RNPS No. 2142
ISSN 2072-6287
Versión Digital

SERIE AZUL

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

REPORTE TÉCNICO
Reconocimiento de Patrones

Combinación de Resultados de Clasificadores No Supervisados

Sandro Vega Pons

RT_044 **diciembre 2011**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

RNPS No. 2142
ISSN 2072-6287
Versión Digital

SERIE AZUL

REPORTE TÉCNICO
Reconocimiento de Patrones

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

Combinación de Resultados de Clasificadores No Supervisados

Sandro Vega Pons

RT_044 **diciembre 2011**

7ma. No. 21812 e/218 y 222,
Rpto. Siboney, Playa;
La Habana.
Cuba. C.P. 12200
www.cenatav.co.cu

UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN
DEPARTAMENTO DE MATEMÁTICA

Combinación de Resultados de Clasificadores No Supervisados

Tesis presentada en opción al grado científico de

Doctor en Ciencias Matemáticas

Autor:

Lic. Sandro Vega Pons

Tutor:

Dr. Cs. José Ruiz Shulcloper

Centro de Aplicaciones de Tecnologías de Avanzada

Ciudad de La Habana, Cuba

2010

A mis padres ...

Agradecimientos

Le agradezco a todas las personas que han contribuido al desarrollo de este trabajo:

A mi mama y mi papa por brindarme siempre su apoyo incondicional.

A Diana, mi *miji*, por su amor y dedicación. Por su valiosa ayuda en todo momento.

A Shul por las horas de trabajo juntos, su visión y guía.

A mi hermano, la flaca, el fernan y resto de mi familia por poder contar con ellos.

A mi cuñi Dania, Rosy, Nikito y demás integrantes de la familia Porro-Muñoz por ser una segunda familia para mí.

Al Prof. Xiaoyi Jiang de la Universidad de M*ü*nster, Alemania, por recibirme en su grupo de investigación y por su colaboración en parte de los resultados de esta tesis.

A los profesores de la UCLV por la forma en que me acogieron y sus oportunas sugerencias.

A las *doctorantas* Noslen y Heydi por ir un pasito adelante y servirme de guía.

A Lucía por las muchísimas búsquedas bibliográficas y al teacher for all his help.

A Oneysis, Rainer, Rolando, Reynel, Airel, Raudel, Chang, Walter, Dustin, Gabriel, Alexis, Annette, Kadir, Gago y en general a los colegas de mi departamento y del CENATAV, por su apoyo y los momentos compartidos.

A mis amistades de toda la vida porque donde quiera que estén se que puedo contar con ellos.

SÍNTESIS

En esta tesis se investigan los métodos de combinación de agrupamientos. A pesar del auge que han tenido estos métodos en los últimos años debido a su utilidad en problemas prácticos, no existe un estudio completo de los diferentes tipos de métodos propuestos. En este documento, primeramente, se presenta un análisis de los diferentes métodos de combinación de agrupamientos existentes en la literatura, teniendo en cuenta las diferentes formas de formular teóricamente el problema, así como las diversas herramientas matemáticas y computacionales utilizadas en cada tipo de método. Posteriormente, a partir de las deficiencias encontradas en los algoritmos estudiados, se proponen los métodos de combinación de agrupamientos basados en funciones núcleo, los cuales presentan algunas ventajas con respecto a los métodos existentes. Dos partes fundamentales de este tipo de métodos son: el uso de un mecanismo automático de análisis de la calidad de las particiones y el uso de una medida de similitud entre particiones que cumpla la propiedad de ser una función núcleo. De esta manera, se proponen tres medidas de similitud entre particiones que satisfacen dicha propiedad. Por otra parte, se estudia la importancia del uso de la información contenida en el conjunto de objetos originales del problema para el paso de combinación de los agrupamientos. Se desarrollan algoritmos capaces de utilizar dicha información en el proceso de combinación y trabajar en problemas con datos numéricos, no numéricos y mezclados. Además, se estudia la relación existente entre el problema de la combinación de agrupamientos y otros problemas de Reconocimiento de Patrones de gran importancia práctica como son: la selección de la partición representativa en una jerarquía de particiones y la combinación de diferentes segmentaciones de una imagen. De esta forma se propone un nuevo enfoque para la selección de la partición representativa en una jerarquía utilizando los resultados obtenidos en el desarrollo de los métodos de combinación de agrupamientos basados en funciones núcleos. También, se propone un nuevo método de combinación de segmentaciones donde se tiene en cuenta la posible alta dimensionalidad de la imagen y se respeta la relación espacial existente entre los pixeles que la conforman. Todos los algoritmos presentados en esta tesis fueron probados en colecciones de datos internacionales, demostrando experimentalmente una mayor eficacia que los reportados en la literatura para los mismos propósitos.

ÍNDICE

INTRODUCCIÓN

Durante los últimos años, ha existido un aumento acelerado de los datos almacenados, provenientes de una gran variedad de disciplinas. Estos datos describen características de objetos y fenómenos naturales, resumen resultados de experimentos científicos, etc. Una de las actividades más importantes dentro del análisis de datos es *clasificarlos* o *agruparlos* en categorías, clases o grupos (clusters), de tal manera que los objetos ubicados en un mismo grupo compartan más propiedades entre sí que con objetos de otros grupos. De esta forma, el estudio y desarrollo de algoritmos y sistemas computacionales capaces de realizar tal tarea, se ha convertido en un área de investigación muy activa en los últimos años.

Los sistemas de clasificación pueden ser supervisados, parcialmente supervisados o no supervisados [17, 26]. De manera general, la clasificación supervisada se basa en la disponibilidad de un conjunto de objetos de entrenamiento, donde para cada objeto, se conoce la clase a la cual está asociado. De esta forma, un nuevo objeto se clasificará en al menos una de las clases existentes (representada por una etiqueta en particular), de acuerdo a su relación de parecido con los objetos en el conjunto de entrenamiento[1]. Sin embargo, en la clasificación no supervisada o agrupamiento (clustering) no existe conjunto de entrenamiento. Su propósito general consiste en crear una estructuración (clustering) de un conjunto de objetos en grupos (clusters). Estos grupos se construyen de manera tal que objetos en un mismo grupo tienden a ser similares en algún sentido, mientras que objetos en distintos grupos tienden a ser diferentes. En estos problemas se dan dos situaciones: cuando se conoce el número de grupos en los que se debe estructurar el conjunto de objetos (*clasificación no supervisada restringida*); y cuando este número es parte del problema a resolver (*clasificación no supervisada libre*). En los problemas de clasificación parcialmente supervisados, la situación es en cierto sentido un híbrido de las dos anteriores: se

[1]En este documento no se están considerando los algoritmos basados en reglas de asociación u otros que en lugar del "parecido" entre los objetos se basan en el cumplimiento de propiedades que caracterizan a las clases dadas en los problemas.

conocen algunas de las clases del problema, se tienen muestras de algunas de ellas pero no de todas e incluso pueden aparecer nuevas clases que no se sabía que existían.

El problema de la clasificación no supervisada es una de las etapas más importantes en un gran número de ciencias, en particular, las naturales y sociales. La Taxonomía es el primer período de trabajo científico en muchas de las áreas del conocimiento y es una de las herramientas necesarias en muchas de las investigaciones en la actualidad. Todos los procesos taxonómicos, en cualquier ciencia, son instancias de este tipo de problemas. La clasificación no supervisada es la piedra angular en la mayoría de los problemas de Minería de Datos y de extracción de conocimiento. Los algoritmos de agrupamiento son de gran utilidad para la solución de un número considerable de problemas prácticos, estos encuentran aplicación directa en diversas ingenierías, ciencias de la computación, ciencias médicas y naturales, astronomía, ciencias sociales, ciencias económicas, entre otras [28, 55, 124].

En la actualidad existe una gran cantidad de algoritmos de agrupamiento reportados en la literatura. Estos se pueden catalogar en *extensionales* o *conceptuales*, en dependencia de la forma en que se determinen los grupos; en *duros* o *difusos*, en dependencia de la Teoría de Conjuntos que se emplee; en *particionales* o *jerárquicos*, si la estructuración obtenida es una sola o si se determina una sucesión anidada de estructuraciones que pueden ser *particiones* o *cubrimientos* en dependencia de las relaciones conjuntuales entre los grupos, si todos son o no disjuntos. De manera general, entre los algoritmos de agrupamiento más conocidos se encuentran: k-Means [72], Expectation Maximization (EM) [76], algoritmos basados en teoría espectral de grafos [96, 117], algoritmos jerárquicos como el Single-Link o Complete-Link [54], algoritmos de agrupamiento de datos mezclados e incompletos [91], algoritmos de estructuración difusa como Fuzzy c-Means [16], etc. (ver resumen en [55, 124]). Como se ha mencionado, todos estos algoritmos pueden ser empleados en la solución de muchos problemas prácticos. Sin embargo, a la selección de un algoritmo en particular debe llegarse a partir de un proceso de modelación matemática que, en alguna medida, asegure un empleo metodológicamente fundamentado del algoritmo. Pero, frecuentemente es muy difícil determinar a priori cuál algoritmo de agrupamiento va a funcionar *correctamente* para un problema en particular. Además, como se conoce, no existe un algoritmo de agrupamiento que pueda ser utilizado en cualquier problema de clasificación no supervisada con resultados satisfactorios. Es importante notar que cuando se aplica un algoritmo de agrupamiento a un conjunto de objetos, este impone una organización en los datos de acuerdo al criterio de agrupamiento interno

del algoritmo, las características de la medida de similitud o disimilitud entre objetos utilizada y de las propias características de los datos. Por tanto, si se tienen diferentes algoritmos de agrupamiento y se aplican sobre el mismo conjunto de objetos, se pueden obtener resultados muy diferentes. Pero cuál es el correcto. ¿Cómo pueden ser evaluados estos resultados? En clasificación no supervisada, la evaluación de los resultados está relacionada con el uso de los Índices de Validación de Agrupamientos[2] (Cluster Validity Indexes; CVI) [18], los cuales son usados para medir la calidad de las estructuraciones obtenidas por los algoritmos de agrupamiento.

De manera general, los CVIs se dividen en dos grandes grupos: *índices externos* e *índices internos*. Los *índices externos* usan como referencia para compararse, una estructuración específica, la cual se obtiene a partir de información previa acerca de los datos, donde esta estructuración es vista como la *ideal* o *verdadera* (ground-truth). Una estructuración de los datos es mejor en la medida que se parece más a dicha estructuración de referencia. De esta forma, un índice externo es una función de comparación entre estructuraciones, que evalúa el parecido entre una estructuración obtenida por un algoritmo de agrupamiento y la estructuración de referencia del problema. Entre los índices externos más usados se encuentran la medida F (F-measure) [107], el índice de Rand [86], el índice de Jaccard [13], la Variación de Información (Variation of Information; VI) [77] y la Información Mutua Normalizada (Normalized Mutual Information; NMI) [99]. Cualquier medida de (di)similitud entre estructuraciones puede ser utilizada como índice externo. Sin embargo, en problemas reales, una limitante que aparece frecuentemente es que esta estructuración de referencia no está disponible, pues de tenerla no sería necesario aplicar los algoritmos de agrupamiento.

Los *índices internos* se basan solamente en la información contenida en la propia estructuración de los datos, es decir, no necesitan información adicional para evaluar una estructuración. Un gran número de estos índices asumen que las estructuraciones obtenidas por los algoritmos de agrupamiento deben cumplir que los objetos en un mismo grupo deben ser más parecidos entre sí que objetos de grupos diferentes. Por tanto, estos evalúan la compacidad u homogeneidad de cada grupo y la separación de cada uno de los grupos en la estructuración, por ejemplo: el índice de validación SD [48], el índice de Davies-Bouldin y el Ancho de la Silueta [90]. Por otra parte, otros índices evalúan qué tanto cumple una estructuración con el concepto de conectividad [49]. Existe una gran cantidad de estos índices, cada uno de los cuales

[2]Por simplicidad, a lo largo del documento se llamarán: índices o CVI.

aplica un criterio de evaluación diferente o la combinación de un conjunto de éstos, ya sea compacidad, separación, conectividad, etc. Un *índice interno* puede decir en qué medida se cumple una o un conjunto de propiedades en una estructuración, pero esto no significa que pueda decir cuándo una estructuración es *correcta* o *mejor* que otra, estos conceptos dependen en gran medida del problema particular. Debido a la naturaleza no supervisada de los problemas de agrupamiento de datos, no siempre es posible conocer qué características o propiedades son relevantes para el problema en cuestión. Como consecuencia, la evaluación de los resultados con *índices internos* podría no responder a las necesidades reales del problema abordado si las propiedades medidas por el índice y las propiedades relevantes en el problema en cuestión no se corresponden.

