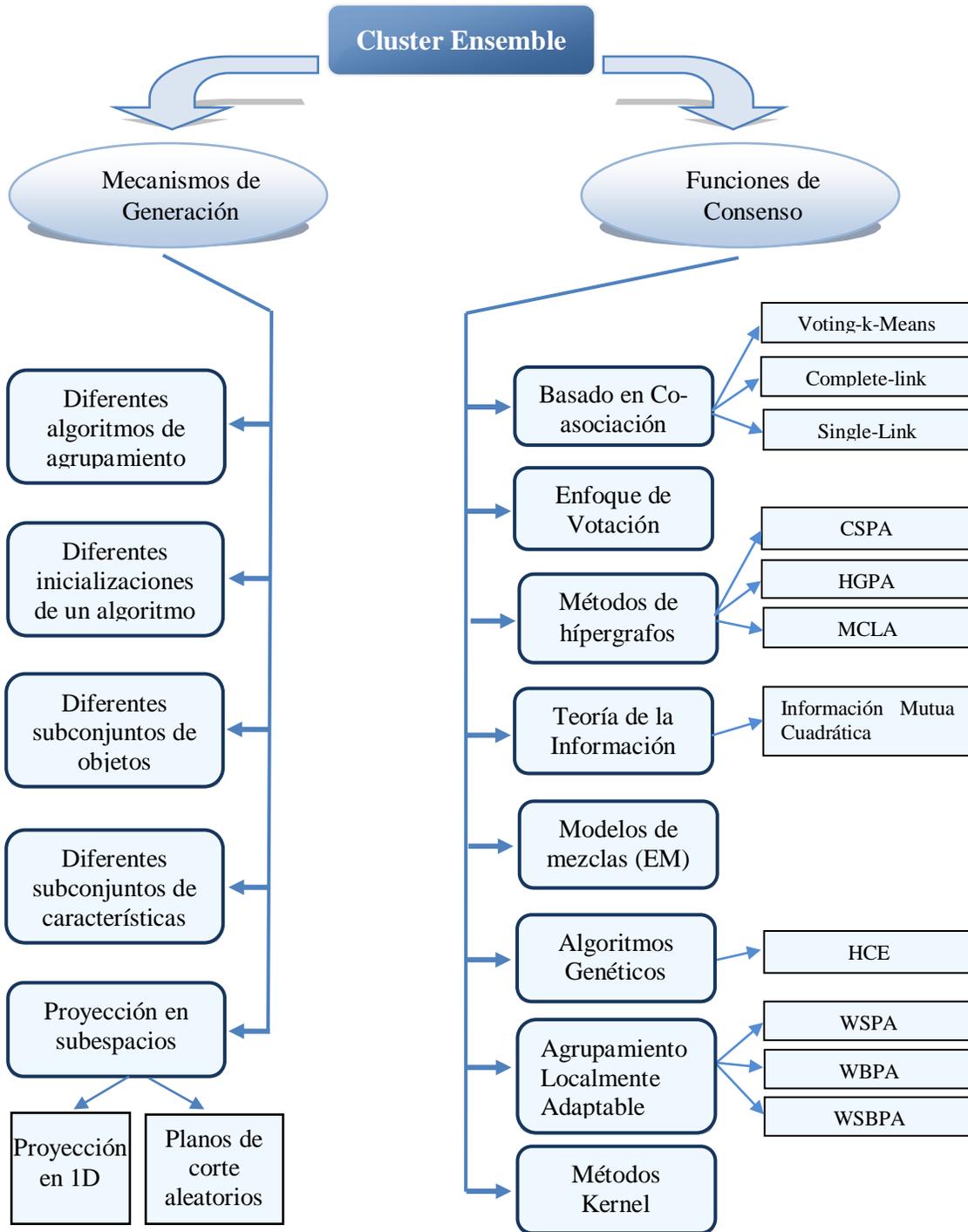


Fig.3. Diagrama del proceso general de la combinación de agrupamientos

En el proceso de generación de la combinación de agrupamientos, las particiones pueden ser obtenidas utilizando múltiples modelos de algoritmos de agrupamiento sobre el mismo conjunto  $O$ , usando diferentes algoritmos o diferentes clasificadores [11, 12, 38], proyectando los datos

sobre diferentes subespacios [38, 49], utilizando diferentes subespacios de características de los objetos [38], seleccionando diferentes subconjuntos de los objetos en las distintas corridas de los algoritmos [50, 51], etc. (ver epígrafe 6.1)

Por otra parte, desde la introducción de la combinación de agrupamientos por [52], una gran cantidad de funciones de consenso han sido reportadas en la literatura [53]: los métodos basados en co-asociación [11, 54-56], el enfoque de votación [50, 57, 58], métodos basados en teoría de la información (ej: información mutua cuadrática) [38], en modelos de mezclas (EM) [39], en algoritmos genéticos [59], en particionamiento de grafos usando algoritmos de agrupamiento adaptativos locales [60], usando métodos kernel [40], etc. En la Fig.4. se muestra una taxonomía de los mecanismos de generación y de consenso, los cuales serán explicados, haciendo un análisis crítico en los epígrafes siguientes.



**Fig. 4.** Taxonomía de los diferentes enfoques de combinación de agrupamientos. Izquierda: Diferentes formas de obtener el conjunto de particiones que conforman la combinación de agrupamientos. Derecha: Diferentes funciones de consenso

## 6.1 Métodos de generación de la combinación de agrupamientos

El primer paso en todo modelo de combinación de agrupamientos es la generación. La idea aquí es obtener un conjunto de agrupamientos (en la mayoría de los casos particiones) a partir del mismo conjunto de objetos. Existen métodos como el *voting-k-means* [52] (epígrafe 6.2.1) que exigen un proceso de generación bien determinado, en este caso todas las particiones deben ser obtenidas a partir de la aplicación del algoritmo k-Means utilizando diferentes inicializaciones de los parámetros, es decir, diferentes clasificadores todos pertenecientes a la familia de algoritmos k-Means. También en [40] se propone un enfoque al problema, en el cual cada partición obtenida en el proceso de generación es evaluada utilizando índices de validación (epígrafe 4), los cuales pueden utilizar las representaciones de los objetos en el proceso de evaluación, por tanto, se necesita conocer el espacio de representación de los objetos y la medida de similaridad utilizada entre los mismos.

Sin embargo, la mayoría de los modelos de combinación de agrupamientos no tienen en cuenta la manera en que se generó el conjunto de agrupamientos (particiones). Ellos parten del hecho de contar con este conjunto y a partir de ahí se diferencian en la manera de consolidar la información, es decir, se diferencian en la función de consenso y en los algoritmos utilizados para llegar a la solución óptima.

Por tanto, en el proceso de generación pueden utilizarse diferentes modelos de algoritmos de agrupamiento, diferentes algoritmos o diferentes clasificadores, pudiendo usarse distintas representaciones de los objetos o utilizando diferentes subconjuntos de objetos.

Otra técnica para la generación del conjunto de agrupamientos es utilizar proyecciones de los objetos en diferentes subespacios, por ejemplo, utilizando los algoritmos de agrupamientos débiles [38] (*weak clustering algorithms*). La idea de estos algoritmos es conformar el conjunto de agrupamientos utilizando métodos que sean sólo un *poco mejor* que una partición aleatoria de los datos. Para esto se proponen dos mecanismos diferentes:

- Agrupamiento de proyecciones aleatorias unidimensionales de los datos, las cuales pueden ser generalizadas a proyecciones aleatorias de más de una dimensión.
- Agrupamiento mediante la separación de los datos usando un número aleatorio de hiperplanos. Por ejemplo, si sólo se usa un hiperplano los datos quedarían separados en dos grupos.

Según los autores, la utilización de los algoritmos de agrupamiento débiles permite obtener un conjunto de agrupamientos muy grande, pues éstos están basados en cálculos simples sobre los datos, lo cual permite que puedan ser aplicados de manera muy eficiente. Tener un conjunto de agrupamientos grande permite, de manera general, que el efecto de estructuraciones no muy buenas sea compensado por la decisión de la gran mayoría de las estructuraciones, y así eliminar un poco el efecto que puede ocasionar un agrupamiento muy *ruidoso* en la estructuración de consenso. Además un conjunto de agrupamientos grande aumenta la diversidad de las estructuraciones dando la posibilidad de aparición a estructuraciones más ajustadas a la naturaleza del conjunto de datos analizados.

## 6.2 Funciones de consenso

El gran reto en la combinación de resultados de clasificadores no supervisados es la definición de una función de consenso apropiada, capaz de mejorar los resultados de los agrupamientos individuales. En la Fig.4. se muestra la diversidad de herramientas utilizadas para definir funciones de consenso (particionamiento de grafos, hipergrafos, algoritmos genéticos, etc.). Sin embargo, las funciones de consenso no sólo se diferencian en la manera de encontrar la

estructuración de consenso. También se diferencian en la formulación teórica de lo que debe ser una estructuración de consenso, qué características debe cumplir dicha estructuración y cómo puede definirse matemáticamente el proceso de búsqueda de la misma.

Atendiendo a este criterio existen dos tendencias principales: La primera está basada en una idea de votación entre objetos, donde es analizada la cantidad de veces que un elemento pertenece a un agrupamiento, o la cantidad de veces que pares de elementos se encuentran juntos en un mismo agrupamiento, es decir, se busca información sobre la distribución de los objetos individuales en los agrupamientos y a partir de dicha información se encuentra una estructuración de consenso. Generalmente este tipo de función de consenso necesita resolver el problema de la correspondencia de las etiquetas de los agrupamientos (el cual sólo puede ser resuelto con cierto grado de eficacia si todas las estructuraciones tienen la misma cantidad de agrupamientos, lo que es una restricción muy fuerte para la combinación) o para evadir el problema de la correspondencia de las etiquetas se analiza la cantidad de veces que dos objetos son colocados en el mismo agrupamiento, de aquí se obtiene la matriz de co-asociación de los objetos. Sin embargo, partiendo de ésta, es necesario un nuevo clasificador para obtener la estructuración de consenso. La segunda variante de función de consenso está basada en la idea de la búsqueda de la partición media. En este enfoque es necesaria la existencia de una medida de similitud entre particiones y el consenso se define como la estructuración que maximice la similitud con todas las particiones en la combinación.

En ambos enfoques existen problemáticas sin una solución definitiva, por ejemplo, en el primer enfoque generalmente es necesario aplicar un clasificador como paso final para hallar el consenso, pero ¿cuál es el clasificador que se debe usar, cuáles son los parámetros correctos?, ¿qué requisitos debe cumplir la estructuración de consenso que se busca?

En el segundo enfoque, primeramente se necesita de una medida de similitud entre estructuraciones, pero ¿cuál es la medida de similitud correcta para comparar estructuraciones? Además, se plantea obtener el consenso a partir de la búsqueda del óptimo en un problema de optimización exponencial, sin embargo, ¿cuál es la mejor heurística para resolver o acercarse a la solución del problema?

Una vez obtenida una estructuración de consenso, por cualquiera de los dos enfoques, ¿cómo evaluar los resultados obtenidos, hasta qué punto podemos asegurar que es mejor utilizar la estructuración obtenida por la combinación en vez de cualquier otra?, ¿cuál es la garantía que da este proceso de combinación?