Por tanto, el uso de CVIs no representa una solución definitiva al problema de hallar la estructuración más adecuada para un problema de agrupamiento dado. De hecho, en términos generales, el problema de buscar la estructuración más adecuada no tiene solución, su solución solo puede ser alcanzada en función del conocimiento específico que se tenga del problema, por ejemplo, si se sabe que la estructuración a obtener debe ser compacta y separada, los índices que midan compacidad y separabilidad serán los más adecuados en este problema. Cuando este conocimiento no está presente, no se puede hablar de la estructuración más adecuada, sino que pueden existir diferentes estructuraciones donde cada una de las cuales aporta una cierta información sobre el problema. La cuestión a resolver es cómo conciliar todas estas informaciones para obtener una *estructuración de consenso*. De ahí surge la idea de la combinación de agrupamientos[3] (Clustering Ensemble). Esta idea está basada en el éxito obtenido en la combinación de clasificadores supervisados [65]. Además, responde a la idea intuitiva de que si no se conoce la calidad de ciertos resultados individuales, la opción de combinarlos puede ser *superior* a seleccionar algún resultado simple.

La mayoría de los algoritmos de agrupamiento forman *particiones*[4] de un conjunto de datos, es decir, dado un conjunto de objetos $X = \{x_1, x_2, \ldots, x_n\}$, después de aplicar un algoritmo de agrupamiento, se obtiene una estructuración $P = \{C_1, C_2, \ldots, C_d\}$ tal que:

- $C_i \neq \emptyset, \forall i = 1, \ldots, d$

[3]Los términos *combinación de agrupamientos* y *combinación de resultados de clasificadores no supervisados* son utilizados indistintamente en este documento.

[4]Se asume que son particiones *duras* de los datos.

- $\bigcup_{i=1}^{d} C_i = X$
- $C_i \cap C_j = \emptyset, \forall i, j = 1, \ldots, d$, con $i \neq j$

Consecuentemente, la mayoría de los métodos de combinación de agrupamientos están diseñados para la combinación de particiones de los datos. Por tanto, en este documento cuando se habla de agrupamiento, estructuración o resultado de algoritmo de clasificación no supervisada, se hace referencia a particiones según la definición anterior. En caso de hablarse de otro tipo de estructuración será dicho explícitamente.

Con el objetivo de formalizar el concepto de *combinación de agrupamientos*, diferentes autores han definido un conjunto de propiedades que sería deseable encontrar en este tipo de algoritmos [37, 100]. Algunas de estas propiedades son:

- *Eficacia*: Los resultados obtenidos por algoritmos de combinación de agrupamientos deben ser, de manera general, *más adecuados* para la solución de problemas de estructuración que los resultados obtenidos por los algoritmos de agrupamiento simples.
- *Consistencia*: El resultado de la combinación debe ser, de alguna manera, muy similar a todas las particiones que se combinaron.
- *Novedad*: Los algoritmos de combinación de agrupamientos deben permitir encontrar estructuraciones de los datos inalcanzables por los algoritmos de agrupamiento simples.
- *Robustez*: Los resultados deben presentar una baja sensibilidad al ruido y valores atípicos (outliers).

Sin embargo, estas propiedades son difíciles de verificar en la práctica debido a la poca formalidad en sus definiciones y a la propia naturaleza no supervisada del proceso de agrupamiento. Además, no se han desarrollado medidas para evaluar este tipo de propiedades en los algoritmos de combinación de agrupamientos.

No obstante, en esta última década se han propuesto numerosos algoritmos de combinación de agrupamientos, motivados principalmente por los buenos resultados alcanzados por los mismos en la solución de problemas prácticos de clasificación no supervisada (ver Sección 1.3). De manera general, los métodos de combinación de agrupamientos consisten en dos pasos fundamentales: *Generación* y *Consenso*[5]. En la *generación* se obtiene un conjunto de estructuraciones (particiones) del conjunto

[5]También conocido por *Mecanismo de Combinación* o *Función de Consenso*.

de objetos originales y en el *consenso* estas particiones se combinan para obtener el resultado final: la *partición de consenso* (ver Figura 1.1).

En la literatura de los métodos de combinación de agrupamientos, se pueden encontrar diferentes definiciones de la *partición de consenso*. En ocasiones, esta se define de manera implícita como la función objetivo del algoritmo propuesto, lo cual imposibilita el análisis teórico de la misma. Por otra parte, existen algoritmos donde la partición de consenso se define rigurosamente a partir de la búsqueda de la *partición mediana* [11]. Esto es, dado un conjunto de objetos $X = \{x_1, x_2, \ldots, x_n\}$ y un conjunto de particiones de X, $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$, la *partición mediana* se define como:

$$P^* = \arg\max_{P \in \mathbb{P}_X} \sum_{j=1}^{m} \Gamma\left(P, P_j\right) \tag{1}$$

donde $\mathbb{P}_X$ es el conjunto de todas las posibles particiones de X y Γ es una medida de similitud[6] entre particiones.

Los primeros estudios matemáticos del problema de la partición mediana (1) se remontan a los trabajos de Régnier [88] y Mirkin [79]. Posteriormente, se realizaron diferentes estudios acerca del problema de la partición mediana. Sin embargo, los principales resultados teóricos han sido obtenidos para el caso particular de cuando Γ es la *diferencia simétrica* o *distancia de Mirkin* [81]. Krivanek y Moravek [63] probaron que el problema de la partición mediana (1) con la distancia de Mirkin es $\mathcal{NP}$-duro ($\underline{\mathcal{NP}\text{-hard}}$).

Este problema, con otras medidas de (di)similitud, no ha sido apropiadamente estudiado. Sin embargo, en la práctica, otras medidas de (di)similitud entre particiones como el índice de Rand, Información Mutua Normalizada, Variación de Información entre muchas otras, son ampliamente utilizadas debido a su capacidad para la comparación de particiones.

Por otra parte, existen otros problemas de Reconocimiento de Patrones de gran importancia práctica que se encuentran estrechamente relacionados con el problema de la búsqueda de la partición mediana (1). Sin embargo, estos se definen y enfrentan usualmente desde otras perspectivas. En particular se hace referencia al problema de encontrar un nivel representativo en una jerarquía de particiones y al problema de la combinación de segmentaciones de una imagen.

De esta forma, al analizar el problema de la partición mediana, los algoritmos

[6] Este problema puede definirse de manera análoga como $P^* = \arg\min_{P \in \mathbb{P}_X} \sum_{j=1}^{m} \Gamma\left(P, P_j\right)$ cuando Γ es una medida de disimilitud entre particiones.

de combinación de agrupamientos existentes en la literatura y los dos problemas de Reconocimiento de Patrones mencionados anteriormente, se encuentran las siguientes **motivaciones** para realizar esta investigación:

- En la actualidad, el desarrollo de algoritmos de combinación de agrupamientos ha ganado un gran auge, sin embargo, no existe un estudio comparativo completo de los métodos propuestos.

- Un problema fundamental en combinación de agrupamientos es cómo definir teóricamente la partición de consenso. En este sentido, el enfoque basado en la búsqueda de la partición mediana es el que mayores garantías presenta. Sin embargo, este enfoque implica la solución de un problema combinatorio exponencial. Por tanto, los algoritmos basados en este enfoque siguen diferentes heurísticas. No obstante, muchas de estas heurísticas no enfrentan directamente el problema combinatorio planteado, en otras palabras, no van en la dirección de encontrar la solución de este problema.

- La gran mayoría de estos algoritmos combinan los agrupamientos sin tener en cuenta la calidad de cada agrupamiento. Sin embargo, en el proceso de generación puede que se obtengan particiones de muy baja calidad, que solo representen ruido para el proceso de combinación. Por tanto, un análisis de la calidad de las particiones antes del proceso de combinación podría mejorar los resultados finales.

- Los algoritmos de combinación de agrupamientos, en el paso de consenso, sólo utilizan información del conjunto de particiones para obtener la partición de consenso. Sin embargo, el conjunto de objetos originales del problema y sus valores de (di)similitud son informaciones valiosas que podrían aumentar la eficacia de estos algoritmos.

- Los problemas de la selección del nivel representativo de una jerarquía de particiones y de la combinación de segmentaciones de una imagen pueden ser modelados como subproblemas del problema de la combinación de agrupamientos. De esta manera, ambos podrían beneficiarse de los resultados obtenidos para el problema general de la combinación de agrupamientos.

El **objetivo general** de esta tesis es obtener un conjunto de resultados que permita el desarrollo de algoritmos eficaces de combinación de agrupamientos y con un

fundamento teórico adecuado, que además permitan resolver problemas tales como la selección de un nivel representativo en una jerarquía de particiones y la combinación de diferentes segmentaciones de una imagen. Desde el punto de vista metodológico este problema se abordó de acuerdo a los siguientes **objetivos específicos**:

- Realizar un análisis crítico de los algoritmos de combinación de agrupamientos existentes en la literatura. Haciendo énfasis en las diferentes maneras de definir la partición de consenso y en las diferentes herramientas matemáticas y computacionales utilizadas para desarrollar los algoritmos existentes.
- Desarrollar nuevas medidas de similitud entre particiones que faciliten el proceso de combinación de particiones.
- Desarrollar métodos para la estimación de la relevancia de cada partición con el objetivo de aumentar la influencia de las particiones más informativas y reducir la influencia de particiones ruidosas en el proceso de combinación.
- Desarrollar nuevos algoritmos de combinación de agrupamientos que sean:
  - Eficaces y eficientes.
  - En los cuales el método computacional propuesto responda a la definición teórica del problema.
  - Permitan encontrar la partición de consenso sin necesidad de especificar como parámetro el número de grupos en la misma.
- Desarrollar algoritmos de combinación de agrupamientos que tengan en cuenta la información de los objetos originales del problema, incluso capaces de trabajar con objetos descritos en términos de datos mezclados e incompletos.
- Desarrollar un nuevo algoritmo para la selección del nivel representativo en una jerarquía de particiones basado en los resultados obtenidos en la combinación de agrupamientos.
- Desarrollar un nuevo algoritmo de combinación de segmentaciones de una imagen basado en los resultados obtenidos en la combinación de agrupamientos.

En esta investigación se formularon las siguientes **hipótesis**:

$H1$- La definición de la partición de consenso a partir de la búsqueda de la partición mediana, utilizando un mecanismo de asignación de pesos a las particiones y una

medida de similitud entre particiones que sea una función núcleo (kernel functions) [94], permite desarrollar algoritmos de combinación de agrupamientos más eficaces que los existentes en la literatura.

*H*2- El uso de los objetos originales del problema y sus valores de similitud en el proceso de combinación de particiones permite aumentar la eficacia de los algoritmos de combinación de agrupamientos.

*H*3- Los problemas de la selección del nivel representativo de una jerarquía de particiones y de la combinación de segmentaciones de una imagen pueden modelarse como subproblemas de la combinación de agrupamientos. De esta manera, se pueden desarrollar algoritmos eficaces para solucionar ambos problemas a partir de los resultados obtenidos para la combinación de agrupamientos.

La **novedad científica** de la investigación radica en:

- Dos medidas de similitud entre particiones que satisfacen la propiedad de ser funciones núcleo. Demostración que el índice de Rand es una función núcleo. Un nuevo método de combinación de agrupamientos basado en el uso de funciones núcleo.

- Demostración de la robustez de la partición mediana como fundamentación del uso de la misma como partición de consenso.

- Un mecanismo automático de análisis de la calidad de las particiones que permite asignarle un peso a cada partición en el proceso de combinación.

- Un nuevo enfoque para la selección de la partición representativa en una jerarquía utilizando los resultados obtenidos en el desarrollo de los métodos de combinación de agrupamientos. Un nuevo método de combinación de segmentaciones donde se tiene en cuenta la posible alta dimensionalidad de la imagen y se respeta la relación espacial existente entre los pixeles que la conforman.

- Incremento de la eficacia de los métodos basados en co-asociación mediante el uso de la información contenida en el conjunto de objetos originales del problema en el paso de combinación de agrupamientos. Extensión del algoritmo de combinación de agrupamiento basado en funciones núcleo para trabajar en problemas con datos numéricos, no numéricos y mezclados.

- Desarrollo de una taxonomía de las diferentes técnicas de combinación de agrupamientos existentes en la literatura a partir de un estudio comparativo de las mismas.

El contenido fundamental de la tesis está estructurado en cuatro capítulos. En el Capítulo 1 se realiza un estudio crítico de los principales métodos de combinación de agrupamientos existentes, y se presenta una taxonomía de los mismos teniendo en cuenta las diferentes formas de definir el problema, así como las herramientas matemáticas y computacionales que son utilizadas para enfrentarlo. A partir de este estudio, en especial de las ventajas y desventajas de los métodos analizados, surgen las motivaciones para desarrollar los métodos y resultados presentados en los capítulos posteriores. En el Capítulo 2 se introduce el enfoque de combinación de agrupamientos basado en funciones núcleo. Como parte de este enfoque se introduce un procedimiento automático de asignación de pesos a las particiones para mejorar la calidad del proceso de combinación, se introducen nuevas medidas de similitud entre particiones que cumplen la propiedad de ser funciones núcleo y se desarrolla un mecanismo de consenso basado en estas. En el Capítulo 3 se presentan algoritmos de combinación de agrupamientos capaces de utilizar la información de los objetos originales y sus valores de similitud en el proceso de combinación. Se estudia y analiza la ventaja que brinda el uso de los objetos originales y los problemas que pueden surgir al utilizarlos. Además, se presenta una vía de solución efectiva para estos problemas. En el Capítulo 4 se estudian dos problemas de reconocimiento de patrones desde la perspectiva de la combinación de agrupamientos. En la sección 4.1 se analiza el problema de la selección de un nivel representativo en una jerarquía de particiones, obtenida por la aplicación de un algoritmo de agrupamiento jerárquico. Se estudia el enfoque usual para enfrentar este problema y se presenta un nuevo enfoque de solución basado en la filosofía de la combinación de agrupamientos usando funciones núcleo. En la sección 4.2 se aborda el problema de la combinación de diferentes segmentaciones de una imagen como un problema con características similares al problema de la combinación de agrupamientos. Se propone un nuevo algoritmo de combinación de segmentaciones donde se tiene en cuenta la estructura espacial de la imagen para aumentar la eficacia y eficiencia del algoritmo. En cada capítulo, se presentan resultados experimentales de los distintos algoritmos propuestos utilizando diferentes conjuntos de datos. Finalmente se presentan las conclusiones, recomendaciones, referencias bibliográficas y un conjunto de anexos que completan el trabajo presentado.

La mayor parte del contenido de esta tesis es una recopilación de los métodos y resultados que aparecen publicados en los siguientes artículos del autor [35, 110, 111, 112, 113, 114, 115, 116].

Capítulo 1

ENFOQUES DE LA COMBINACIÓN DE AGRUPAMIENTOS

En este documento se utilizará la siguiente notación. Sea $X = \{x_1, x_2, \ldots, x_n\}$ un conjunto de objetos, donde cada x_i es una tupla de cierto espacio de características f-dimensional Ω^f para todo $i = 1, \ldots, n$. $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$ es un conjunto de particiones, donde cada $P_i = \{C_1^i, C_2^i, \ldots, C_{d_i}^i\}$ es una partición del conjunto de objetos X con d_i grupos. C_j^i es el j-ésimo grupo de la partición i-ésima, para todo $i = 1, \ldots, m$. Además, se denota $\mathbb{P}_X$ al conjunto de todas las posibles particiones de X, ($\mathbb{P} \subset \mathbb{P}_X$). La *partición de consenso* (objetivo de todo algoritmo de combinación de agrupamientos) se denota por P^* ($P^* \in \mathbb{P}_X$).

Por otra parte, en este documento se utilizan los conceptos *similitud* y *disimilitud* para los cuales no existe un acuerdo a nivel mundial. En esta tesis, se asume una definición general de estos conceptos basado en la definición de similitud propuesta en [75]. Dado un conjunto de objetos X, una medida de similitud (disimilitud) es una función $\Gamma : X \times X \to \mathbb{R}$ ($\pi : X \times X \to \mathbb{R}$) acotada tal que $\Gamma(x, x) = M$ ($\pi(x, x) = m$) $\forall x \in X$ donde M (m) es el máximo (mínimo) valor que alcanza la función. Además, el concepto de medida de similitud (disimilitud) se asocia a la idea intuitiva de que grandes valores de la función de similitud (disimilitud) significan un mayor (menor) parecido entre los objetos comparados, mientras que valores pequeños de la función de similitud (disimilitud) se interpretan como un menor (mayor) parecido entre los objetos. Al añadirle otras propiedades a estas medidas se obtienen conceptos ampliamente conocidos y utilizados, como por ejemplo, una disimilitud π que satisface

$(\pi(x,y) = 0) \Leftrightarrow (x = y)$, $\pi(x,y) = \pi(y,x)$ y $\pi(x,y) \leq \pi(x,z) + \pi(z,y)$ $\forall x, y, z \in X$ se dice que es una distancia o métrica.

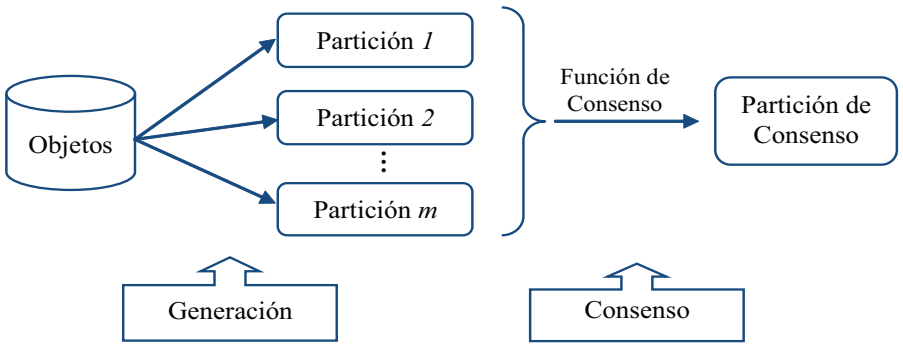


Figura 1.1: Diagrama del proceso general de los algoritmos de combinación de agrupamientos.

Los algoritmos de combinación de agrupamientos pueden dividirse en dos pasos fundamentales: el mecanismo generación y la función de consenso (ver Figura 1.1). Los diferentes mecanismos de generación se describen en la Sección 1.1 y en la Sección 1.2 se analizan las principales funciones de consenso y se presenta una taxonomía de las mismas. Posteriormente, se presentan algunas aplicaciones de estos métodos en la Sección 1.3 y finalmente en la Sección 1.4 se presentan las consideraciones finales de este capítulo.

1.1. Mecanismos de generación

La generación es el primer paso de los algoritmos de combinación de agrupamientos. En este paso se genera el conjunto de particiones que serán combinadas. En un problema particular es muy importante aplicar un mecanismo de generación apropiado, ya que el resultado final va a estar condicionado por la calidad de las particiones iniciales obtenidas en este paso.

Existen algoritmos de combinación de agrupamientos como el Voting-k-means [38] que funcionan con un proceso de generación bien determinado, es decir, no combinan cualquier conjunto de particiones. En este caso todas las particiones deben obtenerse mediante la aplicación del algoritmo k-Means con diferentes inicializaciones del parámetro k que representa el número de grupos. Este método usa un valor grande de k, con el objetivo de obtener una estructura compleja en la partición de consenso a

partir de la combinación de pequeños grupos de forma hiperesférica en las particiones iniciales.

Sin embargo, de manera general, en el paso de generación de los algoritmos de combinación de agrupamientos no hay restricciones acerca de cómo las particiones deben obtenerse. Por lo tanto, en el proceso de generación pueden ser aplicados diferentes algoritmos de agrupamiento o el mismo algoritmo con distintas inicializaciones de los parámetros. Además, pueden ser usados diversas representaciones de los objetos, diferentes funciones de similitud entre los objetos, distintos subconjuntos de objetos o proyecciones de los objetos en diversos subespacios (ver Figura 1.2).

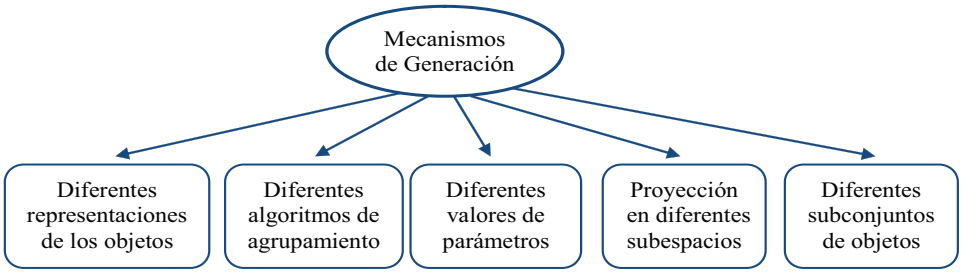


Figura 1.2: Diagrama de las principales técnicas utilizadas en la etapa de generación.

En la etapa de generación, es aconsejable utilizar aquellos algoritmos que mejor se ajusten a las características del problema en cuestión. Sin embargo, en la mayoría de los casos es muy difícil conocer a priori cuál algoritmo de agrupamiento va a ser apropiado para un problema específico. La experiencia de los expertos en el área del problema y una adecuada modelación matemática [20] pueden ser muy útiles en estos casos. Además, si no hay información acerca del problema, se recomienda hacer un conjunto de particiones diversas, ya que mientras más variado es el conjunto de particiones, mayor es la información disponible en la etapa de combinación, debido a que, como se mencionó anteriormente, cada algoritmo de agrupamiento introduce una estructuración de los objetos, basada en las propiedades intrínsecas del propio criterio de agrupamiento. Esta diversidad puede obtenerse usando diferentes mecanismos de generación como se muestra en la Figura 1.2.

1.2. Funciones de consenso

La función de consenso es el paso principal en cualquier algoritmo de combinación de agrupamientos. En este paso, se obtiene la partición final de los datos o partición de consenso P^*. Sin embargo, esta no se define formalmente de la misma manera en

todos los métodos existentes. En la siguiente sección se presenta un estudio de las dos formas fundamentales de definición de la partición de consenso. Posteriormente, se presenta una taxonomía de los diferentes tipos de mecanismos de combinación de agrupamientos basado en las herramientas matemáticas y computacionales que se utilizan en cada caso.

1.2.1. Definiciones de la partición de consenso

Después de un estudio de los métodos de combinación de agrupamientos reportados en la literatura, se detectaron dos enfoques fundamentales para la definición de la partición de consenso, los cuales se nombrarán *co-ocurrencia de objetos* y *partición mediana* [115].

Co-ocurrencia de objetos

En este enfoque, la idea es determinar cuál debe ser la *etiqueta de grupo*[1] asociada a cada objeto en la partición de consenso. Para hacer esto, se analiza cuántas veces un objeto pertenece a un grupo determinado o cuántas veces dos objetos pertenecen al mismo grupo en todas las particiones a combinar. La partición de consenso se obtiene mediante un proceso de votación entre objetos, de alguna manera, cada objeto debe votar por el grupo al cual va a pertenecer en la partición de consenso. En estos casos no es posible obtener una expresión matemática de la partición de consenso, sino que esta se define implícitamente a partir de los pasos de cada uno de los algoritmos basados en este enfoque. Esto limita el estudio teórico de las propiedades de este tipo de métodos. No obstante, una gran parte de los algoritmos de combinación de agrupamientos existentes están basados en este enfoque, por ejemplo, los métodos basados en re-etiquetamiento y votación (Sección 1.2.2) y en matriz de co-asociación (Sección 1.2.3).

Partición mediana

En este enfoque, la partición de consenso se define como la solución de un problema combinatorio: el problema de encontrar la *partición mediana* de un conjunto de

[1] Identificador que denota cada uno de los grupos en una partición.

particiones. Formalmente, la partición mediana se define como:

$$P^* = \arg\max_{P \in \mathbb{P}_X} \sum_{j=1}^{m} \Gamma\left(P, P_j\right) \tag{1.1}$$

donde Γ es una medida de similitud entre particiones. La partición mediana se define como la partición que maximiza la similitud con todas las particiones en el conjunto de particiones[2]. Este enfoque está relacionado con la noción de valor central en estadística, en particular con la *mediana*. En el Anexo 3 se demuestra la robustez de la partición mediana, que significa que dada la partición mediana de un conjunto de n particiones, al añadir $m < n$ particiones tan ruidosas como se quiera, la partición mediana de este nuevo conjunto se mantiene relativamente cerca de la partición mediana inicial. Este resultado fundamenta la utilización de la partición mediana como partición de consenso.

El problema (1.1) puede analizarse en la estructura algebraica asociada al conjunto de todas las particiones de un conjunto de objetos, dada por la siguiente relación de orden parcial.

Definición 1.1. *Sea X un conjunto de objetos y $\mathbb{P}_X$ el conjunto de todas las posibles particiones del conjunto X. Sobre $\mathbb{P}_X$ se puede definir la relación de orden parcial*[3] "anidado en"[4] *denotada por $\preceq$, donde $P \preceq P'$ si y solo si, para todo grupo $C' \in P'$ existen grupos $C_{i_1}, C_{i_2}, \ldots, C_{i_v} \in P$, con $v \geq 1$ tal que $C' = \bigcup_{j=1}^{v} C_{i_j}$. En este caso se dice que P es* más fina *(finer) que P' (o P' es* más gruesa *(coarser) que P).*

Una estructura de retículo está asociada con esta relación de orden (ver ejemplo en Figura 1.3). En esta, para cada par de particiones P, P' se definen las operaciones binarias: *ínfimo* (meet) $P \wedge P'$ que es la mayor de las cotas inferiores, i.e., la partición más gruesa de las particiones más finas que P y P'; *supremo* (join) $P \vee P'$ que es la menor de las cotas superiores, i.e., la partición más fina de las particiones más gruesas que P y P'.

El problema de la partición mediana, utilizando la *diferencia simétrica* o *distancia de Mirkin* [81] como medida de disimilitud entre particiones, ha sido estudiado desde los años sesenta del pasado siglo. Precisamente, el primer tratamiento matemático del problema de la partición mediana (1.1) fue presentado por Régnier [88] y poste-

[2] La partición mediana se puede definir equivalentemente minimizando la disimilitud respecto al conjunto de estructuraciones en el caso de que Γ sea una medida de disimilitud entre particiones.