En este epígrafe se muestran las distintas variantes de solución propuestas en la literatura haciendo un análisis crítico a partir de las interrogantes anteriores.

### 6.2.1 Basadas en Co-asociación

Partiendo de la idea de que diferentes agrupamientos de los datos pueden tener cada uno sus fortalezas y debilidades, se espera que la contribución conjunta de los mismos tenga un efecto compensatorio en los resultados y que permita mejorar los mismos respecto a agrupaciones obtenidas utilizando un solo clasificador.

Considerando las particiones obtenidas a partir de diferentes clasificadores, éstas son mapeadas en un espacio intermedio: la matriz de co-asociación, donde la posición  $(i, j)$  representa el número de veces que el par de objetos  $O_i$  y  $O_j$  co-ocurren en el mismo agrupamiento. Cada co-ocurrencia es vista como un voto para que ambos objetos se encuentren en el mismo agrupamiento en la partición de consenso. Dividiendo los valores de la matriz por el número de particiones en la combinación de agrupamientos se obtienen los votos normalizados.

Partiendo de la matriz de co-ocurrencia existen varios métodos propuestos en la literatura. En [52] la partición resultado es obtenida a partir de la comparación de la matriz con el umbral fijo 0.5 y uniendo en el mismo agrupamiento todos los objetos que mantienen una asociación mayor que 0.5. En la Tabla 1 se resumen los pasos de la metodología propuesta:

**Tabla 1.** Función de consenso usando el esquema de votación mayoritaria

<p><i>Entrada:</i> <math>n</math> objetos; <math>\mathbb{P}</math> combinación de agrupamientos conformada por <math>m</math> particiones del conjunto de objetos.</p> <p><i>Salida:</i> Partición de consenso.</p> <p><i>Inicialización:</i> Hacer la matriz de co-asociación <math>co\_assoc</math> una matriz nula de dimensión <math>n \times n</math>.</p> <p><i>Pasos:</i></p> <ol style="list-style-type: none"> <li>1. Producir las particiones de los datos e inicializar la matriz de co-asociación:           <ul style="list-style-type: none"> <li><i>For</i> <math>i = 1</math> <i>to</i> <math>m</math> <i>do</i>:</li> <li>1.1. Correr el <math>i</math>-ésimo método de agrupamiento y producir la partición <math>P_i</math>.</li> <li>1.2. Actualizar la matriz de co-asociación de acuerdo con:               <ul style="list-style-type: none"> <li>Para cada par de objetos <math>(i, j)</math> en el mismo agrupamiento en <math>P</math> hacer:                   <math display="block">co\_assoc(i, j) = co\_assoc(i, j) + 1/m</math> </li> </ul> </li> </ul> </li> <li>2. Obtener los agrupamientos de la <i>partición resultado</i> aplicando un umbral sobre la matriz de co-asociación.           <ol style="list-style-type: none"> <li>2.1. Encontrar las asociaciones por votaciones mayoritarias:               <ul style="list-style-type: none"> <li>Para cada par de objetos <math>(i, j)</math>, tal que <math>co\_assoc(i, j) &gt; 0.5</math> unir los objetos en un mismo agrupamiento; si los objetos pertenecen a dos agrupamientos diferentes previamente formados, se unen en uno solo.</li> </ul> </li> <li>2.2. Para cada objeto no incluido en ningún agrupamiento, formar un agrupamiento que sólo lo contenga a él.</li> </ol> </li> <li>3. Devolver los agrupamientos formados.</li> </ol>
---

Igualmente en [52] se propone el algoritmo *voting-k-means*, en el cual se incorpora la metodología anterior en el contexto del algoritmo de agrupamiento k-Means. La idea es lograr obtener estructuraciones de los datos complejos a partir de la unión de estructuraciones hiperesféricas pequeñas, obtenidas mediante la aplicación del algoritmo k-Means. Para esto se utiliza un valor grande de  $k$  (cantidad de agrupamientos en que va a particionar el conjunto de objetos el algoritmo k-Means), normalmente igual a  $\sqrt{n}$ . Si  $k$  toma un valor grande en comparación con el número de agrupamientos que se quiere obtener de los objetos, cada agrupamiento va a ser intrínsecamente separado en pequeños agrupamientos, entonces cuando sean promediadas todas las asociaciones producidas por los  $m$  algoritmos de k-Means aplicados, es esperado que se obtengan altos valores de votación en los patrones vecinos permitiendo que la estructura subyacente de los datos sea recuperada aplicando el umbral antes mencionado a la matriz de co-asociación.

**Tabla 2.** Algoritmo Voting-k-Means

<p><i>Entrada:</i> <math>n</math> objetos; <math>k</math> número inicial de agrupamientos (por defecto <math>k = \sqrt{n}</math>); <math>m</math> número de iteraciones.</p> <p><i>Salida:</i> Partición de los datos.</p> <p><i>Inicialización:</i> Hacer la matriz de co-asociación <math>co\_assoc</math> una matriz nula de</p>
---

dimensión  $n \times n$ .

*Pasos:*

1. Hacer  $m$  veces:
  - 1.1. Escoger aleatoriamente  $k$  centroides entre los  $n$  elementos del conjunto de datos.
  - 1.2. Organizar las  $n$  muestras en orden aleatorio, manteniendo en algún lugar la configuración inicial.
  - 1.3. Aplicar k-Means con los objetos reordenados y los centroides escogidos y actualizar la matriz de co-asociación.
2. Detectar los agrupamientos de la partición de consenso a partir de la matriz de co-asociación, usando la técnica propuesta en Tabla 1.

*Características:*

- Es el primer algoritmo de combinación de agrupamientos reportado en la literatura. Aquí es donde se propone la idea de combinar algoritmos de agrupamientos.
- El problema que plantea resolver es encontrar la “*estructuración natural*” de los objetos, la cual no tiene por qué ser única. Esto depende del problema particular que se quiera resolver. Sin un previo conocimiento de las características de los objetos, diversas estructuraciones de los mismos pueden ser igualmente válidas.
- Requiere una gran cantidad de memoria  $O(n^2)$ , en el paso de formación de la matriz de co-asociación, por tanto no puede ser aplicado a grandes volúmenes de datos.
- El umbral para obtener las agrupaciones finales está fijo en 0.5, el cual en ciertas ocasiones pudiera no ser el valor más apropiado.

En [11] se explora la idea de *acumulación de evidencia*, que es un trabajo muy similar al [52] pero se formalizan algunos aspectos fundamentales, como la función objetivo de la combinación de agrupamientos.

Los objetos son separados en un gran número de pequeños y compactos agrupamientos, obtenidos a partir de diferentes corridas del algoritmo k-Means con inicializaciones aleatorias de los parámetros. Igualmente, los resultados obtenidos son mapeados dentro de una matriz de co-asociación la cual provee una medida de similaridad entre las muestras. La partición final de los objetos es obtenida a partir de la aplicación de un nuevo algoritmo de agrupamiento sobre la nueva matriz de similaridad. Este método cumple con las características de un enfoque separa-y-mezcla:

*Separa:* Descomponer los objetos en un gran número de agrupaciones pequeñas e hiperesféricas. El algoritmo k-Means es el que se usa para llegar a éstas.

*Combina:* Las diferentes particiones se combinan a través del algoritmo de votación, en el cual se va actualizando los valores de la matriz de co-asociación a partir de la fórmula:

$$co\_assoc(i, j) = \frac{vote\ s_{ij}}{n}.$$

*Mezcla:* Se hace énfasis en las relaciones entre agrupamientos vecinos y se aplica un algoritmo para encontrar un árbol abarcador de costo mínimo (“minimum spanning tree”) es decir, viendo la matriz de co-asociación como la matriz de adyacencia de un grafo, se busca un árbol que contenga a todos los nodos del grafo y que el costo de sus aristas sea mínimo y se cortan vínculos débiles en el árbol utilizando un umbral  $t$ . Esto es equivalente a cortar el dendrograma producido por el algoritmo Single Link (SL) sobre la matriz de similaridad usando el umbral  $t$ . También como variante de este algoritmo puede utilizarse el Complete-Link (CL).

**Tabla 3.** Combinación de agrupamientos usando *acumulación de evidencia*

<p><i>Entrada:</i> <math>n</math> objetos.  <math>k</math> número inicial de agrupamientos.  <math>m</math> tamaño de la combinación de agrupamientos  <math>t</math> umbral.</p> <p><i>Salida:</i> Partición de consenso.</p> <p><i>Inicialización:</i> Hacer <math>co\_assoc</math> una matriz nula de dimensión <math>n \times n</math>.</p> <p><i>Pasos:</i></p> <ol style="list-style-type: none"> <li>1. Hacer <math>m</math> veces: <ol style="list-style-type: none"> <li>1.1. Seleccionar <math>k</math> centroides aleatoriamente.</li> <li>1.2. Aplicar k-Means con dichos centroides y producir la partición <math>P</math>.</li> <li>1.3. Actualizar la matriz de co-asociación. Para cada par <math>(i, j)</math> en el mismo agrupamiento en <math>P</math>, hacer: <math>co\_assoc(i, j) = co\_assoc(i, j) + 1/m</math>.</li> </ol> </li> <li>2. Detectar los agrupamientos consistentes en la matriz de co-asociación usando el algoritmo Single Link (SL): <ol style="list-style-type: none"> <li>2.1. Para cada par <math>(i, j)</math> tal que <math>co\_assoc(i, j) &gt; t</math>, mezclar los patrones en un mismo agrupamiento, y si éstos ya pertenecían a agrupamientos previamente formados, unirlos.</li> <li>2.2. Para cada patrón restante no incluido en ningún agrupamiento, formar un nuevo agrupamiento que sólo lo contenga a él.</li> </ol> </li> </ol>
--

*Características:*

- Elimina el umbral fijo en 0.5, el cual no siempre era el mejor valor de corte en la matriz de co-asociación, pero introduce un parámetro  $t$  el cual es difícil de estimar su valor más apropiado para un problema en particular.
- La matriz de co-asociación es vista como una nueva medida de similaridad entre los patrones. En un problema donde no se tenga ninguna información de los objetos puede ser muy útil, pero en caso de que el problema a resolver haya sido bien modelado matemáticamente, la nueva medida de similaridad no tiene por qué ser más ajustada a los objetos que la propia medida que se utilizó para la generación de las particiones iniciales. Además, esta nueva medida de similaridad está basada en los agrupamientos obtenidos a partir del modelo de algoritmos de agrupamiento k-Means, el cual agrupa los objetos de forma hipersférica. Por tanto, esta similaridad va a estar condicionada implícitamente por este tipo de estructuración de los datos.
- Después de obtenida la matriz de co-asociación, se llega a la partición de consenso a partir de la aplicación de un nuevo algoritmo de agrupamiento (en este caso SL). De esta manera no se resuelve la esencia del problema, ya que el SL va a imponer una estructura sobre los datos, y por tanto la aplicación de cualquier otro algoritmo de agrupamiento sobre la matriz daría un resultado diferente.