[3] Relación binaria reflexiva, antisimétrica y transitiva.

[4] También conocida como *refinamiento*.

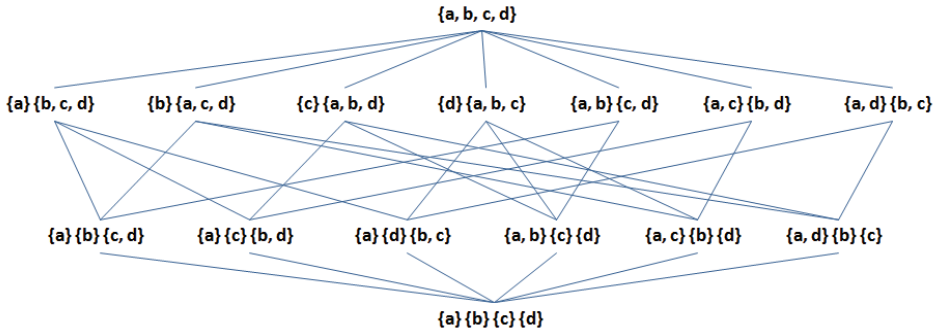


Figura 1.3: Diagrama de Hasse o representación gráfica del retículo asociado al conjunto de las particiones del conjunto de objetos $X = \{a, b, c, d\}$.

riormente por Mirkin [79], desde el punto de vista de la combinación de relaciones de equivalencia[5]. Como es conocido, una partición de un conjunto de objetos X induce una relación de equivalencia sobre X y viceversa. Por tanto el problema de hallar la partición de consenso es equivalente a hallar la relación de equivalencia consenso. En estos trabajos, junto a [11, 10, 12, 68], se presenta un tratamiento axiomático del problema utilizando las propiedades del retículo asociado a $\mathbb{P}_X$, donde se prueban algunos resultados que permiten conocer un poco acerca de la *posición* de la partición mediana en dicho retículo, como son:

- Principio de Pareto: $\bigwedge_{1 \leq i \leq m} P_i \preceq P^*$
- Principio de Co-Pareto: $P^* \preceq \bigvee_{1 \leq i \leq m} P_i$

No obstante, ninguna de estas propiedades permite hallar una solución polinomial para el problema (1.1). Krivanek y Moravek [63] y también Wakabayashi [118] probaron por vías diferentes que el problema de la partición mediana (1.1) definido con la distancia de Mirkin es $\mathcal{NP}$-duro. Esta demostración fue dada para el caso cuando hay un número variable de particiones m en el conjunto de particiones. Sin embargo, no se conoce si es un problema $\mathcal{NP}$-duro para cualquier valor m en particular [30]. Para $m = 1$ o $m = 2$ la solución del problema es trivial, pero para $m \geq 3$ no se conoce nada acerca de la complejidad computacional. Este problema desde el punto de vista de la combinación de relaciones fue extensivamente estudiado por Wakabayashi [119], donde se probó que para relaciones transitivas, el problema de hallar la relación mediana es $\mathcal{NP}$-duro. Siendo así el problema de hallar la relación de equivalencia mediana un caso particular de este.

[5]Relación binaria *reflexiva*, *simétrica* y *transitiva*

A pesar de que este problema es $\mathcal{NP}$-duro, se han propuesto algunos algoritmos para encontrar su solución exacta [119, 46]. Sin embargo, estos algoritmos solo pueden ser aplicados en pequeñas instancias del problema, es decir, en problemas con un número pequeño de objetos y de particiones.

El problema de la partición mediana (1.1) con otras medidas de (di)similitud ha sido muy poco estudiado. Sin embargo, además de la distancia de Mirkin, existen un gran número de medidas de (di)similitud entre particiones que pueden ser utilizadas en la definición de este problema. Un análisis detallado de las diferentes medidas de (di)similitud entre particiones puede ser encontrado en Meilă [77], Pfitzner *et al.* [84] y Amigó *et al.* [3]. Sin embargo, estos análisis han sido motivados por el hecho de encontrar el mejor *índice de validación externo*. Por lo tanto, las propiedades de estas medidas no han sido estudiadas desde la perspectiva de cómo ellas pueden ser apropiadas para la definición del problema de la partición mediana.

Entre las principales medidas de (di)similitud entre particiones se pueden encontrar:

- Medidas basadas en conteo de pares de objetos. Estas medidas cuentan los pares de objetos en los cuales dos particiones concuerdan o discrepan. Algunas de estas son el *Índice de Rand* [86], el *Índice de Fowlkes-Mallows* [34], el *Coeficiente de Jaccard* [13], la *distancia de Mirkin* [81] y algunas versiones de estas medidas.

- Medidas basadas en cotejo de conjuntos. Estas medidas están basadas en la comparación de la cardinalidad de conjuntos. Algunas de ellas son la *Pureza* y la *Pureza Inversa* [132], la *medida F* [107] y la *medida de Dongen* [106].

- Medidas basadas en teoría de la información. Estas medidas cuantifican la información compartida por dos particiones. Algunas de estas son la *Clase Entropía* [9], la *Información Mutua Normalizada* [99], la *Función de Utilidad* [80], la *Variación de Información* [77] y la *medida V* [89].

De manera general, este enfoque permite definir el problema de la combinación de agrupamientos de una manera más rigurosa. Sin embargo, en este caso el problema es definido a partir de un problema combinatorio exponencial. No obstante, existen algoritmos que siguen este enfoque, en los cuales el problema es abordado siguiendo diferentes heurísticas para tratar de encontrar o acercarse a la solución óptima de este problema. Este es el caso, por ejemplo, de los métodos basados en factorización de matrices no negativas (Sección 1.2.9) y los métodos basados en distancia de Mirkin (Sección 1.2.7).

Tipos de funciones de consenso

Siguiendo cualquiera de los dos enfoques de definición de la partición de consenso se pueden encontrar diferentes tipos de mecanismos de combinación de particiones. El problema de la combinación de agrupamientos ha sido abordado utilizando diferentes herramientas matemáticas y computacionales. En la literatura pueden encontrarse métodos basados en re-etiquetamiento y votación, matriz de co-asociación, particionamiento de (hiper)grafos, distancia de Mirkin, Teoría de la Información, modelos de mezclas, algoritmos genéticos, factorización de matrices no negativas y estructuraciones difusas. En la Figura 1.4 se presenta una taxonomía de las principales funciones de consenso.

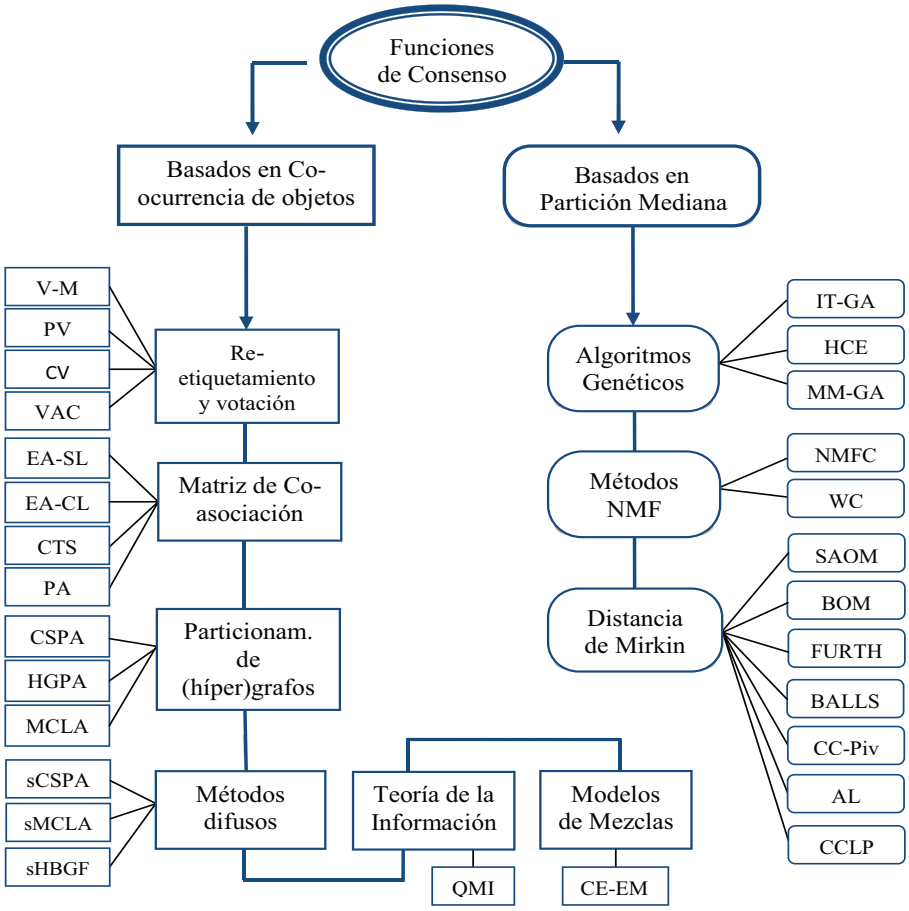


Figura 1.4: Diagrama de las principales funciones de consenso. Los métodos basados en el enfoque de co-ocurrencia de objetos son representados por un rectángulo (izquierda) y los basados en el enfoque de la partición mediana son representados por un rectángulo redondeado en las puntas (derecha).

En la Figura 1.4, además de la taxonomía basada en las herramientas matemáticas

o computacionales utilizadas en cada método de combinación de agrupamientos, se presenta una correspondencia entre cada tipo de método y uno de los dos enfoques de definición de la partición de consenso (co-ocurrencia de objetos o búsqueda de la partición mediana). Es importante notar que algunas de las funciones de consenso presentan la peculiaridad de ser definidas a través del enfoque de la partición mediana, pero en la práctica, la partición de consenso se obtiene mediante un mecanismo más relacionado con el enfoque de co-ocurrencia entre objetos. Este es el caso, por ejemplo, de los métodos basados en particionamiento de (hiper)grafos y los métodos basados en teoría de la información. En la Figura 1.4 estos métodos son clasificados según el enfoque de co-ocurrencia entre objetos.

En las siguientes secciones se presenta un análisis de cada uno de los tipos de métodos de combinación de agrupamientos. Además, en el Anexo 2 se realiza un estudio comparativo de los diferentes tipos de funciones de consenso teniendo en cuenta su comportamiento respecto a un conjunto de características, lo cual permitirá una mejor comprensión de las ventajas y desventajas de las mismas. En estos estudios, se analizan algunas propiedades de cada tipo de método, a partir de las cuales surgieron algunas de las motivaciones para realizar esta investigación.

1.2.2. Métodos basados en re-etiquetamiento

Los métodos basados en re-etiquetamiento y votación se proponen resolver como primer paso el *problema de la correspondencia de las etiquetas de los grupos*[6] y después, en un proceso de votación, cada objeto selecciona la etiqueta de grupo que le será asignada en la partición de consenso.

Por ejemplo, Dudoit y Fridlyand [27], Fischer y Buhmann [31] presentaron algoritmos de votación para calcular la partición de consenso similares al voto mayoritario usado en la combinación de clasificadores supervisados [67]. En estos métodos, se asume que el número de grupos en cada partición es el mismo e igual al número de grupos en la partición de consenso. El problema de la correspondencia de las etiquetas se resuelve utilizando el algoritmo Húngaro [64]. Después de esto, se aplica un procedimiento de votación mayoritaria para obtener la etiqueta del grupo ganador para cada objeto en la partición de consenso.

Entre otros algoritmos basados en re-etiquetamiento se encuentran:

[6]El problema de la correspondencia de las etiquetas consiste en que la etiqueta asociada a cada objeto en una partición es simbólica, es decir, no existe una relación entre los conjuntos de etiquetas utilizados por diferentes algoritmos de agrupamiento.

Plurally Voting (PV) [31], Voting-Merging (VM) [25], Voting Active Clusters (VAC) [104], Cumulative Voting (CV) [8] y los métodos propuestos por Zhou y Tang [133] y Gordon y Vichi [44].

De este tipo de método existen muchas variantes, sin embargo todas ellas tienen como objetivo enfrentar y tratar de dar solución al problema de la correspondencia de las etiquetas de los grupos. Sin embargo, este problema solo puede ser resuelto, con cierto grado de eficacia, si todas las particiones a combinar tienen el mismo número de grupos, lo cual impone una restricción muy fuerte al problema de la combinación de agrupamientos. Por lo tanto, estos métodos no son recomendables cuando el número de grupos en todas las particiones no es el mismo.