En [44] se proponen las siguientes propiedades que debe cumplir la partición de consenso obtenida de la combinación de agrupamientos:

- a) *Consistencia:* Esto significa que la partición de consenso  $P^*$  debe de alguna manera estar de acuerdo con las particiones individuales  $\mathcal{P}_i, i = 1, \dots, m$ .
- b) *Robustez a pequeñas variaciones* en  $\mathbb{P}$ : El número de agrupamientos y la pertenencia de los elementos a cada agrupamiento en  $P^*$  no debe cambiar significativamente con pequeñas perturbaciones en las particiones en  $\mathbb{P}$ .

c) *Ajustarse al “ground truth”*: Si se tiene la información de la *estructuración natural de los datos*;  $P^*$  debe tener una estructuración similar.

La similaridad entre dos particiones es calculada a partir de la información mutua normalizada (epígrafe 5.1.3) y la consistencia entre una partición  $P$  y la combinación de agrupamientos  $\mathbb{P}$  se garantiza a partir de la información mutua normalizada promedio [12] (epígrafe 6.2.3)

La propiedad de la robustez es analizada de la siguiente manera:

Sea  $\mathbb{P}^B = \{\mathbb{P}^{b_1}, \dots, \mathbb{P}^{b_B}\}$  un conjunto de combinaciones de agrupamientos obtenidas mediante un proceso de muestreo con reemplazo a partir de  $\mathbb{P}$  y sea  $\mathbb{P}^{*B} = \{P^{*b_1}, \dots, P^{*b_B}\}$  el correspondiente conjunto de funciones de consenso  $\mathbb{P}^B$ . El valor medio de la información mutua normalizada promedio entre  $\mathbb{P}^B$  y  $\mathbb{P}^{*B}$  es calculado mediante:

$$\overline{NMI(P^{*k_b}, \mathbb{P}^b)} = \frac{1}{B} \sum_{i=1}^B NMI(P^{*k_{b_i}}, \mathbb{P}^{b_i})$$

y la varianza correspondiente se define de la siguiente manera:

$$var\{NMI(P^{*k_b}, \mathbb{P}^b)\} = \frac{1}{B-1} \sum_{i=1}^B \left( NMI(P^{*k_{b_i}}, \mathbb{P}^{b_i}) - \overline{NMI(P^{*k_b}, \mathbb{P}^b)} \right)^2$$

Es esperado que una combinación de particiones robusta sea estable a pequeñas variaciones en la combinación de agrupamientos. Atendiendo a esta propiedad la función objetivo se puede definir como:

$$P^* = \min_k \left\{ var\{NMI(P^{*k_b}, \mathbb{P}^b)\} \right\}$$

A pesar de la definición de una función objetivo que garantice la consistencia y la robustez de la partición de consenso, el algoritmo propuesto es exactamente igual al propuesto en [11], por tanto, en este trabajo no se propone ningún algoritmo nuevo de combinación de agrupamientos, sino que se trata de dar una justificación formal para el uso de las técnicas basadas en la matriz de co-asociación para la combinación de agrupamientos. No obstante, no se puede afirmar que los algoritmos propuestos satisfagan con la función objetivo antes expuesta, sino que son heurísticas a seguir que deben dar soluciones *cercanas* a la solución de la función objetivo.

## 6.2.2 Enfoque de votación

En [58] se propone un mecanismo de combinación de agrupamientos basado en un esquema de votación que consta de 3 pasos fundamentales: Generación de la combinación de agrupamientos; proceso de votación utilizando el conjunto de particiones y el proceso de mezclado de las votaciones.

Para la generación se pueden utilizar diferentes clasificadores, en el proceso de votación la principal dificultad es que es necesario hacer una correspondencia entre las distintas etiquetas de los grupos.

Teniendo  $m$  particiones de los objetos, primeramente para dos de ellas  $P_1, P_2$ , se calcula la cantidad de elementos comunes que hay en cada par de agrupamientos  $(C_i^1, C_j^2)$ , donde  $C_i^1 \in P_1$  y  $C_j^2 \in P_2$  y se hacen corresponder las etiquetas de los agrupamientos que más elementos

compartan. Este proceso se repite para los agrupamientos restantes hasta obtener una correspondencia total entre las etiquetas de ambas particiones, luego a partir de éstas se van ajustando las etiquetas de las particiones restantes.

Posteriormente, cada objeto va a dar un voto de pertenencia a un agrupamiento (representado por una etiqueta) con valor igual a la cantidad de veces que a este elemento le fue asignado dicha etiqueta en las distintas particiones pertenecientes al conjunto de agrupamientos.

El resultado del proceso de votación es una partición difusa. Después de votar las  $m$  particiones, se obtiene para cada objeto  $O_i$  y para cada agrupamiento representado por una etiqueta  $j$  un valor  $D_j(O_i)$  el cual representa la fracción de la cantidad de veces que al objeto  $O_i$  le fue otorgada la etiqueta  $j$  entre la cantidad de particiones.

En el último paso es asignado cada objeto  $O_i$  al agrupamiento  $t = \arg \max_j D_j(O_i)$ , es decir, se hace pertenecer cada objeto al agrupamiento en el cual tiene mayor grado de pertenencia.

Posteriormente se define  $n(C_k, C_j) = \text{mean}_{O \in C_k} D_j(O)$  que es una medida de qué tanto los elementos del agrupamiento  $k$ -ésimo pertenecen al agrupamiento  $j$ -ésimo. Por tanto  $n(C_k, C_j)$  es una medida (no simétrica) de relación de vecindad de dos agrupamientos y se dice que el agrupamiento  $C_l$  es el más cercano al  $C_k$  si  $n(C_k, C_l) = \max_{j \neq k} n(C_k, C_j)$ . Utilizando esta medida es que se procede a mezclar agrupamientos para obtener la partición de consenso. Para esto se unen agrupamientos que cumplan una de las siguientes condiciones:

1. Se unen los agrupamientos  $C_k$  y  $C_j$  si  $C_k$  es el agrupamiento más cercano a  $C_j$  y  $C_j$  es el agrupamiento más cercano a  $C_k$ .
2. Se une un conjunto de agrupamientos  $C_{k_1}, C_{k_2}, \dots, C_{k_n}$ , si  $C_{k_i}$  es el agrupamiento más cercano al  $C_{k_{i+1}}$ , para todo  $i = 1, 2, \dots, n-1$ , y el agrupamiento  $C_{k-1}$  es el más cercano al  $C_{k_n}$ .

#### Características

- Lo primero que hace es tratar de resolver el problema de la correspondencia de las etiquetas, el cual cuando las particiones no tienen la misma cantidad de agrupamientos no puede hacerse de manera satisfactoria. Por tanto este método sólo dará resultados aceptables cuando todas las particiones de los objetos tengan el mismo número de agrupamientos, lo cual es una restricción fuerte al problema.
- Producto de la manera en que se trata de resolver el problema de la correspondencia de etiquetas, la solución obtenida al final del proceso no es única. Es decir, diferentes corridas del algoritmo pueden dar resultados diferentes.

### 6.2.3 Combinación de agrupamientos basada en hipergrafos

En [12] se define la combinación de agrupamientos como una herramienta para consolidar los resultados de diferentes algoritmos de agrupamiento, para los autores de este trabajo una buena combinación debe cumplir con las propiedades siguientes:

- *Calidad y Robustez:* La combinación de varios algoritmos de agrupamiento, debe mejorar la calidad y la robustez de los resultados.
- *Reutilización de conocimiento:* En muchas aplicaciones, puede existir varios resultados de algoritmos de agrupamiento aplicados sobre un conjunto de objetos. La idea es poder combinar estos resultados que ya existían sin tener que regresar a los objetos originales del problema.

- *Ejecución distribuida*: Permitir la combinación de resultados de algoritmos de agrupamiento utilizando programación distribuida.

La partición de consenso se define como la partición que mayor información comparte con todas las particiones a combinar. Para medir la información que comparten dos particiones se utiliza la información mutua [43], a partir de la cual se define la *información mutua normalizada* (NMI), la cual es una medida de similitud entre particiones ( ver epígrafe 5.1.3):

$$NMI(P^{(a)}, P^{(b)}) = \frac{-2 \sum_{i=1}^{k^a} \sum_{j=1}^{k^b} \frac{N_{ij}}{n} \log \left( \frac{N_{ij} n}{N_i N_j} \right)}{\sum_{i=1}^{k^a} N_i \log \left( \frac{N_i}{n} \right) + \sum_{j=1}^{k^b} N_j \log \left( \frac{N_j}{n} \right)}$$

Se propone como resultado óptimo la partición que maximice la información mutua normalizada con todas las particiones de la combinación de agrupamientos, para esto se busca la partición  $P_{(k)}^*$  más parecida a  $\mathbb{P}$  (conjunto de particiones) para cada uno de los valores de  $k$  (cantidad de agrupamientos en la partición) fijo, de la siguiente manera:

$$P_{(k)}^* = \arg \max_P \sum_{j=1}^m NMI(P, P_j) \quad (1)$$

donde  $P$  varía por todas las particiones posibles con  $k$  agrupamientos. La solución trivial de esta función objetivo es la búsqueda exhaustiva sobre todas las posibles particiones con  $k$  agrupamientos (aproximadamente  $k^n / k!$  para  $n \gg k$ ), lo cual es computacionalmente intratable. En [12] se proponen tres heurísticas eficientes para resolver el problema, las tres parten de representar la combinación de agrupamientos a partir de un hipergrafo.

Para cada partición  $P_i$ , se construye la matriz  $H^{(i)} \in \mathbb{N}^{n \times k^i}$ , en la cual cada agrupamiento es representado como una hiperarista (columna) como se indica en la Tabla 4. Notar que filas con objetos para los cuales no se conoce su etiqueta, quedan con cero en todas las columnas.

La concatenación de las matrices  $H = \{H^{(1)} \dots H^{(m)}\}$  define la matriz de adyacencia del hipergrafo con  $n$  vértices y  $\sum_{i=1}^m k^{(i)}$  hiperaristas. Cada vector columna  $h_a$  corresponde a una hiperarista, donde 1 indica que el vértice correspondiente a dicha fila pertenece a la hiperarista y 0 indica que no. De esta forma queda mapeado cada agrupamiento como una hiperarista y el conjunto de agrupamientos como un hipergrafo.

**Tabla 4.** Representación por hiper-grafo de un problema de combinación de agrupamientos. Izquierda:  $\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)}, \lambda^{(4)}$  son 4 particiones del conjunto de objetos  $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ . Derecha: hiper-grafo equivalente con 11 hiper-aristas

	$\lambda^{(1)}$	$\lambda^{(2)}$	$\lambda^{(3)}$	$\lambda^{(4)}$		$H^{(1)}$			$H^{(2)}$			$H^{(3)}$			$H^{(4)}$		
						$h_1$	$h_2$	$h_3$	$h_4$	$h_5$	$h_6$	$h_7$	$h_8$	$h_9$	$h_{10}$	$h_{11}$	
$x_1$	1	2	1	1	$\Leftrightarrow$	$v_1$	1	0	0	0	1	0	1	0	0	1	0
$x_2$	1	2	1	2		$v_2$	1	0	0	0	1	0	1	0	0	0	1
$x_3$	1	2	2	?		$v_3$	1	0	0	0	1	0	0	1	0	0	0
$x_4$	2	3	2	1		$v_4$	0	1	0	0	0	1	0	1	0	1	0
$x_5$	2	3	3	2		$v_5$	0	1	0	0	0	1	0	0	1	0	1
$x_6$	3	1	3	?		$v_6$	0	0	1	1	0	0	0	0	1	0	0
$x_7$	3	1	3	?		$v_7$	0	0	1	1	0	0	0	0	1	0	0

A partir de la construcción de dicho hipergrafo se definen las 3 heurísticas siguientes para darle solución al problema:

(CSPA) Algoritmo de particionamiento basado en similaridad de agrupaciones.

Esencialmente, si dos objetos están en el mismo agrupamiento, éstos son considerados como totalmente similares, sino, son completamente disimilares. Basándose en este presupuesto, se define como similaridad entre dos estructuraciones, a la fracción de las estructuraciones en la cual dos objetos están en el mismo agrupamiento en ambas. La matriz de similaridad  $S$  puede obtenerse a partir de la siguiente multiplicación de matrices esparcidas  $S = \frac{1}{m} H \cdot H^T$ . Luego se usa  $S$  para volver a agrupar los objetos, en este caso se obtiene primero el grafo inducido  $G$  en el cual se toma (*vértices = objetos; peso de las aristas = similaridad*) y se aplica el algoritmo de particionamiento de grafos METIS [61].

(HGPA) Algoritmo de particionamiento de hipergrafos.

El problema de la combinación de agrupamientos es reformulado como un problema de particionamiento de un hipergrafo cortando el menor número de hiperaristas posible. Se considera que todas las hiperaristas tienen el mismo peso al igual que los vértices, y se busca cortar el menor número de hiperaristas que particione el hipergrafo en  $k$  componentes conexas de aproximadamente el mismo tamaño. Para la implementación del método se usó el paquete de particionamiento de hipergrafos HMETIS [62].

(MLCA) Algoritmo Meta-Clustering.

En este método, primero se trata de resolver el problema de la correspondencia de las etiquetas de los agrupamientos y entonces usar un mecanismo de votación para ubicar los objetos en los agrupamientos de la estructuración de consenso final.

Primeramente se construye la matriz  $H$  como se vio anteriormente, en la cual se representa cada agrupamiento como un vector binario de longitud  $n$ . La similaridad entre dos agrupamientos  $C_i$  y  $C_j$  es calculado en base a la cantidad de objetos que son agrupados en ambos, usando el índice de Jaccard (epígrafe 5.1.2). Luego se forma la matriz de similaridad entre los agrupamientos, la cual representa la matriz de adyacencia del grafo formado, tomando a los agrupamientos como nodos y asignándole un peso a la arista entre dos nodos igual a la similaridad entre los agrupamientos. Después este grafo es particionado utilizando el algoritmo METIS [61] y los agrupamientos obtenidos son llamados meta-clusters. Finalmente para hallar la partición final, se calcula la cantidad de veces que cada objeto está en un meta-cluster, y se asigna dicho objeto al meta-cluster al cual es asignado un mayor número de veces. Este algoritmo es cuadrático respecto a la cantidad de agrupamientos en el conjunto de particiones  $\mathbb{P}$ , el cual en la mayoría de las aplicaciones es significativamente menor que  $n^2$ , por tanto este algoritmo es bastante eficiente.

Características

- En este trabajo se define formalmente el problema de encontrar una partición de consenso como un problema de optimización a partir del cálculo de la información compartida entre dos particiones utilizando la información mutua.
- A pesar de ser la más usada, la información mutua entre particiones es una medida de similaridad entre las mismas, en la literatura existen otras medidas de similaridad entre particiones (epígrafe 5), las cuales podrían ser utilizadas para la definición del problema de optimización (1).
- La solución teórica propuesta exige la resolución de un problema de optimización para el cual no se conoce otra solución que la exponencial (analizando todas las posibles particiones y quedarse con la que maximice la ecuación (1)), sin embargo, se proponen 3

heurísticas para encontrar la partición de consenso, las cuales no están relacionadas directamente con el problema de optimización propuesto.

- De las tres heurísticas propuestas, la primera, *CSPA*, tiene una complejidad cuadrática por tanto está limitada para trabajar con grandes volúmenes de información.

#### 6.2.4 Combinación de Agrupamientos basada en Teoría de la Información

Es un tipo de función de consenso basado en la noción de la partición media. La partición media  $P^*$  es la *mejor* síntesis de las particiones en  $\mathbb{P}$ . A diferencia del enfoque de co-asociación, la búsqueda de la partición media tiene sus bases en la estimación de similaridades entre las particiones en vez de similaridades entre los objetos.

En [12] se sugiere (como vimos en el epígrafe 6.2.3) una función objetivo basado en la información mutua, donde la partición media puede obtenerse resolviendo el problema de optimización (1), sin embargo, esta expresión no ofrece ninguna idea de cómo implementar un algoritmo para obtener la solución óptima.

En [38] se define la función de utilidad  $U(\sigma, P_i)$ , la cual evalúa la calidad de una partición media candidato  $P_c = \{C_1, C_2, \dots, C_k\}$  contra otra partición  $P_i = \{L_1^i, L_2^i, \dots, L_{k(i)}^i\}$  con etiquetas  $L_j^i$  en el  $j$ -ésimo cluster:

$$U(P_c, P_i) = \sum_{r=1}^k p(C_r) \sum_{j=1}^{k(i)} p(L_j^i | C_r)^2 - \sum_{j=1}^{k(i)} p(L_j^i)^2$$

$$\text{donde } p(C_r) = \frac{|C_r|}{n}, p(L_j^i) = \frac{|L_j^i|}{n} \text{ y } p(L_j^i | C_r) = \frac{|L_j^i \cap C_r|}{|C_r|}.$$

La función  $U(P_c, P_i)$  mide el acuerdo entre dos particiones a partir de la diferencia entre el número esperado de etiquetas asignadas a la partición  $P_i$  que pueden ser predichas conociendo la distribución de las etiquetas en la partición  $P_c$  y sin conocer la distribución de las etiquetas en  $P_c$ . La utilidad conjunta de la partición  $P_c$  respecto a todas las particiones en la combinación de agrupamientos  $\mathbb{P}$  puede ser calculada a partir de la suma de las funciones individuales:

$$U(P_c, \mathbb{P}) = \sum_{j=1}^m U(P_c, P_j)$$

Por tanto la mejor partición media debe maximizar el valor de la utilidad conjunta:

$$P^* = \arg \max_{P_c} U(P_c, \mathbb{P})$$

En [63] se prueba que la maximización de la función de utilidad es equivalente a minimizar el error cuadrático del proceso de agrupamiento si es conocido el valor  $k$  (cantidad de agrupamientos en la partición). Este problema puede resolverse haciendo la siguiente transformación: primeramente, para cada objeto  $O$  se calculan los valores de nuevas características  $y_{ij}(O)$ , para todo  $j = 1 \dots k(i)$ ,  $i = 1 \dots m$  de la siguiente manera:

$$y_{ij}(O) = \delta(L_j^i, P_i(O)) - p(L_j^i)$$

donde  $P_i(O)$  representa la etiqueta asociada al objeto  $O$  en la partición  $P_i$  y

$$\delta(a, b) = \begin{cases} 1, & \text{si } a = b \\ 0, & \text{si } a \neq b \end{cases}$$

Entonces la partición media puede ser obtenida utilizando el algoritmo k-Means operando sobre el nuevo espacio de características  $y_{ij}$ .

En el contexto de la teoría de la información, la calidad de la partición de consenso  $P^*$  puede ser determinada por la cantidad de información  $I(P^*, \mathbb{P})$  que ésta comparte con el conjunto de particiones  $\mathbb{P}$ . En este método se utiliza otra definición del concepto de entropía, con la cual se reduce la definición de información mutua a la función de utilidad definida anteriormente.

Partiendo de la entropía generalizada de grado  $s$ , la información mutua normalizada entre  $P^a$  y  $P^b$  es definida como:

$$I^s(P^a, P^b) = H^s(P^a) - H^s(P^a | P^b)$$

donde  $H$  representa la entropía. Usando la entropía cuadrática ( $s = 2$ ), la información mutua normalizada se convierte en:

$$NMI^2(P^*, P_i) = -2 \left( \sum_{j=1}^{k(i)} p(L_j^i)^2 - 1 \right) + 2 \sum_{r=1}^k p(C_r) \left( \sum_{j=1}^{k(i)} p(L_j^i | C_r)^2 - 1 \right) = 2U(P^*, P_i)$$

Por tanto la información mutua normalizada ofrece el mismo criterio de consenso que la función de utilidad.

#### Características

- La contribución principal de este método es la incorporación de una vía de solución al problema de la partición media. Anteriormente se había definido como partición de consenso a la partición que más información comparte con el conjunto de particiones en la combinación de agrupamientos, esta información era medida utilizando la información mutua normalizada, sin embargo el problema de optimización planteado no ofrecía ningún algoritmo de solución, había que recurrir a diferentes heurísticas que garantizaran un acercamiento a la solución. En este enfoque se demuestra que la información mutua normalizada es equivalente a la función de utilidad, la cual permite llegar a la partición media a partir de la transformación del problema en uno de minimización de la varianza inter-agrupamientos, que puede ser resuelto utilizando el algoritmo k-Means.
- Es un algoritmo bastante eficiente, su costo computacional es  $O(knm)$ .
- Sin embargo, en ocasiones, según los propios autores, necesita ser ejecutado varias veces para evitar la convergencia a un mínimo local que provoque baja calidad en los resultados.

#### 6.2.5 Modelos de mezclas (EM)

En [39] se propone una nueva función de consenso que viene dada por la solución de un problema de máxima verosimilitud (*likelihood*) para un conjunto de particiones dado. La función de verosimilitud de la combinación de particiones es optimizada respecto a los parámetros de una distribución de mezcla finita. El problema de máxima verosimilitud es resuelto usando el algoritmo EM [64].

Este enfoque de consenso está basado en un modelo de mezclas finitas para la probabilidad de asignación de etiquetas a los objetos, en las particiones. La principal asunción es que las etiquetas  $y_i$  (etiqueta asignada al objeto  $O_i$ ) son modeladas como variables aleatorias a partir de una distribución de probabilidad descrita como una mezcla de densidades multivariadas:

$$\rho(y_i|\Theta) = \sum_{t=1}^T \alpha_t \rho_t(y_i|\theta_t) \quad (2)$$

donde cada componente es parametrizada por  $\theta_t$ . Las  $T$  componentes en la mezcla son identificadas con los agrupamientos en la partición de consenso  $P^*$ . Los coeficientes  $\alpha_t$  corresponden a las probabilidades a priori de los agrupamientos. Para los datos  $Y = \{y_i\}_{i=1}^n$  se asumen que son independientes e idénticamente distribuidos.