1.2.3. Métodos basados en matrices de co-asociación

La idea de co-asociación se usa para evitar el problema de la correspondencia entre etiquetas de grupos en diferentes particiones. Los métodos basados en co-asociación [37] utilizan la información en las particiones para generar una representación intermedia de los datos: la matriz de co-asociación. Esta es una matriz de $n \times n$, donde cada posición (i, j) tiene el siguiente valor:

$$CA(i,j) = \sum_{t=1}^{m} \delta_{ij}(P_t), \qquad \text{donde} \quad \delta_{ij}(P) = \begin{cases} 1, & \exists C \in P \ (x_i \in C \wedge x_j \in C) \\ 0, & \text{en otro caso} \end{cases} \tag{1.2}$$

Es decir, el valor en cada posición (i, j) de la matriz es una medida acerca de cuántas veces los objetos x_i y x_j están en el mismo grupo para todas las particiones en $\mathbb{P}$. Esta matriz puede ser vista como una nueva matriz de similitud entre objetos. Mientras más veces los objetos x_i y x_j aparezcan en el mismo grupo más similares estos serán. De esta forma, la partición de consenso se puede obtener mediante la aplicación de un algoritmo de agrupamiento sobre esta matriz de similitud de los objetos.

Fred [38] propuso una primera variante de uso de la matriz CA donde la partición de consenso se obtiene mediante la aplicación de un umbral fijo igual a 0.5. Objetos con valor de co-asociación mayor que 0.5 se unen en el mismo grupo para formar la partición de consenso.

Fred y Jain [37] propusieron una modificación al método anterior, donde después de obtener la matriz de co-asociación se aplica un algoritmo para hallar un *árbol abarcador de costo mínimo* (minimum spanning tree), es decir, viendo a la matriz de co-

asociación como la matriz de adyacencia de un grafo, encontrar un árbol que contenga todos los nodos del grafo y que el peso de sus aristas sea mínimo. Posteriormente, las aristas de este árbol son cortadas usando un umbral r. Esto es equivalente a cortar el dendrograma producido por el algoritmo de agrupamiento jerárquico Single-Link (SL) [55] utilizando el umbral r. Los algoritmos Complete-Link (CL), Average-Link (AL) [55] y otros algoritmos de agrupamiento jerárquicos también han sido utilizados como alternativas para obtener la partición de consenso en este tipo de métodos.

Otras variantes de esta matriz fueron presentadas en [70] y [120] con el objetivo de extraer más información del conjunto de particiones a la hora de conformar dicha matriz. Sin embargo, de manera general, en todos los métodos basados en co-asociación, como primer paso, se construye una nueva matriz de similitud entre objetos a partir de la información en el conjunto de particiones a combinar y posteriormente, se aplica un algoritmo de agrupamiento jerárquico para obtener la partición de consenso. Por tanto, la partición de consenso estará condicionada por la manera en que se construye la nueva matriz de similitud y por el algoritmo de agrupamiento aplicado (y los valores asignados a sus parámetros). Además, este tipo de algoritmo tiene un alto costo computacional $\mathcal{O}(n^2 \cdot m)$, por lo que no pueden ser aplicados en grandes volúmenes de datos.

1.2.4. Métodos basados en particionamiento de grafos e hipergrafos

Este tipo de algoritmos de combinación de agrupamientos transforman el problema de combinación de las particiones en un problema de particionamiento de grafos o hipergrafos. Las principales diferencias entre estos métodos se encuentran en la manera en que el (hiper)grafo es construido a partir del conjunto de estructuraciones y cómo son definidos los cortes en el (hiper)grafo para obtener la partición de consenso.

Strehl y Ghosh [99] definieron la partición de consenso como la partición que más información comparte con todas las particiones que se quieren combinar. Para medir esta información compartida por dos particiones se define la *Información Mutua Normalizada* (Normalized Mutual Information; NMI) basada en los conceptos, *Entropía* e *Información Mutua* de Teoría de la Información [22]. La NMI es una medida de similitud entre particiones definida de la siguiente manera:

Sea $P_a = \{C_1^a, C_2^a, \ldots, C_{d_a}^a\}$ y $P_b = \{C_1^b, C_2^b, \ldots, C_{d_b}^b\}$ dos particiones de X, siendo d_a el número de grupos en P_a y d_b el número de grupos en P_b. Sea n_{ia} el número de objetos en el i-ésimo grupo de la partición P_a, n_{bj} el número de objetos en el grupo

j-ésimo de la partición P_b y n_{ij} el número de objetos que están juntos en el grupo i-ésimo de la partición P_a y en el grupo j-ésimo de la partición P_b. La NMI entre P_a y P_b se define de la siguiente manera:

$$NMI(P_a, P_b) = \frac{-2\sum_{i=1}^{d_a}\sum_{j=1}^{d_b}\frac{n_{ij}}{n}\log(\frac{n_{ij}\cdot n}{n_{ia}\cdot n_{bj}})}{\sum_{i=1}^{d_a} n_{ia}\log(\frac{n_{ia}}{n}) + \sum_{j=1}^{d_b} n_{bj}\log(\frac{n_{bj}}{n})}$$

y toma valores en el intervalo real $[0, 1]$.

De esta manera, la partición de consenso se define como:

$$P^* = \arg\max_{P\in\mathbb{P}_X}\sum_{j=1}^{m} NMI(P, P_j) \tag{1.3}$$

donde $\mathbb{P}_X$ es el conjunto de todas las particiones posibles con el conjunto X.

Es decir, se propone como partición de consenso a la partición mediana utilizando la medida NMI como similitud entre particiones. Sin embargo, una búsqueda exhaustiva para resolver este problema es computacionalmente intratable. Por tanto, para enfrentar este problema fueron propuestas en [99] tres heurísticas (CSPA, HGPA y MCLA) basadas en el particionamiento de grafos e hipergrafos. Las tres comienzan representando el conjunto de particiones como un hipergrafo, donde cada grupo en cada partición es representado por un hiperarista.

En el método Cluster-based Similarity Partitioning Algorithm(CSPA), a partir del hipergrafo se construye una matriz $n \times n$ de similitud (la matriz de co-asociación CA (1.2)). Esta es vista como la matriz de adyacencia de un grafo completamente conectado, donde los nodos son los objetos y una arista entre dos objetos tiene un peso asociado igual al número de veces que los objetos fueron colocados en el mismo grupo en todas las particiones. Posteriormente, el algoritmo de particionamiento de grafos METIS [57] se usa para obtener la partición de consenso.

El método HyperGraphs Partitioning Algorithm (HGPA) particiona el hipergrafo directamente, eliminando el mínimo número de hiperaristas. Se considera que todas las hiperaristas tienen el mismo peso y se particiona el hipergrafo en k componentes de aproximadamente la misma dimensión cortando el menor número de hiperaristas. Para la implementación de este método, se utiliza el algoritmo de particionamiento de hipergrafos HMETIS [58].

En el método Meta-CLustering Algorithm (MCLA), primero que todo, se define la similitud entre dos grupos C_i y C_j en términos de la cantidad de objetos que están agrupados juntos. Para esto se utiliza el índice de Jaccard [13]. Entonces, se conforma

una matriz de similitud entre todos los grupos de todas las particiones a combinar, la cual representa la matriz de adyacencia del grafo que se forma al considerar los grupos como nodos y asignando un peso a cada arista igual a la similitud entre los grupos que componen la arista. Después, este grafo se particiona utilizando el algoritmo METIS [57] y los nuevos grupos obtenidos son llamados *meta-grupos*. Finalmente, para encontrar la partición de consenso, se calcula la cantidad de veces que cada objeto pertenece a un *meta-grupo* y se asigna al meta-grupo al cual perteneció en más ocasiones.

Otras variantes de métodos basados en particionamiento de (hiper)grafos se pueden encontrar en [29] y [1]. Este tipo de método es bastante popular debido a que son sencillos y en la mayoría de los casos tienen un bajo costo computacional, i.e., lineal respecto al número de objetos. Por ejemplo, HGPA ($\mathcal{O}(k \cdot n \cdot m)$) y MCLA ($\mathcal{O}(k^2 \cdot n \cdot m^2)$) donde n es el número de objetos, m el número de particiones y k el número de grupos en la partición de consenso. Solamente el método CSPA tiene una complejidad computacional igual a $\mathcal{O}(k \cdot n^2 \cdot m)$, la cual es cuadrática en el número de objetos. Se le presta mayor atención a la complejidad respecto al número de objetos ya que en la práctica $m << n$ y k casi siempre toma valores relativamente pequeños.

En este trabajo se considera que la mayor debilidad de este tipo de métodos es que en estos la partición de consenso se define como solución al problema de la partición mediana (1.3) utilizando la medida de similitud NMI, pero en la práctica, estos no resuelven ni se enfrentan directamente a este problema. Estos métodos están más relacionados con el enfoque de co-ocurrencia entre objetos ya que en el proceso de formación del (hiper)grafo, lo que se tiene en cuenta de manera implícita son las relaciones entre objetos individuales. Además, estos métodos necesitan un algoritmo de particionamiento de (hiper)grafos en el paso final, por lo tanto, si se cambia el algoritmo el resultado final puede ser totalmente distinto. A pesar de que el METIS y HMETIS son los más utilizados, estos no tienen por qué producir los resultados más apropiados en todas las situaciones.

1.2.5. Métodos basados en Teoría de la Información

Topchy *et al.* [102] propusieron otro método basado en la búsqueda de la solución del problema (1.1). En este caso, la *función de utilidad de categoría* (category utility function) [41] $U : \mathbb{P}_X \times \mathbb{P}_X \to \mathbb{R}$ se define como una medida de similitud entre las particiones $P_h = \{C_1^h, C_2^h, \ldots, C_{d_h}^h\}$ y $P_i = \{C_1^i, C_2^i, \ldots, C_{d_i}^i\}$ de la

siguiente manera:

$$U(P_h, P_i) = \sum_{r=1}^{d_h} \rho(C_r^h) \sum_{j=1}^{d_i} \rho(C_j^i|C_r^h)^2 - \sum_{j=1}^{d_i} \rho(C_j^i)^2 \tag{1.4}$$

donde $\rho(C_r^h) = \frac{|C_r^h|}{n}, \rho(C_j^i) = \frac{|C_j^i|}{n}$ y $\rho(C_j^i|C_r^h) = \frac{|C_j^i \cap C_r^h|}{|C_r^h|}$.

En este caso, esta función puede interpretarse como la diferencia entre la predicción de los grupos de la partición P_i teniendo en cuenta la partición P_h y sin tenerla en cuenta. De esta manera, a mayores valores de esta función, mayor parecido entre las dos particiones.

Entonces, la partición de consenso puede definirse utilizando U como medida de similitud entre particiones:

$$P^* = \arg\max_{P \in \mathbb{P}_X} \sum_{i=1}^{m} U(P, P_i)$$

En [80] se probó que este problema es equivalente a agrupar el conjunto de objetos minimizando la varianza intra-grupo, utilizando un nuevo espacio de representación para los objetos. Por tanto, la solución de este puede ser aproximada mediante la aplicación del algoritmo de agrupamiento k-Means sobre ese nuevo espacio de representación. Por otra parte, utilizando una definición general de entropía [22], la función de utilidad U puede ser transformada en la información mutua normalizada. Luego, este método propone el mismo criterio de consenso que la información mutua normalizada (NMI), con la ventaja de que el algoritmo k-Means puede ser utilizado como heurística de solución.

Este algoritmo define el problema de la combinación de particiones según la búsqueda de la partición mediana y se propone una heurística de solución. En este método, la *función de utilidad de categoría* se usa como medida de similitud entre particiones. Sin embargo, la heurística propuesta para obtener la partición de consenso utiliza el algoritmo k-Means para determinar las etiquetas de grupo asociadas a cada objeto en la partición de consenso. En la práctica, este método está más relacionado con el enfoque de co-ocurrencia que con el de la partición mediana. Por otra parte, la partición final está condicionada por la estructura impuesta sobre los datos por el algoritmo k-Means. Además, este algoritmo requiere ser aplicado en varias ocasiones para evitar la convergencia a mínimos locales de baja calidad. Sin embargo, la complejidad computacional de este método es baja $\mathcal{O}(k \cdot n \cdot m)$.

1.2.6. Métodos basados en modelos de mezclas

Topchy *et al.* [100] propusieron un método de combinación de agrupamientos, donde la partición de consenso se obtiene como la solución de un problema de estimación de máxima verosimilitud. El problema de máxima verosimilitud es resuelto mediante el uso del algoritmo de agrupamiento Expectation Maximization; EM [76].

Este enfoque está basado en un modelo de mezclas finitas para modelar la probabilidad de asignar cada etiqueta a los objetos en la partición de consenso. La principal asunción es que las etiquetas y_i (etiqueta asignada al objeto x_i en la partición de consenso) se modelan a partir de variables aleatorias con una distribución de probabilidad descrita a partir de una mezcla de componentes multivariadas:

$$\rho(y_i|\Theta) = \sum_{t=1}^{k} \lambda_t \rho_t(y_i|\theta_t) \tag{1.5}$$

donde cada componente es parametrizada por θ_t. Las k componentes en la mezcla son identificadas con los k grupos en la partición de consenso $P^* = \{C_1, C_2, \ldots, C_k\}$. Los coeficientes de las mezclas λ_t corresponden a las probabilidades a priori de los grupos. Todos los datos $Y = \{y_1, \ldots, y_n\}$ se asumen independientes e idénticamente distribuidos.