Esto permite representar la función de verosimilitud logarítmica para los parámetros  $\Theta = \{\alpha_1, \dots, \alpha_T, \theta_1, \dots, \theta_T\}$  dado el conjunto de datos  $Y$  como:

$$\log L(\Theta|Y) = \log \prod_{i=1}^n \rho(y_i, \Theta) = \sum_{i=1}^n \log \sum_{t=1}^T \alpha_t \rho_t(y_i|\theta_t) \quad (3)$$

La búsqueda de la partición de consenso es formulada como un problema de estimación de máxima verosimilitud:

$$\Theta^* = \arg \max_{\Theta} \log L(\Theta|Y) \quad (4)$$

El problema de máxima verosimilitud en la ecuación (3) generalmente no puede ser resuelto de forma cerrada (sólo en términos de funciones y operaciones elementales) cuando los parámetros  $\Theta$  son desconocidos. Sin embargo, la función de verosimilitud (2) puede ser optimizada usando el algoritmo EM, asumiendo la existencia de los datos ocultos  $Z$  y la verosimilitud de los datos completos  $(Y, Z)$ . Para esto se comienza con unos parámetros iniciales arbitrarios  $\{\alpha'_1, \dots, \alpha'_T, \theta'_1, \dots, \theta'_T\}$  después son repetidos mientras se cumpla un criterio de convergencia los pasos E y M, donde el paso E calcula los valores esperados de las variables ocultas  $E[z_{it}]$  y el paso M maximiza la verosimilitud calculando una nueva y mejor estimación de los parámetros. Los criterios de convergencia pueden estar dados por el valor de crecimiento de la verosimilitud entre dos pasos M consecutivos.

La partición de consenso es obtenida inspeccionando los valores esperados de las variables  $E[z_{it}]$  debido a que  $E[z_{it}]$  representa la probabilidad de que el patrón  $y_i$  haya sido generado por la componente de mezcla  $m$ -ésima. Cuando se alcanza algún criterio de convergencia, la etiqueta  $y_i$  es asignada a la componente que tenga mayor valor de la variable oculta  $z_i$ .

**Tabla 5.** Pseudocódigo del algoritmo

<p><u>Entrada:</u> Conjunto de objetos y conjunto de particiones de los objetos.  <u>Salida:</u> Partición de consenso <math>P^*</math>  Inicializar los parámetros del modelo <math>\Theta = \{\alpha_1, \dots, \alpha_T, \theta_1, \dots, \theta_T\}</math>  Repetir hasta que se cumpla algún criterio de convergencia:      Calcular los valores esperados <math>E[z_{it}]</math>, <math>i = 1 \dots n</math>, <math>t = 1 \dots T</math>      Calcular los valores esperados para las variables con ausencia de información (si hay)      Re-estimar los parámetros.  <math>P^*(O_i)</math>=índice de la componente de <math>z_i</math> con mayor valor esperado.</p>
--

### Características

- Baja complejidad computacional  $O(knm)$ , producto del uso del algoritmo EM cuya complejidad es comparable con la del algoritmo k-Means.
- La eficiencia de este algoritmo permite que pueda ser aplicado sobre grandes conjuntos de datos, a diferencia de otras funciones de consenso que tienen una complejidad cuadrática (ejemplo las basadas en co-asociación).
- Posibilidad de trabajar con datos incompletos (*missing values*), en este caso sería la falta de asignación de alguna etiqueta de agrupamiento para algún elemento en el conjunto de objetos.

### 6.2.6 Basados en Algoritmos Genéticos

En [59] se propone un método para la combinación de resultados de clasificadores no supervisados llamado HCE (*heterogeneous clustering ensemble*, por sus siglas en inglés) basado en el uso de un algoritmo genético para la combinación de particiones. El método consta de dos fases, un mecanismo de reproducción y uno de cruzamiento.

Primero en este algoritmo es necesaria la creación de una población inicial que funcione como punto de partida del algoritmo genético. Para esto se aplican diferentes algoritmos de agrupamientos (en general es posible cualquier variante de generación de agrupamientos, ver epígrafe 6.1). Con cada par de estructuraciones obtenidas de los objetos se forma un par ordenado. Por ejemplo, si se tienen 4 estructuraciones ( $P_1, P_2, P_3, P_4$ ) se forman 12 pares iniciales (ver Fig. 5)

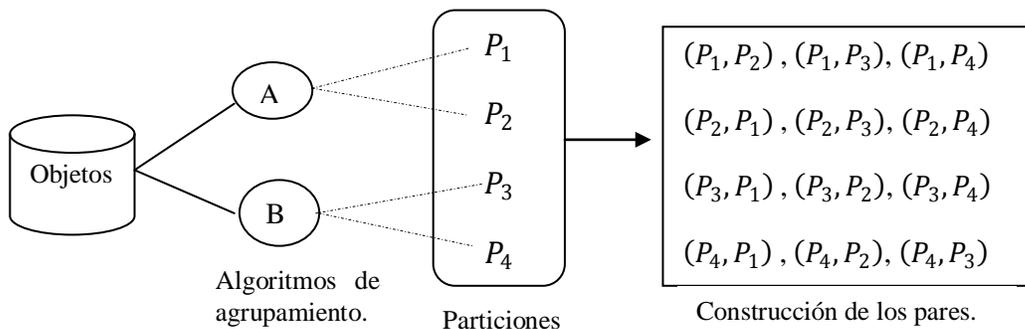


Fig. 5 Generación de la población inicial

El proceso de reproducción usa una función de ajuste como única vía para determinar si un par de estructuraciones (cromosoma) sobrevivirá o no sobrevivirá en la próxima etapa. En este algoritmo la función de ajuste viene dada por la comparación de la cantidad de solapamientos que hay entre los agrupamientos de las particiones en cada cromosoma.

Por ejemplo, si se tienen los objetos  $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$  y las siguientes particiones de los mismos:  $P_1 = \{(x_1, x_2, x_3) (x_4, x_5, x_6) (x_7, x_8, x_9, x_{10})\}$  y  $P_2 = \{(x_1, x_2) (x_3, x_4) (x_5, x_6) (x_7, x_8, x_9, x_{10})\}$ , la función de ajuste funcionaria de la siguiente manera:

Para cada agrupamiento en la primera partición se busca el agrupamiento con el que más objetos comparte en la segunda partición, en este caso se escoge  $(x_1, x_2, x_3)$  de la primera partición y  $(x_1, x_2)$  de la segunda partición, dando un valor de solapamiento igual a 2. Así

mismo se escogen a  $(x_4, x_5, x_6)$  y  $(x_5, x_6)$ , dando un valor de solapamiento igual a 2 y  $(x_7, x_8, x_9, x_{10})$  con  $(x_7, x_8, x_9, x_{10})$ , con solapamiento igual a 4. Por tanto la comparación de  $P_1$  con  $P_2$  da un valor igual a  $2+2+4 = 8$ . Haciendo el mismo análisis pero ahora calculando el solapamiento de  $P_2$  con  $P_1$  se obtiene un valor igual a  $2+1+2+4=9$ . Luego la función de ajuste entre estas dos particiones es igual a  $8+9 = 17$ .

Esta función da un valor representativo a cada par de particiones, y el par que mayor valor tenga se le aplica el cruzamiento.

En el proceso de cruzamiento se obtienen descendientes a partir del par seleccionado manteniendo la mayor cantidad de información posible de los padres en las nuevas particiones obtenidas. Posteriormente las particiones padres son reemplazadas por sus descendientes y se aplica otra iteración del algoritmo completo. En la Tabla 5 se muestra el pseudocódigo del algoritmo.

**Tabla 6.** Pseudocódigo del algoritmo HCE

Algoritmo (HCE)
<p><b>Entrada:</b></p> <ol style="list-style-type: none"> <li>(1) El conjunto de n objetos.</li> <li>(2) El conjunto de algoritmos</li> </ol> <p><b>Salida:</b> Combinación de los agrupamientos.</p> <ol style="list-style-type: none"> <li>1. Correr los diferentes algoritmos de agrupamiento y obtener las particiones de los objetos.</li> <li>2. Construir un subconjunto no vacío de pares <math>SM^{(g)}</math>, para cada agrupamiento S.</li> <li>3. Iterar hasta que converja (permutando los pares):             <ol style="list-style-type: none"> <li>3.1. Calcular la función de ajuste <math>F(t)</math> y seleccionar dos padres de <math>\square\square^{(g)}</math>.</li> <li>3.2. Cruzar a los dos padres.</li> <li>3.3. Reemplazar el subconjunto <math>\square\square^{(g)}</math> de los padres por los nuevos creados para la descendencia.</li> </ol> </li> </ol>

### Características

- No se define formalmente el problema que se va a resolver, es decir, no se dice qué características debe tener la partición que se está buscando, ni se formula matemáticamente el problema, sólo se presenta el algoritmo genético y no se fundamenta por qué dicho algoritmo se acerca a la partición de consenso.
- Si se cuenta con un gran número de particiones de los datos, el conjunto de pares de particiones sería muy grande y esto afectaría la eficiencia del algoritmo.

### 6.2.7 Basados en agrupamientos localmente adaptables

Este es un tipo de función de consenso que necesita de un mecanismo de generación específico. En [60] se proponen tres tipos de funciones de consenso, las cuales están basadas en la utilización de particiones de los objetos obtenidas a partir de la utilización de algoritmos de agrupamiento localmente adaptables (LAC, por sus siglas en inglés) [65, 66]. Estos algoritmos de agrupamiento dan como resultado una partición dura de los objetos y un vector de peso asociado a cada agrupamiento, donde el valor  $i$ -ésimo de este vector se corresponde con el peso asociado al rasgo  $i$ -ésimo de la representación de los objetos. Es decir, cada valor de peso está midiendo la

importancia de cada rasgo, en término de los cuales son representados los objetos, para el agrupamiento en cuestión.

La salida de los algoritmos de agrupamiento LAC, pueden ser identificados por dos conjuntos  $\{c_1, \dots, c_k\}$  y  $\{w_1, \dots, w_k\}$ . Cambiando los parámetros de los LAC se genera el conjunto de particiones de los datos y el conjunto de pesos asociados a cada partición. Por tanto es necesaria una función de consenso capaz de trabajar con toda esta información, las estructuraciones y sus pesos asociados.

El primer algoritmo propuesto es nombrado algoritmo de particionamiento de similitud pesada, (WSPA por sus siglas en inglés). Para cada objeto  $O_i$  la distancia pesada a un agrupamiento  $c_t$  viene dada por:

$$d_{it} = \sqrt{\sum_{s=1}^l w_{ts} (O_{is} - c_{ts})^2} \quad (5)$$

donde  $O_{is}$  representa la componente  $s$ -ésima del  $i$ -ésimo objeto, representando cada objeto a partir de  $l$  rasgos y  $c_t$  es el centroide del  $t$ -ésimo agrupamiento.

Sea  $D_i = \max_t \{d_{it}\}$  la máxima distancia de  $O_i$  a todos los agrupamientos. Asumiendo que las etiquetas  $C_t$  son una variable aleatoria, con probabilidades  $\{P(C_t|O_i)\}_{t=1}^k$ , se define  $P(C_t|O_i) = \frac{D_i - d_{it} + 1}{kD_i + k - \sum_t d_{it}}$ , donde el denominador sirve como un factor de normalización que garantiza que  $\{P(C_t|O_i)\}_{t=1}^k = 1$ . Mediante esta ecuación se puede construir el vector de probabilidades posteriores asociado al objeto  $O_i$ :

$$P_i = (P(C_1|O_i), P(C_2|O_i), \dots, P(C_k|O_i))'$$

La transformación  $O_i \rightarrow P_i$ , mapea los objetos  $l$  dimensionales en un espacio de coordenadas relativas a los centroides de los agrupamientos. Para calcular la similitud entre dos objetos  $O_i, O_j$  se utiliza la similitud del coseno entre los vectores asociados a dichos objetos, o sea:

$$s(O_i, O_j) = \frac{P_i^t P_j}{\|P_i\| \|P_j\|}$$

donde  $t$  denota la transpuesta del vector.

Posteriormente se combinan todas las similitudes entre objetos en una matriz  $S$  de  $n \times n$ , donde  $S_{ij} = s(O_i, O_j)$ . Después de correr el algoritmo LAC  $m$  veces utilizando diferentes inicializaciones de los parámetros, utilizando este mismo procedimiento se obtienen  $m$  matrices de similitud  $S_1, S_2, \dots, S_m$ . Sea  $\Psi$  la combinación de estas matrices de la siguiente manera:

$$\Psi = \frac{1}{m} \sum_{p=1}^m S_p$$

donde  $\Psi_{ij}$  (valor en la posición  $i, j$  de la matriz  $\Psi$ ) refleja el promedio de la similitud entre los objetos  $O_i, O_j$  (a través de los distintos valores de  $P_i, P_j$  para las  $m$  particiones de los datos). Como último paso, se asocia el problema de encontrar la partición de consenso con el particionamiento de un grafo. Para esto se construye el grafo  $G = (V, E)$  donde cada vértice  $v_i$  se identifica con un objeto  $O_i$  y la arista  $E_{ij}$  que conecta los vértices  $v_i$  y  $v_j$  tiene un valor

asociado igual a  $\Psi_{ij}$ . Para particionar este grafo se usa el algoritmo METIS [61]. Un resumen de los pasos del algoritmo de particionamiento de similaridad pesada se muestra en Tabla 7.

**Tabla 7.** Pseudocódigo del algoritmo WSPA

Algoritmo (WSPA)
<p><b>Entrada:</b>  <math>\{C_1^v, C_2^v, \dots, C_m^v\}, \{w_1^v, w_2^v, \dots, w_m^v\} \rightarrow m</math> particiones de los objetos usando algoritmos de agrupamiento localmente adaptables (LAC).</p> <p><b>Salida:</b> Estructuración de consenso.</p> <ol style="list-style-type: none"> <li>1. Para cada partición <math>v = 1, \dots, m</math>: <ol style="list-style-type: none"> <li>a) Calcular la similaridad máxima entre cada objeto y el conjunto de agrupamientos <math>D_i^v</math>.</li> <li>b) Calcular <math>P_i^v = (P(C_1^v O_i), P(C_2^v O_i), \dots, P(C_m^v O_i))^t</math> vector representativo de cada objeto.</li> <li>c) Construir la matriz <math>S^v</math> donde <math>S_{ij}^v = \text{similaridad}(O_i, O_j)</math> utilizando la similaridad del coseno sobre los vectores representativos de cada objeto.</li> </ol> </li> <li>2. Construir la matriz <math>\Psi = \frac{1}{m} \sum_{v=1}^m S^v</math>.</li> <li>3. Construir el grafo <math>G = (V, E)</math> donde <math>V_i = O_i</math> y asignándole <math>\Psi_{ij}</math> a la arista <math>E_{ij}</math> (arista que une los vértices <math>v_i, v_j</math>).</li> <li>4. Aplicar el algoritmo METIS (o algoritmos de agrupamiento espectrales), para obtener el particionamiento del grafo <math>G</math>.</li> </ol>

El segundo método llamado algoritmo de particionamiento bipartito pesado (WBPA, por sus siglas en inglés), como su nombre indica, brinda una función de consenso basada en el particionamiento de un grafo bipartito. Supongamos nuevamente que se aplican  $m$  veces el algoritmo de agrupamiento LAC, con diferentes valores en los parámetros. Para cada objeto  $O_i$  y para cada estructuración  $v = 1, \dots, m$  se calcula el vector de probabilidades posteriores  $P_i^v$ , siguiendo los mismos pasos que en el algoritmo WSPA. Usando este vector se construye la matriz  $A$ :

$$A = \begin{pmatrix} (P_1^1)^t & \dots & (P_1^m)^t \\ \vdots & \ddots & \vdots \\ (P_n^1)^t & \dots & (P_n^m)^t \end{pmatrix} \quad (6)$$

donde  $(P_i^v)$  son los vectores filas. La dimensionalidad de  $A$  es  $n \times km$ , asumiendo que cada uno de los  $m$  algoritmos aplicados produce  $k$  agrupamientos de los objetos. Basado en  $A$  se define un grafo bipartito el cual será particionado para obtener la estructuración de consenso.

Sea el grafo  $G = (V, E)$ , con  $V = V^c \cup V^l$ , donde  $V^c$  contiene  $km$  vértices, cada uno representando un agrupamiento y  $V^l$  contiene  $n$  vértices, cada uno asociado a un objeto. Las aristas van a tener un peso asociado de la siguiente manera:

- a)  $E(i, j) = 0$ , En el caso en que ambos vértices  $v_i, v_j$ , representen objetos o ambos representen agrupamientos.

b)  $E(i, j) = A(i - km, j)$ , En el caso que  $v_i$  representa un objeto y  $v_j$  representa un agrupamiento.

c)  $E(i, j) = E(j, i)$  Cuando  $v_i$  representa un agrupamiento y  $v_j$  representa un objeto.

Una partición del grafo bipartito  $G$  agruparía objetos y agrupamientos simultáneamente, y la partición de consenso puede ser obtenida a partir del particionamiento de los objetos. Teniendo en cuenta de que este grafo es esparcido (la mayor parte de las aristas tienen valor cero), el algoritmo sería de complejidad  $kmn$  que asumiendo que  $km \ll n$  es más eficiente que el WSPA que es de orden  $n^2$ .

**Tabla 8.** Pseudocódigo del algoritmo WBPA

Algoritmo (WBPA)
<p><b>Entrada:</b>  <math>\{C_1^v, C_2^v, \dots, C_m^v\}, \{w_1^v, w_2^v, \dots, w_m^v\} \rightarrow m</math> particiones de los objetos usando algoritmos de agrupamiento localmente adaptables (LAC).</p> <p><b>Salida:</b> Estructuración de consenso.</p> <ol style="list-style-type: none"> <li>1. Para cada partición <math>v = 1, \dots, m</math>: <ol style="list-style-type: none"> <li>a) Calcular la similaridad máxima entre cada objeto y el conjunto de agrupamientos <math>D_i^v</math>.</li> <li>b) Calcular <math>P_i^v = (P(C_1^v O_i), P(C_2^v O_i), \dots, P(C_m^v O_i))^t</math> vector representativo de cada objeto.</li> </ol> </li> <li>2. Construir la matriz <math>A</math> (ecuación (6)).</li> <li>3. Construir el grafo bipartito <math>G = (V, E)</math> de objetos y agrupamientos.</li> <li>4. Aplicar el algoritmo METIS (o algoritmos de agrupamiento espectrales), para obtener el particionamiento del grafo <math>G</math>.</li> </ol>

El tercer método propuesto llamado algoritmo de particionamiento bipartito de subespacios pesados (WSBPA, por sus siglas en inglés) a diferencia de los dos métodos anteriores produce una partición de consenso con un peso asociado a cada agrupamiento. Básicamente este algoritmo se diferencia del WBPA en la adición de un vector de pesos a cada agrupamiento en la partición final.

Partiendo del grafo bipartito  $G = (V, E)$ , construido de manera similar al algoritmo WBPA. Después de aplicar un algoritmo de particionamiento de grafos, la partición  $P_{V^I}$  del conjunto  $V^I$  (vértices asociados a objetos), da la partición de consenso. Por otra parte, también se obtiene la partición  $P_{V^c}$  del conjunto  $V^c$  (vértices asociados a agrupamientos), cada elemento  $V_1^c$  en  $P_{V^c}$  es un conjunto de vértices de agrupamientos, los cuales tienen asociados un vector de peso. Sea  $w_{i_1}^c$  el vector de pesos asociado al vértice de agrupamiento  $v_{i_1}^c$ , entonces se promedian todos los vectores de pesos  $w_{i_1}^c$ , con  $i = 1, \dots, |V_1^c|$  para obtener los pesos para los agrupamientos  $V_1^I$  de la partición de consenso. Es decir, se obtiene como resultado de este algoritmo  $k$  agrupamientos con sus respectivos  $k$  vectores de pesos asociados  $\{(V_1^I, w_1)\}_{i=1}^k$  donde  $w_1 = \frac{1}{|V_1^c|} \sum_{i=1}^{|V_1^c|} w_{i_1}^c$ .

**Tabla 9.** Pseudocódigo del algoritmo WSBPA

Algoritmo (WSBPA)
<p><b>Entrada:</b>  <math>\{C_1^v, C_2^v, \dots, C_m^v\}, \{w_1^v, w_2^v, \dots, w_m^v\} \rightarrow m</math> particiones de los objetos usando algoritmos de agrupamiento localmente adaptables (LAC).</p> <p><b>Salida:</b> Estructuración de consenso con un vector de peso asociado a cada agrupamiento.</p> <ol style="list-style-type: none"> <li>1. Para cada partición <math>v = 1, \dots, m</math>: <ol style="list-style-type: none"> <li>a) Calcular la similaridad máxima entre cada objeto y el conjunto de agrupamientos <math>D_i^v</math>.</li> <li>b) Calcular <math>P_i^v = (P(C_1^v O_i), P(C_2^v O_i), \dots, P(C_m^v O_i))^t</math> vector representativo de cada objeto.</li> </ol> </li> <li>2. Construir la matriz <math>A</math> (ecuación (6)).</li> <li>3. Construir el grafo bipartito <math>G = (V, E)</math> de manera similar a como se hace en el algoritmo WBPA.</li> <li>4. Aplicar el algoritmo METIS (o algoritmos de agrupamiento espectrales), para obtener el particionamiento del grafo <math>G</math>.</li> <li>5. Calcular el vector de pesos promedio <math>w_l</math> asociado a la partición de consenso obtenida.</li> </ol>

### Características

- La función de consenso está muy ligada al mecanismo de generación, sólo funciona si las estructuraciones iniciales de los datos son obtenidas a partir de la aplicación del algoritmo de agrupamiento localmente adaptable.
- Este tipo de función de consenso sólo es capaz de trabajar con conjuntos de objetos representados en espacios métricos, ya que en un paso intermedio del algoritmo es necesario el cálculo de la distancia euclidiana entre cada objeto y los centroides de los agrupamientos.
- Se necesita saber la cantidad de agrupamientos que va a tener la partición de consenso, ya que todas las particiones de los datos a combinar deben tener la misma cantidad de agrupamientos.
- El primer algoritmo WSPA tiene una complejidad  $O(n^2)$ , lo que no permite que sea utilizado en problemas donde se manipulen grandes volúmenes de datos.
- El segundo algoritmo WBPA tiene una complejidad  $O(kmn)$ , por tanto es más eficiente que el algoritmo WSPA siempre que  $km \ll n$ .
- El tercer algoritmo WSBPA es muy parecido al segundo, la estructuración de consenso va a ser igual a la obtenida por el algoritmo WBPA, sin embargo el WSBPA le añade a los resultados obtenidos un vector de peso asociado a cada agrupamiento, que es una información adicional que podría ser aprovechada en la solución de algún problema.