Esto permite representar la función de verosimilitud logarítmica con parámetros $\Theta = \{\lambda_1, \ldots, \lambda_k, \theta_1, \ldots, \theta_k\}$ dado el conjunto Y como:

$$\log L\left(\Theta|\mathrm{Y}\right) = \log \prod_{i=1}^{n} \rho\left(y_i, \Theta\right) = \sum_{i=1}^{n} \log \sum_{t=1}^{k} \lambda_t \rho_t\left(y_i|\theta_t\right) \tag{1.6}$$

La búsqueda de la partición de consenso se formula como un problema de estimación de máxima verosimilitud:

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

El problema de máxima verosimilitud (1.6) generalmente no puede ser resuelto en forma cerrada (en términos de funciones y operaciones elementales) cuando los parámetros Θ son desconocidos. Sin embargo, la función de verosimilitud (1.5) puede ser optimizada utilizando el algoritmo EM, asumiendo la existencia de datos ocultos[7] Z y la verosimilitud de los datos completos (Y, Z). Para hacer esto, se comienza con una

[7] Datos desconocidos

inicialización arbitraria de los parámetros $\{\lambda'_1, \ldots, \lambda'_k, \theta'_1, \ldots, \theta'_k\}$. Posteriormente, se aplica un proceso iterativo dado por dos pasos: Esperanza ($\mathcal{E}$) y Maximización ($\mathcal{M}$), los cuales se repiten hasta que se satisfaga algún criterio de parada.

El paso $\mathcal{E}$ calcula los valores esperados de las variables ocultas y el paso $\mathcal{M}$ maximiza la verosimilitud calculando una nueva y mejor estimación de los parámetros. Los criterios de convergencia pueden estar basados en el incremento del valor de la función de verosimilitud entre dos pasos $\mathcal{M}$ consecutivos.

La partición de consenso se obtiene mediante una simple inspección de los valores esperados de las variables ocultas $E[z_{it}]$ ya que $E[z_{it}]$ representa la probabilidad de que un patrón y_i haya sido generado por la t-ésima componente de la mezcla, la cual representa el t-ésimo grupo. Cuando se alcanza algún criterio de parada, cada etiqueta y_i se asigna a la componente que tiene el mayor valor de la variable oculta.

En este método, se asume que las etiquetas asociadas a los objetos en la partición de consenso pueden ser modeladas a partir de variables aleatorias independientes e igualmente distribuidas. Además, el número de grupos en la partición de consenso debe fijarse ya que es necesario conocer el número de componentes en el modelo de mezclas. Finalmente, el algoritmo de EM se utiliza para obtener la partición de consenso, la cual dependerá de los parámetros de dicho algoritmo y de las condiciones de parada utilizadas. Por otra parte, este método tiene una baja complejidad computacional $\mathcal{O}(k \cdot n \cdot m)$.

1.2.7. Métodos basados en la distancia de Mirkin

Dadas dos particiones P_a y P_b del mismo conjunto de objetos X, se pueden definir las siguientes cuatro categorías:

- n_{00}: Número de pares de objetos que fueron agrupados en diferentes grupos tanto en P_a como P_b.

- n_{01}: Número de pares de objetos que fueron agrupados en diferentes grupos en P_a, pero en el mismo grupo en P_b.

- n_{10}: Número de pares de objetos agrupados en el mismo grupo en P_a y en diferentes grupos en P_b.

- n_{11}: Número de pares de objetos que fueron agrupados en el mismo grupo en ambas particiones.

La *diferencia simétrica* o *distancia de Mirkin* $\mathcal{M}$ se define como $\mathcal{M}(Pa, P_b) = n_{01} + n_{10}$, la cual representa el número de desacuerdos entre ambas particiones. El problema de la partición mediana utilizando esta medida queda de la siguiente manera:

$$P^* = \arg\min_{P \in \mathbb{P}_X} \sum_{j=1}^{m} \mathcal{M}(P, P_j) \tag{1.7}$$

Como se dijo en la Sección 1.2.1, se demostró que este problema es $\mathcal{NP}$-duro. No obstante, numerosas heurísticas se han introducido para enfrentarlo, destacándose en esta dirección los trabajos de Filkov y Skiena [30], Gionis *et al.* [40], Bertolacci y Wirth [15] y Goder y Filkov [42].

Por ejemplo, la más simple de estas heurísticas es la llamada Best-of-k (BOK), la cual simplemente consiste en seleccionar la partición $P \in \mathbb{P}$ que más se aproxima a la solución del problema (1.7). En otras palabras, la salida de esta heurística es la partición en $\mathbb{P}$ que minimiza la distancia desde ella a todas las demás particiones en $\mathbb{P}$.

Best One-element Move (BOM), sigue la idea de comenzando con una partición inicial, ir haciendo cambios iterativamente mediante movimientos de objetos de un grupo a otro. De esta forma se trata de encontrar particiones mejores que la inicial. La partición inicial puede ser seleccionada al azar o puede ser, por ejemplo, el resultado del algoritmo BOK. En este caso se utiliza un algoritmo glotón. En cada paso, si se encuentra una mejor partición, esta se toma como nueva solución.

Por otra parte, CC-Pivot se basa en la idea del conocido algoritmo de ordenamiento *Quicksort*. En el *CC-Pivot*, se seleccionan repetidamente objetos *pivote* y se obtiene una partición a partir de la relación existente entre los objetos y estos pivotes. Los pivotes son usualmente seleccionados aleatoriamente, sin embargo otro tipo de heurísticas se pueden utilizar para su selección [108]. Este método tiene una baja complejidad computacional $\mathcal{O}(n \cdot m \cdot k)$.

Sin embargo, no todas las heurísticas basadas en distancia de Mirkin son eficientes, muchas de ellas tienen una complejidad cuadrática respecto al número de objetos del problema, lo que impide el uso de las mismas en grandes volúmenes de datos.

De manera general, en este tipo de métodos, la partición de consenso se obtiene mediante la solución del problema de la partición mediana usando la distancia de Mirkin (1.7). La distancia de Mirkin como medida de disimilitud entre particiones ha sido la más estudiada en el problema de la partición mediana. Sin embargo, en la práctica, esta no es la más apropiada en todas las situaciones. Por otra parte, muchas

de las heurísticas basadas en este enfoque son demasiado simples y no presentan garantías teóricas acerca de la eficacia de los resultados, mientras que otras sufren de una alto costo computacional.

1.2.8. Métodos basados en algoritmos genéticos

Estos métodos utilizan las capacidades de búsqueda de los algoritmos genéticos para obtener la partición de consenso. Generalmente, se crea la población inicial con el conjunto de particiones a combinar y se aplica una función de ajuste para determinar cuáles particiones están más cerca de la partición de consenso buscada. Después de esto, se aplican los pasos de cruzamiento y mutación para obtener una nueva descendencia y renovar la población. Durante este proceso, si se alcanza algún criterio de parada, la partición con mayor valor de función de ajuste se selecciona como partición de consenso.

Entre los métodos basados en algoritmos genéticos, uno de los más conocidos es el (Heterogeneous Clustering Ensemble; HCE) [126, 127]. En este método se obtiene la población inicial utilizando cualquier mecanismo de generación, donde con todo par de particiones obtenidas en el paso de generación se crea un par ordenado. El proceso de reproducción utiliza una función de ajuste que determina si un par de particiones (cromosoma) sobrevivirá o no en el próximo estado. En este algoritmo, la función de ajuste se obtiene por la comparación de la cantidad de solapamientos entre los grupos en ambas particiones en el cromosoma. En cada iteración, se aplica un algoritmo de cruzamiento al par de particiones que tiene un mayor valor de la función de ajuste. En este proceso de cruzamiento, se obtiene nueva descendencia a partir del par seleccionado, manteniendo en las nuevas particiones la mayor cantidad de información posible de los padres. Finalmente, las particiones padres se reemplazan por las nuevas (descendientes) y se aplica otra iteración del algoritmo completo.

Otros algoritmos de combinación de particiones basados en algoritmos genéticos fueron propuestos en [71] y [4]. De manera general, en este tipo de métodos se utilizan las capacidades de búsqueda de los algoritmos genéticos. Esto permite explorar particiones que no son fácilmente encontrables por otros métodos. Sin embargo, en estos algoritmos la solución encontrada es solo *mejor* en comparación con otra; este tipo de algoritmo carece de un concepto global de solución óptima, y de una manera de saber si una solución es óptima o no. Además, sucesivas corridas de este tipo de algoritmos puede producir resultados muy diferentes debido a la naturaleza extremadamente heurística de los mismos.

1.2.9. Métodos basados en factorización de matrices no negativas

Li *et al.* [69] introdujeron los métodos de combinación de estructuraciones basados en factorización de matrices no negativas (Non Negative Matrix Factorization; NMF). La factorización de matrices no negativas [21] se refiere al problema de factorizar una matriz no negativa[8] M en dos matrices factores, es decir, $M \approx AB$, de tal forma que tanto A como B sean no negativas [69].

En este método de combinación de agrupamientos, dado el conjunto X de n objetos y el conjunto $\mathbb{P}$ de m particiones a combinar, primero que todo se define la siguiente medida de disimilitud entre particiones:

$$\mu(P, P') = \sum_{i,j=1}^{n} \mu_{ij}(P, P') \tag{1.8}$$

donde $\mu_{ij}(P, P') = 1$ si x_i y x_j pertenecen al mismo grupo en una partición y diferente grupo en la otra, en cualquier otro caso $\mu_{ij}(P, P') = 0$.

Además, la *matriz de conectividad* se define como:

$$M_{ij}(P_v) = \begin{cases} 1, & \exists C_t^v \in P_v, \text{ tal que } x_i \in C_t^v \text{ y } x_j \in C_t^v; \\ 0, & \text{en otro caso.} \end{cases} \tag{1.9}$$

Es sencillo ver que $\mu_{ij}(P, P') = |M_{ij}(P) - M_{ij}(P')| = (M_{ij}(P) - M_{ij}(P'))^2$

La partición de consenso P^* se define a través del enfoque de la partición mediana usando μ como medida de disimilitud entre particiones.

$$P^* = \arg\min_{P \in \mathbb{P}_X} \frac{1}{m} \sum_{v=1}^{m} \mu(P, P_v) = \arg\min_{P \in \mathbb{P}_X} \frac{1}{m} \sum_{v=1}^{m} \sum_{i,j}^{n} (M_{ij}(P) - M_{ij}(P_v))^2$$

Sea $U_{ij} = M_{ij}(P^*)$ la solución a este problema de optimización, la cual es la matriz de conectividad de P^*. Este problema de optimización puede ser transformado de la siguiente manera:

$$\min_U \sum_{i,j=1}^{n} (\widetilde{M}_{ij} - U_{ij})^2 = \min_U \|\widetilde{M} - U\|_F^2$$

donde $\widetilde{M}_{ij} = \frac{1}{m} \sum_{v=1}^{m} M_{ij}(P_v)$ y $\|\cdot\|_F$ denota la norma de Frobenius.

A partir de este momento, se hacen diferentes transformaciones del problema para

[8] Matriz con todos los elementos mayores o iguales que cero.

transformarlo en el siguiente problema de optimización:

$$\min_{Q\geq 0, S\geq 0} \|\widetilde{M} - QSQ^T\|_F^2, \quad \text{sujeto a} \quad Q^TQ = I \tag{1.10}$$

donde la matriz solución U es expresada en términos de dos matrices Q y S.

El problema (1.10) puede ser solucionado utilizando el siguiente proceso de actualización de las matrices Q y S:

$$Q_{ab} \leftarrow Q_{ab}\sqrt{\frac{(\widetilde{M}QS)_{ab}}{(QQ^T\widetilde{M}QS)_{ab}}} \quad y \quad S_{bc} \leftarrow S_{bc}\sqrt{\frac{(Q^T\widetilde{M}Q)_{bc}}{(Q^TQSQ^TQ)_{bc}}}$$

mediante este proceso se obtienen las matrices Q y S, y con estas dos matrices se obtiene $U = QSQ^T$ la cual, como se vió anteriormente, es la matriz de conectividad de la partición de consenso P^*.

Este método define la partición de consenso a través del enfoque de la partición mediana, utilizando la distancia μ (1.8) como medida de proximidad entre particiones. El planteamiento original del problema se modifica consecutivamente hasta transformar el problema original en uno que puede ser resuelto mediante un proceso iterativo. Sin embargo, la solución encontrada no se corresponde con la solución del problema original y no existe un estudio detallado sobre cómo puede afectar en la calidad de la solución las modificaciones realizadas al problema original.

1.2.10. Métodos basados en estructuración difusa

Hasta ahora, se han presentado los principales métodos de combinación de agrupamientos que utilizan particiones duras de los datos en el proceso de combinación. Sin embargo, existen algoritmos de combinación de estructuraciones que trabajan con particiones difusas. Algoritmos de agrupamiento tan populares como *EM* y fuzzy-c-means [16] de manera natural producen estructuraciones difusas de los datos. Si estas particiones difusas se convierten en particiones duras para combinarse después, puede perderse información valiosa. Por lo tanto, combinar estas particiones en su variante difusa directamente puede ser más apropiado que convertirlas primero en particiones duras y posteriormente combinarlas utilizando alguno de los métodos de combinación de particiones duras presentados en las secciones anteriores [23]. La partición de consenso obtenida por algoritmos de combinación de agrupamientos difusos puede ser difusa o dura. En esta sección, solo se abordarán métodos que producen una partición

de consenso dura, es decir, las particiones difusas de los datos solo son utilizadas en pasos internos de los métodos.