### 6.2.8 Basadas en funciones kernel para grafos

En [40] se propone la incorporación de un paso intermedio en la metodología tradicional de la combinación de resultados de algoritmos de agrupamientos. Se propone estimar la importancia de cada partición en un paso anterior a la combinación de las diferentes estructuraciones.

En este trabajo se refleja la importancia de cada partición a partir de la asignación de un peso a cada una, de acuerdo a un análisis del cumplimiento o no, de un conjunto de propiedades relevantes para el proceso de combinación. Para calcular los pesos asociados a cada partición se prosigue de la manera siguiente:

Primero se define un conjunto de índices de validación de propiedades, estos índices se definen como una función  $I: \mathbb{P}_o \rightarrow [0,1]$ , donde  $\mathbb{P}_o$  es el conjunto de todas las posibles particiones a formar con los objetos. La idea es que cada índice mida el cumplimiento de una propiedad en particular por cada una de las particiones, de manera que el valor obtenido al evaluar un índice en una partición representa el grado de cumplimiento de la propiedad representada por el índice, por dicha partición.

Sea  $\mathbb{I} = \{I_1, I_2, \dots, I_s\}$  el conjunto de índices definidos, para determinar el vector de pesos asociado a conjunto de particiones. Para cada índice  $I_j$  se calcula  $S_j = \sum_{i=1}^m I_j(P_i)$ . Si se define la función  $\varphi_j: \mathbb{P}_o \rightarrow [0,1]$  tal que  $\varphi_j(P_i) = \frac{I_j(P_i)}{S_j}$ . Como  $\sum_{i=1}^m \varphi_j(P_i) = 1$ , cada función  $\varphi_j$  puede ser relacionada con la función de distribución de cierta variable aleatoria  $Y_j$ . Luego se define:

$$H(I_j) = H(Y_j) = - \sum_{i=1}^m \varphi_j(P_i) \cdot \log(\varphi_j(P_i))$$

donde  $H(Y_j)$  es la entropía de  $Y_j$  [43]. Utilizando las propiedades de la entropía, se llega a la conclusión de que mayores valores de  $H(I_j)$  implica mayor parecido entre los valores  $I_j(P_i)$ , es decir, el índice  $I_j$  tiene un comportamiento más similar en todas las particiones de la combinación. Siguiendo el criterio de que las propiedades que más información brindan del conjunto de particiones son aquellas que son cumplidas de manera similar por un gran número de particiones, ya que permite detectar particiones ruidosas que tengan un comportamiento diferente a la mayoría de las particiones. Utilizando  $H(I_j)$  se llega a la expresión siguiente para determinar el peso asociado a cada partición:

$$\lambda_i = \sum_{j=1}^m H(I_j) \cdot \left( 1 - \left| I_j(P_i) - \frac{1}{m} S_j \right| \right)$$

donde el segundo factor de esta expresión es una comparación entre los valores  $I_i(P_j)$  y el valor medio  $\frac{1}{m} S_j$ .

La función de consenso propuesta en este trabajo está basada en la idea de la búsqueda de la partición media, sin embargo, al tener el conjunto de pesos asociado al conjunto de particiones, se hace una búsqueda de la partición media pesada, la cual puede ser escrita de la manera siguiente:

$$P^* = \arg \max_{P \in \mathbb{P}_o} \sum_{i=1}^m \lambda_i \cdot \Gamma(P, P_i) \quad (7)$$

donde  $\Gamma$  es una medida de similitud entre particiones. En este trabajo [40] se utiliza como medida de similitud entre particiones la similitud basada en kernel para grafos (ver epígrafe

5.1.4), donde cada partición es representada de manera única mediante un grafo, es decir, existe un biyección entre el conjunto de las particiones formadas por los objetos,  $\mathbb{P}_o$  y el conjunto de todos los grafos que son representaciones de particiones,  $\mathbb{G}_o$ . Esta similaridad es calculada mediante la expresión:

$$\Gamma(G, G') = \sum_{(h \in \Sigma(G))} \sum_{(h' \in \Sigma(G'))} k_\delta(h = h') p(h|G) p(h'|G')$$

donde  $G$  y  $G'$  son grafos asociados a particiones y  $h, h'$  son caminos sobre  $G$  y  $G'$  respectivamente,  $p(h|G)$  representa la probabilidad de encontrar el camino  $h$  en el grafo  $G$  que para este tipo de grafo tiene un valor  $p(h|G) = \frac{1}{n|C_h|^{l-1}}$ , donde  $C_h$  es la componente conexa que contiene al camino  $h$  y  $l$  es la longitud de dicho camino.

En [40] se demuestra que esta función es un kernel, por tanto existe un espacio de Hilbert  $\mathcal{H}$  y una función  $\phi: \mathbb{G}_o \rightarrow \mathcal{H}$ , tal que  $\forall G, G' \in \mathbb{G}_o, \Gamma(G, G') = \langle \phi(G), \phi(G') \rangle_{\mathcal{H}}$ . Luego el problema de optimización (7) tiene una solución única  $\psi$ , en el Espacio de Hilbert de Kernel Reproductores (RKHS, siglas en inglés) [45] asociado a la función  $\Gamma$ , con una representación de la forma  $\psi = \sum_{i=1}^m \beta_i \phi(G_i)$ , con  $\beta_i = \frac{\lambda_i}{\sum_{j=1}^m \lambda_j}$ .

Sin embargo, para obtener el grafo que representa a la partición de consenso es necesario resolver el problema de la pre-imagen  $G^* = \phi^{-1}(\psi)$ . Esto es, encontrar un grafo cuya representación en el espacio  $\mathcal{H}$  a través de  $\phi$  sea igual a la solución  $\psi$ . Encontrar una solución exacta de este problema es una tarea muy compleja, por lo que se escoge como solución aproximada:

$$G^* = \arg \min_{G \in \mathbb{G}_o} \|\psi - \phi(G)\|^2 \quad (8)$$

donde  $\|\psi - \phi(G)\|^2 = \Gamma(G, G) - 2 \sum_{i=1}^m \beta_i \Gamma(G, G_i) + \sum_{i=1}^m \sum_{j=1}^m \beta_i \beta_j \Gamma(G_i, G_j)$ .

Para resolver este problema en este trabajo se siguió una estrategia introducida en [67], basada en una búsqueda estocástica para encontrar la matriz de adyacencia del grafo solución, que representa a la partición de consenso.

### Características

- El análisis del cumplimiento de propiedades relevantes por el conjunto de particiones, permite asignarle un peso a cada partición y así darle una mayor importancia en la combinación a las particiones que mayor peso obtengan.
- Este análisis del cumplimiento de propiedades, permite además, obtener particiones más robustas, ya que a particiones ruidosas se le debe asignar un valor de peso pequeño y por tanto se disminuye su influencia en el resultado final.
- Esta función de consenso, al utilizar como similaridad entre particiones una función kernel, permite tener una representación exacta de la partición de consenso en un espacio de Hilbert. Esto permite saber a qué distancia se encuentra cualquier partición de la partición de consenso buscada.
- Para resolver el problema de la pre-imagen, la heurística utilizada puede ser mejorada, explotando más las características particulares del tipo de grafo con el que se trabaja.

## 7 Conclusiones

La creciente aparición de nuevos métodos de combinación de resultados de clasificadores no supervisados, muestra el auge que ha alcanzado esta técnica en los últimos años, la cual se ha presentado como una alternativa de solución al problema de la clasificación no supervisada.

En nuestra opinión, esta herramienta no ha sido explotada al máximo, incluso en ocasiones se ha interpretado su utilidad de manera errónea. La combinación de agrupamientos sigue siendo un problema no supervisado, por tanto no se puede esperar que dé como resultado la *mejor* estructuración de los datos, la *natural* o *intrínseca*. Para este problema estos términos son totalmente ambiguos, y no pueden ser el objetivo ni lo que se espere de esta técnica. Una buena combinación de agrupamientos debe ser capaz de encontrar un consenso entre todas las estructuraciones presentes, resaltando propiedades relevantes en dichas estructuraciones que permitan obtener una agrupación final de los datos que combine la información presente en todas las estructuraciones, que sea más robusta al ruido, y por tanto que brinde una estructura de los objetos más confiable que la obtenida por algún clasificador no supervisado simple, la cual conste de algún sentido o muestre cierta información importante para la solución de algún problema en particular.

La evaluación de una combinación de agrupamientos se realiza sobre la misma base con que se evalúa un clasificador simple, es decir, mediante el uso de índices de validación (internos o externos). Para obtener buenos resultados en la combinación es importante tener en cuenta que la calidad de las estructuraciones puede ser distinta, dependiendo ésta, de los clasificadores seleccionados, los datos del problema y la manera en que se mida dicha calidad, pudiendo existir incluso estructuraciones que sólo representen ruido para el consenso. Por tanto la utilización de índices de validación para obtener información del cumplimiento de diferentes propiedades en las estructuraciones permite hacer una combinación más eficaz, en la cual tengan mayor participación las estructuraciones que fueron mejor evaluadas. En este sentido hay muy pocos trabajos en la actualidad, la gran mayoría combina las diferentes estructuraciones independientemente de la validez que éstas puedan tener, obteniéndose mecanismos de combinación muy sensibles al fallo cuando existen estructuraciones muy ruidosas en la combinación.

Por otra parte, entre las distintas definiciones de función de consenso, las basadas en la búsqueda de la partición media, son las que han sido fundamentadas teóricamente de manera más rigurosa. Sin embargo, en la mayoría de las heurísticas propuestas para darle solución al problema, no queda claro cómo es que estas heurísticas se acercan a la solución del problema teórico, más bien son algoritmos que generan una estructuración a partir de un conjunto de estructuraciones, siguiendo un criterio que de alguna manera combina la información en las mismas.

Otro tema que ha sido poco tratado, es la influencia que puede tener en los resultados de la combinación la naturaleza de los datos. Cómo puede influir que estos sean numéricos, nominales o mezclados, qué tratamiento se le puede dar a la ausencia de información.

Además, hasta la actualidad la gran mayoría de los métodos de combinación propuestos son diseñados para la combinación de particiones, y casi la totalidad para particiones duras, existiendo aun muy pocos trabajos para particiones difusas. No se han desarrollado mecanismos de combinación para algoritmos de clasificación no supervisada que den como resultado una estructuración que sea un cubrimiento del conjunto de objetos (ya sean cubrimientos duros o difusos).

Por tanto podemos concluir que este es un tema nuevo, que ha ganado mucho auge en los últimos años, el cual puede ser de gran ayuda en la solución de problemas de clasificación no

supervisada, para el cual no sólo es necesario desarrollar nuevos algoritmos más eficaces y eficientes, sino también formalizar conceptos básicos como el de función de consenso y lo que se espera de ésta, proporcionando métodos de comparación realmente efectivos, que puedan determinar cuándo es preferible utilizar un modelo de combinación u otro.

A la búsqueda de un nuevo esquema de combinación de resultados de clasificadores no supervisados en el cual se le dé una solución efectiva a las problemáticas anteriormente planteadas dedicaremos nuestros esfuerzos futuros.

## Referencias bibliográficas

1. Jain, A.K., Law, M.H.C.: Data Clustering: A User's Dilemma. In: Bandyopadhyay, S., Biswas, S. (eds.): PReMI 2005, Vol. 3776. Springer, Heidelberg (2005) 1-10
2. MacQueen, J.: Some methods for classification and analysis of multivariate observations.: Fifth Berkeley Symposium on Math. Stat. and Prob. University of California Press (1967) 281-297
3. McLachlan, G., Peel, D.: Finite Mixture Models. John Wiley & Sons, New York (2000)
4. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 603-619
5. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000) 888-905
6. Verma, D., Meila, M.: A comparison of spectral clustering algorithms. University of Washington (2003)
7. Ruiz-Shulcloper, J., Sánchez-Díaz, G.: Clustering in mixed incomplete data. In: H.A. Abbass, e.a. (ed.): *Heuristics & Optimization for Knowledge Discovery*. Idea Group Publishing, USA (2002) 88-106
8. Ruiz-Shulcloper, J., Abidi, M.: Logical Combinatorial Pattern Recognition: A Review. In: Pandalai, S.G. (ed.): *Recent Reserch Developments in Pattern Recognition*, Vol. 3. Transword Research Networks, USA (2002) 133-176
9. Bezdek, J.C.: FCM: The fuzzy c-means clustering algorithm. *Computers and Geosciences* **10** (1984) 191-203
10. Jain, A.K., M.N., M., Flynn, P.J.: Data Clustering: A Review. *ACM Computing Surveys (CSUR)* **31** (1999) 264-323
11. Fred, A., Jain, A.K.: Data Clustering Using Evidence Accumulation. 16th International Conference on Pattern Recognition, ICPR 2002. Quebec City (2002) 276-280
12. Strehl, A., Ghosh, J.: Cluster ensembles --- a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3** (2003) 583-617
13. Punera, K., Ghosh, J.: Soft Cluster Ensembles. In: Oliveira, J.V.d., Pedrycz, W. (eds.): *Advances in Fuzzy Clustering and Its Applications*. John Wiley & Sons (2007)
14. Avogadri, R., Valentini, G.: Ensemble Clustering with a Fuzzy Approach. *Studies in Computational Intelligence (SCI)* **126** (2008) 49-69
15. Duin, R.P.W., Pekalska, E.: Open Issues in Pattern Recognition. In: Kurzynski, M., Puchala, E., Wozniak, M., Zolnnerek, A. (eds.): *Fourth International Conference on Computer Recognition Systems (CORES 2005)*. Springer-Verlag, Wroclaw, Poland (2005) 27-42
16. He, Z., Xu, X., Deng, S.: An Efficient Algorithm for Clustering Categorical Data. *Journal of Computer Science and Technology* **17** (2002) 611-625
17. Guha, S., Rastogi, R., Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Int. Conf. Data Engineering 1999*, Sydney, Australia (1999) 512-521
18. Gibson, D., Kleiberg, J., Raghavan, P.: Clustering Categorical Data: An Approach Based on Dynamic Systems. *Int. Conf. On Very Large Databases*, New York (1999) 311-323
19. Martínez-Trinidad, J.F., Ruiz-Shulcloper, J., Lazo-Cortés, M.S.: Structuralization of universes. *Fuzzy set and systems* **112** (2000) 485-500

20. Simovici, D.A.: Data Mining Algorithms: Clustering. In: Nayak, A., Stojmenovic, I. (eds.): Handbook of Applied Algorithms: Solving Scientific, Engineering and Practical Problems. John Wiley & Sons (2008)
21. Xu, R., II, D.W.: Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks* **16** (2005)
22. Everitt, B., Landay, S., Leese, M.: Cluster Analysis. Arnold, London (2001)
23. Gordon, A.: Classification, 2nd ed. Chapman & Hall, London, U.K. (1999)
24. Hansen, P., Jaumard, B.: Cluster analysis and mathematical programming. *Math. Program.* **79** (2000) 191-215
25. Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., Dougherty, E.R.: Model-based evaluation of clustering validation measures. *Pattern Recognition* **40** (2007) 807-824
26. Rijsbergen, C.: Information Retrieval. second edition Butterworths (1979)
27. Rand, W.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **66** (1971) 846-850
28. Hubert, A.: Comparing partitions. *Journal of Classification* **2** (1985) 193-198
29. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. *Intell. Inf. Syst. J.* **17** (2001) 107-145
30. Jardine, N., Sibson, R.: Mathematical Taxonomy. John Wiley & Sons (1971)
31. Yeung, K.Y., Haynor, D.R., W.L.Ruzzo: Validating clustering for gene expression data. *Bioinformatics* **17** (2001) 301-318
32. Lange, N., Braun, M., Roth, V., Buhmann, J.M.: Stability-based model selection. *Advances in Neural Information Processing Systems*
33. Roth, V., M.Braun, Lange, T., Buhmann, J.M.: Stability-Based Model Order Selection in Clustering with Applications to Gene Expression Data. Springer, Berlin (2002)
34. Bezdek, J.C., Pal, N.: Some new indexes of cluster validity. *IEEE Transactions Syst. Man Cybernet.* **28** (1998) 301-315
35. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20** (1987) 53-65
36. Ding, C., He, C.: K-nearest neighbor consistency in data clustering: incorporation local information into global optimization. In: al., H.H.M.e. (ed.): ACM Symposium on Applied Computing 2004. ACM Press, New York (2004) 584-589
37. Handl, J., Knowles, J.: Exploiting the trade-off the benefits of multiple objectives in data clustering. In: al., C.L.A.e. (ed.): Third International Conference on Evolutionary Multicriterion Optimization. Springer-Verlag, Berlin (2005) 547-560
38. Topchy, A., Jain, A.K., Punch, W.: Combining Multiple Weak Clusterings. *IEEE Intl. Conf. on Data Mining, ICDM* (2003)
39. Topchy, A., Jain, A.K., Punch, W.: A Mixture Model of Clustering Ensembles. *SIAM Int. Conference on Data Mining* (2003)
40. Vega-Pons, S., Correa-Morris, J., Ruiz-Shulcloper, J.: Weighted Cluster Ensemble Using a Kernel Consensus Function. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.): CIARP 2008. Springer-Verlag Heidelberg (2008) 195-202
41. Kuncheva, L.I.: Combining Pattern Classifiers. Methods and Algorithms. John Wiley & Sons, New York (2004)
42. Ben-Hur, A., Elisseeff, A., Guyon, I.: A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing* (2002) 6-17
43. Cover, T.M., Thomas, J.A.: Elements of Information Theory. In: Schilling, D.L. (ed.). John Wiley & Sons, Inc., New York (2006)
44. Fred, A., Jain, A.K.: Robust data clustering. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, USA* (2003)
45. Scholkopf, B., Smola, A.J.: Learning with Kernels. Mit Press (2002)
46. Gartner, T., Flach, P.A., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives. *16th Annual Conference on Computational Learning Theory and the 7th Kernel Workshop* (2003)

47. Gartner, T., Lloyd, J.W., Flach, P.A.: Kernel for structured data. 12th International Conference on Inductive Logic Programming Springer-Verlag (2002)
48. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized Kernels Between Labeled Graphs. Twentieth Int. Conf. on Machine Learning (2003)
49. Fern, X.Z., Brodley, C.E.: Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. 20th Int. conf. on Machine Learning ICML (2003)
50. Fischer, B., Buhmann, J.M.: Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 513-518
51. Minaei-Bidgoli, B., Topchy, A., Punch, W.F.: Ensembles of Partitions via Data Resampling. Intl. Conf. on Information Technology ITCC 2004, Las Vegas (2004) 188-192
52. Fred, A.L.N.: Finding Consistent Clusters in Data Partitions. In: Roli, F., Kittler, J. (eds.): 3rd. Int. Workshop on Multiple Classifier Systems (2001) 309-318
53. Minaei-Bidgoli, B., Topchy, A., Punch, W.F.: A comparison of re-sampling methods for clustering ensembles.: IC-AI (2004) 939-945
54. Monti, S., Tamayo, P., Mesirov, J., Golub, T.: Consensus Clustering: A resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. Jorunal on Machine Learning **52** (2003)
55. Fred, A.L.N., Jain, A.K.: Combining multiple clustering using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 835-850
56. Fred, A.L.N., Lourenco, A.: Cluster Ensemble Methods: from single clustering to combined solutions. Studies in Computational Intelligence (SCI) **126** (2008) 3-30
57. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics **19** (2003) 1090-1099
58. Dimitriadou, E., Weingessel, A., Hornik, K.: Voting-Mergin: An Ensemble Method for Clustering. In: Dorffner, G., Bischof, H., Hornik, K. (eds.): ICANN 2001 (2001) 217-224
59. Hye-Sung, Y., Sang-Ho, L., Sung-Bum, C., Ju-Han, K.: A Novel Framework for Discovering Robust Cluster Results. In: Lavrac, N., Todorovski, L., Jantke, K.P. (eds.): LNAI. Springer-Verlag Berling Heidelberg (2006) 373-377
60. Al-Razgan, M., Domeniconi, C.: Weighted clusteing ensembles. In: Ghosh, J., Lambert, D., Skillicorn, M.D., Srivastava, J. (eds.): SIAM Int. Conf. Data Mining. SIAM, Philadelphia (2007) 258-269
61. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal of Scientific Computing **20** (1998) 359-392
62. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: Applications in VLSI domain. Design and Automation Conference (1997)
63. Mikin, B.: Reinterpreting the Category Utility Function. Machine Learning **45** (2001) 219-228
64. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood From Incomplete Data Via the EM Algorithm. Journal of Royal Satistical Society **39** (1997) 1-22
65. Domeniconi, C., Gunopulos, D., Ma, S., Yan, B., Al-Razgan, M., Papadopoulos, D.: Locally adaptive metrics for clustering high dimensionl data. Data Mining and Knowledge Discovery Journal **14** (2007) 63-97
66. Domeniconi, C., Papadopoulos, D., Gunopulos, D., Ma, S.: Subspace clustering of high dimensional data. SIAM International Conference on Data Mining (2004)
67. Bakir, G.H., Zien, A., Tsuda, K.: Learning to Find Graph Pre-Images. In: Rasmussen, C.E., Bulthoff, H.H., Scholkopf, B., Giese, M.A. (eds.): DAGM 2004, Vol. 3175. Springer, Heidelberg (2004) 253-261

RT\_029, enero 2010

Aprobado por el Consejo Científico CENATAV

Derechos Reservados © CENATAV 2010

**Editor:** Lic. Lucía González Bayona

**Diseño de Portada:** DCG Matilde Galindo Sánchez

RNPS No. 2142

ISSN 2072-6287

**Indicaciones para los Autores:**

Seguir la plantilla que aparece en [www.cenatav.co.cu](http://www.cenatav.co.cu)

C E N A T A V

7ma. No. 21812 e/218 y 222, Rpto. Siboney, Playa;

Ciudad de La Habana. Cuba. C.P. 12200

*Impreso en Cuba*