Como en el caso de las particiones duras, se define el conjunto de objetos $X = \{x_1, x_2, \ldots, x_n\}$ y $\mathbb{P} = \{P_1, P_2, \ldots, P_m\}$ es el conjunto de particiones difusas de X a combinar, donde $P_i = \{S_1^i, S_2^i, \ldots, S_{d_i}^i\}$ para todo $i = 1, 2 \ldots m$. Sin embargo, en los algoritmos de combinación de particiones difusas cada S_j^i tiene asociado una función de pertenencia $\mu_{S_j^i} : X \to [0, 1]$, donde $\mu_{S_j^i}(x_r)$ es el grado de pertenencia del objeto $x_r \in X$ al j-ésimo grupo de la partición i-ésima.

Entre este tipo de métodos, se encuentran el sCSPA, sMCLA y sHBPA [85] los cuales son versiones difusas de los algoritmos CSPA, MCLA y HGBPA respectivamente (ver Sección 1.2.4).

Por ejemplo, el sCSPA extiende el método CSPA cambiando la forma de calcular la matriz de similitud. En vez de usar la matriz de co-asociación como nueva matriz de similitud entre objetos, se calcula la matriz SC de la siguiente manera: cada objeto es visto como un vector en un espacio de dimensión $\sum_{i=1}^{m} d_i$, donde d_i es la cantidad de grupos en la partición P_i y se calcula la distancia $\pi_{a,b}$ entre los objetos x_a y x_b como:

$$\pi_{a,b} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{d_i} \left(\mu_{S_j^i}(x_a) - \mu S_j^i(x_b) \right)^2} \tag{1.11}$$

Esta puede interpretarse como una medida de la diferencia en el grado de pertenencia de los objetos para cada grupo. La matriz SC se obtiene convirtiendo esta distancia en una medida de similitud donde $SC_{a,b} = e^{-\pi_{a,b}^2}$. Finalmente, se usa el algoritmo METIS para obtener la partición de consenso de igual manera que en el método CSPA.

De manera similar, existen las versiones difusas de los otros métodos basados en particionamiento de (hiper)grafos. Por tanto, estos tienen las mismas limitantes que sus versiones duras (ver Sección 1.2.4). No obstante, si para el proceso de generación se cuenta con algoritmos de agrupamiento difusos, el uso de este tipo de algoritmos de combinación de agrupamientos permite hacer una mejor modelación del problema.

1.3. Aplicaciones

El reciente progreso en combinación de agrupamientos se debe en gran medida a su empleo en diferentes campos de investigación aplicada. Existe una gran variedad de problemas en los cuales los algoritmos de combinación de agrupamientos pueden

emplearse. En principio, como los algoritmos de combinación de agrupamientos tratan de mejorar la calidad de los resultados obtenidos por algoritmos de agrupamiento, estos se pueden aplicar directamente en casi todos los problemas de clasificación no supervisada, los cuales aparecen con frecuencia en disciplinas como bioinformática, recuperación de información y minería de datos [55].

Además, existen algunos trabajos acerca de aplicaciones directas de los algoritmos de combinación de agrupamientos en varios campos de investigación como son la segmentación de imágenes [19, 32, 105, 121, 122, 129, 131]; agrupamiento de documentos: [43, 45, 97, 125]; extracción de rasgos: [51, 52]; bioinformática: [7, 24, 30, 53, 59, 128]; problemas físicos: [123]; aplicaciones médicas: [95]; entre otros.

En particular, Gionis *et al.* [40] mostraron cómo los algoritmos de combinación de agrupamientos pueden ser útiles para mejorar la robustez de los agrupamientos, para agrupar datos categóricos, para identificar el número correcto de grupos en un problema y para la detección de valores atípicos (outliers).

1.4. Consideraciones finales del capítulo

Las funciones de consenso basadas en el enfoque de la partición mediana (1.1) han sido teóricamente más estudiadas que las basadas en co-ocurrencia de objetos. Sin embargo, en ambos enfoques existen problemas sin solución definitiva, por ejemplo, en el enfoque de co-ocurrencia generalmente es necesaria la aplicación de un algoritmo de agrupamiento como paso final para encontrar la partición de consenso, pero: ¿Qué algoritmo de agrupamiento se debe aplicar? ¿Cuáles son los parámetros correctos?

En el enfoque de la partición mediana, es necesaria una medida de (di)similitud entre particiones, pero: ¿Cuál es la más apropiada? Además, la partición de consenso usualmente se define como el óptimo de un problema de optimización exponencial, sin embargo: ¿Cuál es la mejor heurística para resolver el problema o acercarse a su solución? Siguiendo cualquiera de estos enfoques de definición de la partición de consenso, es necesario crear algoritmos de combinación de agrupamientos con una adecuada fundamentación teórica, donde el proceso de combinación esté sustentado por un análisis que justifique el uso de dicho método. De manera general, el estudio presentado en este capítulo, unido a los resultados del Anexo 2, permitieron detectar un conjunto de problemas sin solución definitiva, los cuales motivaron esta investigación y a partir de los cuales se definieron los objetivos trazados, los cuales junto a las motivaciones fueron presentados en la Introducción de este documento.

Capítulo 2

COMBINACIÓN DE AGRUPAMIENTOS BASADA EN FUNCIONES NÚCLEO

En este capítulo, se presenta el enfoque de combinación de agrupamientos basado en funciones núcleo. En este enfoque la partición de consenso se define a partir de la búsqueda de la partición mediana pesada, expresada de la siguiente manera:

$$P^* = \arg\max_{P \in \mathbb{P}_X} \sum_{i=1}^{m} \omega_i \cdot k(P, P_i) \tag{2.1}$$

donde ω_i es un peso asociado a la partición P_i, y k es una medida de similitud entre particiones.

El objetivo de los pesos asignados a las particiones es *mover* la partición de consenso en la dirección de las particiones que mayor valor de peso tienen asociado. De esta manera, es necesario el desarrollo de un mecanismo de asignación de pesos automático, capaz de determinar qué particiones tienen mayor *importancia* para el proceso de combinación. En la Sección 2.1, se presenta el proceso de determinación del peso asociados a cada una de las particiones.

Por otra parte, la medida de similitud k por motivos que se verán en la Sección 2.2 es una función núcleo definida positiva[1] (positive definite kernel). Esta medida de similitud debe cumplir las siguientes dos características:

- Expresividad: Se espera que utilice *suficiente* información de ambas particiones

[1] La definición formal de este concepto será dada en la Sección 2.2. Por simplicidad, en lo adelante, estas funciones se nombrarán como *núcleos* solamente.

para decidir si estas se parecen o no. Esta propiedad se refiere a qué tan *buena* es la función k como medida de similitud entre particiones.

- Idoneidad: Se espera que de alguna manera ayude a solucionar el problema combinatorio planteado (2.1). Esta propiedad se refiere a qué tan apropiada es la medida k para la definición del problema (2.1), en el sentido de cuánto simplifica la solución del mismo.

En la Sección 2.2 se muestra cómo las funciones núcleo permiten definir medidas de similitud que satisfacen estas características. En particular se definen dos medidas de similitud, las cuales son funciones núcleo. Además, en el Anexo 4 se prueba que el índice de Rand es una función núcleo. Posteriormente, en la Sección 2.3 se presenta la función de consenso o mecanismo de combinación, el cual está basado en la utilización de una medida de similitud núcleo en la definición del problema (2.1). Finalmente, en la Sección 2.4 se presentan y discuten algunos resultados experimentales del método propuesto en este capítulo.

2.1. Análisis de la importancia de las particiones

En esta sección, se presenta un mecanismo para estimar la *importancia* de cada partición para el proceso de combinación. Esto se lleva a cabo teniendo en cuenta el cumplimiento de un conjunto de propiedades elementales dadas por un conjunto de índices de validación de agrupamientos (CVI). La idea es evaluar en qué medida cada partición satisface un conjunto de características, con el objetivo de asignarle un peso, de forma tal que este peso represente la relevancia de cada partición para el proceso de combinación. Utilizando estos pesos se puede disminuir la influencia de particiones que solo representen ruido para el proceso de combinación y aumentar la influencia de las *mejores* particiones en el proceso de determinación de la partición de consenso.

Al usar CVIs para asignar un peso a cada partición, se encuentran dos situaciones prácticas que deben ser consideradas separadamente para realizar un mejor proceso de asignación de pesos. Estas dos situaciones se nombran en este trabajo *Agrupamiento Sin Información* (ASI) y *Agrupamiento Con Información (ACI)* y se presentarán en las secciones 2.1.1 y 2.1.2 respectivamente. En ambos casos, el proceso de asignación de pesos está basado en el uso de CVIs para medir el comportamiento de cada partición respecto a las propiedades que miden estos índices. Para resaltar que estos

índices miden una propiedad, se llamarán en este documento Índices de Validación de Propiedades (IVP) y quedan definidos formalmente de la siguiente manera:

Definición 2.1. *Sea X un conjunto de objetos y $\mathbb{P}_X$ el conjunto de todas las posibles particiones de X. Un Índice de Validación de Propiedades (IVP) es una función* $\mathrm{I} : \mathbb{P}_X \to \mathbb{R}_+$, *donde para cada $P \in \mathbb{P}_X$, $I(P)$ se interpreta como el grado en que la partición P satisface la propiedad representada por I. Por lo tanto, si $P, P' \in \mathbb{P}_X$ e $I(P) > I(P')$, entonces la partición P satisface la propiedad representada por I en un mayor grado que P'.*

2.1.1. Agrupamiento sin información

Este es el caso en que los usuarios no tienen conocimiento acerca de cómo validar los resultados. El usuario tiene un problema de agrupamiento y decide aplicar una técnica de combinación de agrupamientos para enfrentarlo, pero no conoce o no tiene completa certeza de las propiedades que serán *buenas* para sus resultados. En este caso, como no se tiene información acerca de qué partición es más apropiada se opta por seguir la decisión de la mayoría. Esto significa que se asignará pequeños pesos a particiones que se comporten muy diferentes del resto. Por otra parte, a particiones con un comportamiento similar al promedio se le asignará un valor alto de peso.

En este caso, se recomienda el uso de un conjunto de IVPs que evalúe el cumplimiento de un conjunto de propiedades tan variadas como sea posible. Cualquier IVP puede ser utilizado ya que estos son usados para extraer información del comportamiento de las particiones respecto a la propiedad medida por cada índice.

Supóngase que se tiene un conjunto de l propiedades representadas por un conjunto de IVPs, $\mathbb{I} = \{I_1, I_2, \ldots, I_l\}$. Para cada índice $I_j \in \mathbb{I}$, se calcula $A_j = \sum_{i=1}^{m} I_j(P_i)$, y se define la función $\varphi_j : \mathbb{P}_X \to [0,1]$ tal que $\varphi_j(P) = \frac{I_j(P)}{A_j}$, por tanto $\sum_{i=1}^{m} \varphi_j(P_i) = 1,\ \forall j = 1, \ldots, l$. Entonces, cada φ_j puede ser relacionada con la función de distribución de cierta variable aleatoria discreta Y_j. Entonces, se define:

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

donde $H(Y_j)$ es la entropía de Y_j [22]. Usando las propiedades de la entropía, se tiene que $H(I_j) \geq 0$, $H(I_j) \leq \log|\mathbb{P}| = \log m$, y $H(I_j)$ alcanza el máximo valor cuando $\varphi_j(P_1) = \ldots = \varphi_j(P_m)$. Utilizando la propiedad de continuidad de $H(Y_j)$, se puede concluir que a mayores valores de $H(Y_j)$, mayor es el parecido entre los valores $I_j(P_i)$.

Por lo tanto, $H(I_j)$ es una buena medida para determinar qué tan informativa es la propiedad medida por el índice I_j.

Finalmente, a cada partición P_i, se le asigna un peso ω_i que viene dado por:

$$\omega_i = \sum_{j=1}^{l} \left(H\left(I_j\right) \left(1 - \left| I_j\left(P_i\right) - \frac{1}{m} A_j \right| \right) \right) \tag{2.2}$$

En esta expresión la entropía se usa como una medida de cuán discriminativo es cada índice y el segundo factor de la sumatoria es una evaluación de $I_j(P_i)$, basada en el valor absoluto de la diferencia de este valor y el valor medio $\frac{1}{m}A_j$.

2.1.2. Agrupamiento con información

Este es el caso en que el usuario conoce qué características se consideran *buenas* para los resultados. El usuario conoce un índice o un conjunto de índices[2] que desea maximizar[3]. En este caso, el peso asociado a una partición es una medida de qué tan cerca del valor máximo para cada índice está cada una de las evaluaciones de cada índice en dicha partición. Particiones con un comportamiento más similar al máximo valor para cada índice se benefician asignándoseles un alto valor de peso. De esta manera, se asignan los mayores valores de pesos a particiones con el comportamiento más parecido al esperado por los usuarios.

Dado un conjunto de IVPs $\mathbb{I} = \{I_1, I_2, \ldots, I_l\}$, para cada índice I_j se calcula $M_j = \max_{P_i \in \mathbb{P}} I_j(P_i)$. El peso asignado a cada partición P_i se obtiene mediante:

$$\omega_i = \sum_{j=1}^{l} \left(1 - |I_j(P_i) - M_j|\right) \tag{2.3}$$

En este caso el uso de la entropía no es necesario ya que no se necesita *evaluar* los índices debido a que los índices utilizados son aquellos que el usuario desea maximizar en sus resultados finales. Por tanto, todos los índices se asumen como relevantes.

De una forma u otra, en el proceso de Análisis de la Importancia de las Particiones se calcula un conjunto de pesos $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$, donde el peso ω_i está asociado a la partición P_i, y representa su importancia en el proceso de combinación de agru-

[2]El usuario podría asignarle un peso a cada índice para diferenciarlos de acuerdo a su importancia. Por simplicidad se asume que todos estos pesos son iguales a 1.

[3]Se asume que el usuario quiere maximizar los índices, porque en la práctica si el usuario tiene un índice que desea minimizar, este puede ser fácilmente transformado en su inverso u opuesto el cual debe ser maximizado, preservando la misma semántica del índice inicial.

pamientos. En un problema práctico, para aumentar la eficacia de este mecanismo de asignación de pesos es importante determinar a qué situación (ASI o ACI) se ajusta más el problema.

2.2. Medidas de similitud entre particiones

Como se puede apreciar en la ecuación (2.1), la medida de similitud entre particiones k es una pieza fundamental en la definición de la partición de consenso.

Entre las medidas de similitud existentes, los *productos internos* son medidas que tienen un gran atractivo matemático. Dados dos vectores x, x' (normalizados a longitud 1), el producto interno de éstos $\langle x, x' \rangle$ puede interpretarse geométricamente como el coseno del ángulo entre los mismos. De igual manera, un producto interno permite calcular la *longitud* o *norma* de un vector de la siguiente manera $\|x\| = \sqrt{\langle x, x \rangle}$. Además, la distancia entre los mismos puede calcularse como la norma del vector diferencia. Por tanto, poder calcular productos internos permite la utilización de las herramientas matemáticas que pueden formularse en términos de ángulos, longitudes y distancias.

Sin embargo, los productos internos son un subconjunto muy limitado del conjunto de todas las posibles medidas de similitud entre objetos. Para definir una medida de similitud que sea un producto interno, es necesario que los objetos estén representados como elementos de un espacio vectorial. Lo cual, en ocasiones no es posible, las características de los objetos no se ajustan a los requerimientos de un espacio vectorial, es decir, forzar que los objetos sean representados como elementos de un espacio vectorial conllevaría a una mala modelación del problema. Este es, por ejemplo, el caso en cuestión, donde se necesitan crear medidas de similitud entre particiones, sin embargo las particiones no son, de manera natural, elementos de un espacio vectorial.

Uno de los grandes atractivos que representa utilizar una función núcleo como medida de similitud entre objetos es que esta permite calcular la similitud como un producto interno en un cierto espacio vectorial asociado al núcleo [14]. Esto se cumple para cualquier tipo de datos, es decir, datos en los cuales su dominio de definición no tiene asociado ninguna estructura, solamente se asume la existencia de un conjunto no vacío de objetos. Luego, las funciones núcleo permiten generalizar las bondades de los productos internos a problemas donde una representación vectorial de los objetos no es posible o no es apropiada.

Formalmente, un núcleo se define de la siguiente manera [94]:

Definición 2.2. *Sea $\mathcal{X}$ un conjunto no vacío. Una función $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ simétrica se llama* núcleo *(*definido positivo[4]*) si $\forall t \in \mathbb{N}$, todo $\mathtt{x}_1, \ldots, \mathtt{x}_t \in \mathcal{X}$ y toda secuencia de números reales $\alpha_1, \ldots, \alpha_t \in \mathbb{R}$ se cumple que:*

$$\sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i \alpha_j k(\mathtt{x}_i, \mathtt{x}_j) \geq 0 \tag{2.4}$$

Si $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ es un núcleo, existe una función $\phi : \mathcal{X} \to \mathcal{H}$ de $\mathcal{X}$ en cierto espacio de Hilbert $\mathcal{H}$ tal que:

$$k(\mathtt{x}, \mathtt{x}') = \langle \phi(\mathtt{x}), \phi(\mathtt{x}') \rangle_{\mathcal{H}}$$

para todo $\mathtt{x}, \mathtt{x}' \in \mathcal{X}$ ($\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denota el producto interno en el espacio de Hilbert $\mathcal{H}$). Este espacio de Hilbert $\mathcal{H}$ es conocido como *espacio de Hilbert con núcleo reproductor* (Reproducing Kernel Hilbert Space; RKHS)[5] [6, 92] y viene dado por la clausura topológica del conjunto de todas las posibles combinaciones lineales de funciones $k(\mathtt{x}, \cdot)$

$$\mathcal{H} = \overline{span\left\{k\left(\mathtt{x}, \cdot\right) \ / \ \mathtt{x} \in \mathcal{X}\right\}}$$

donde $k(\mathtt{x}, \cdot)$ es una función de $\mathcal{X}$ en $\mathbb{R}$ definida para cada $\mathtt{x} \in \mathcal{X}$, la cual asocia cada objeto $\mathtt{x}'$ con el número real $k(\mathtt{x}, \mathtt{x}')$.

A continuación, se expone cómo una medida de similitud que sea una función núcleo puede ayudar a solucionar el problema (2.1), en otras palabras, cómo influye el hecho de que una medida de similitud sea un núcleo en la *idoneidad* de dicha medida para el problema de la partición mediana pesada (2.1).

Sea $k : \mathbb{P}_X \times \mathbb{P}_X \to \mathbb{R}$ una función núcleo definida sobre el conjunto de todas las posibles particiones y supóngase que esta es una función núcleo normalizada, i.e., $\sqrt{k(P,P)} = \sqrt{\langle \phi(P), \phi(P) \rangle_{\mathcal{H}}} = \|\phi\left(P\right)\|_{\mathcal{H}} = 1$, $\forall P \in \mathbb{P}_X$. Se puede asumir sin perder generalidad que k es normalizada ya que para una función núcleo $\tilde{k}$ cualquiera se puede definir su versión normalizada de la siguiente forma $k\left(P_i, P_j\right) = \frac{\tilde{k}(P_i,P_j)}{\sqrt{\tilde{k}(P_i,P_i)\tilde{k}(P_j,P_j)}}$. De esta manera, dado el problema (2.1) se puede considerar el problema equivalente en

[4] Aquí se sigue la terminología utilizada en [94] donde la ecuación (2.4) se asocia al término *definido positivo*. Por otra parte, el caso en que el valor 0 solo se obtiene si $\alpha_1 = \ldots = \alpha_t = 0$ se nombra *estrictamente definido positivo*.

[5] También se nombrará a este espacio simplemente como espacio de Hilbert.

el espacio de Hilbert $\mathcal{H}$ asociado a k:

$$\phi\left(P^*\right) = arg \max_{\phi(P)\in\mathcal{H}} \sum_{i=1}^{m} \omega_i \langle \phi\left(P\right), \phi\left(P_i\right)\rangle_{\mathcal{H}} \tag{2.5}$$

y usando las propiedades del producto interno, este puede ser reescrito como:

$$\phi\left(P^*\right) = arg \max_{\phi(P)\in\mathcal{H}} \left\langle \phi\left(P\right), \sum_{i=1}^{m} \omega_i \phi(P_i) \right\rangle_{\mathcal{H}} \tag{2.6}$$

Como ϕ es la aplicación asociada al núcleo normalizado k, para cada partición P, se tiene que $\|\phi\left(P\right)\|_{\mathcal{H}} = 1$. Por lo tanto, con el objetivo de simplificar el problema de optimización (2.1), se considera la versión normalizada de los pesos ω_i':

$$\omega_i' = \frac{\omega_i}{\left\|\sum_{j=1}^{m} \omega_j \phi(P_j)\right\|_{\mathcal{H}}} \tag{2.7}$$

De esta manera $\|\sum_{i=1}^{m} \omega_i' \phi\left(P_i\right)\|_{\mathcal{H}} = 1$, y finalmente el problema se reduce a:

$$\phi\left(P^*\right) = arg \max_{\phi(P)\in\mathcal{H}} \left\langle \phi\left(P\right), \sum_{i=1}^{m} \omega_i' \phi(P_i) \right\rangle_{\mathcal{H}} \tag{2.8}$$

Proposición 2.1. *Si ψ es una solución del problema de optimización (2.8), entonces*

i) Existen escalares $\alpha_1, \alpha_2, \ldots, \alpha_m \in \mathbb{R}$, tal que $\psi = \sum_{i=1}^{m} \alpha_i \phi(P_i)$

ii) Para todo $i = 1, \ldots, m$, $\alpha_i = \omega_i'$.

Demostración. Sea $\mathcal{H} = \mathcal{H}_1 \bigoplus \mathcal{H}_1^{\perp}$, donde $\mathcal{H}_1$ es la clausura topológica del conjunto $span\left\{\phi\left(P_1\right), \phi\left(P_2\right), \ldots, \phi\left(P_m\right)\right\}$ y $\mathcal{H}_1^{\perp} = \{z \in \mathcal{H} : \langle z, y_1\rangle_{\mathcal{H}} = 0,\ \forall y_1 \in \mathcal{H}_1\}$ es el complemento ortogonal de $\mathcal{H}_1$ en $\mathcal{H}$. Sea ψ una solución del problema de optimización (2.8), entonces $\psi = y_1 + y_1^{\perp}$ con $y_1 \in \mathcal{H}_1$ y $y_1^{\perp} \in \mathcal{H}_1^{\perp}$.

$$\left\langle \psi, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\rangle_{\mathcal{H}} = \left\langle y_1, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\rangle_{\mathcal{H}} + \left\langle y_1^{\perp}, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\rangle_{\mathcal{H}}$$

Considerando que $y_1^{\perp} \in \mathcal{H}_1^{\perp}$ y $\sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \in \mathcal{H}_1$, se tiene que:

$$\left\langle y_1^{\perp}, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\rangle_{\mathcal{H}} = 0$$

y por lo tanto $\langle \psi, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right)\rangle_{\mathcal{H}} = \langle y_1, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right)\rangle_{\mathcal{H}}$. Esto significa que, si ψ es

una solución del problema de optimización (2.8), entonces su proyección y_1 sobre el subespacio $\mathcal{H}_1$ es también solución de este problema.

Ahora, utilizando la desigualdad de Cauchy-Buniakovski [33] se tiene que:

$$\left\langle \psi, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\rangle_{\mathcal{H}} \leq \|\psi\|_{\mathcal{H}} \left\| \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\|_{\mathcal{H}}$$

Pero,

$$\|\psi\|_{\mathcal{H}} = \left\| \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\|_{\mathcal{H}} = 1,$$

luego,

$$\left\langle \psi, \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\rangle_{\mathcal{H}} \leq 1.$$

Como la igualdad en la desigualdad de Cauchy-Buniakovski se obtiene solamente cuando los vectores son linealmente dependientes, se cumple que:

$$\psi = \alpha \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) = y_1$$

para algún $\alpha \in \mathbb{R}_+$. Entonces

$$\|\psi\|_{\mathcal{H}} = |\alpha| \left\| \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right) \right\|_{\mathcal{H}} = \|y_1\|_{\mathcal{H}}$$

$$1 = |\alpha| \cdot 1$$

Se concluye que:

$$\psi = \sum_{i=1}^{m} \omega_i' \phi\left(P_i\right)$$

con lo que la demostración queda probada. □

Una vez que se tiene la solución en el espacio de Hilbert $\mathcal{H}$, solo se necesita resolver el problema de la pre-imagen. Este problema consiste en encontrar la partición P^* en el espacio de las particiones dada la solución $\psi = \phi\left(P^*\right)$ en el espacio de Hilbert. Cuando se trabaja con datos estructurados, por ejemplo, árboles, grafos o como en este caso, particiones, este problema puede ser complejo y obtener la solución exacta puede ser una tarea difícil desde el punto de vista computacional, debido al problema de optimización combinatorio que tiene asociado. De hecho, dada $\psi \in \mathcal{H}$, la solución

exacta $P^* \in \mathbb{P}_X$ tal que $\psi = \phi(P^*)$ no tiene por qué existir, dado que $\mathcal{H}$ es usualmente un espacio mucho más grande que $\mathbb{P}_X$. Por estas razones, se escoge P^* como a una solución aproximada del problema de la pre-imagen:

$$P^* = arg\min_{P \in \mathbb{P}_X} \|\phi(P) - \psi\|^2_{\mathcal{H}} \tag{2.9}$$

donde $\|\phi(P) - \psi\|^2_{\mathcal{H}}$ se calcula solamente en términos del producto interno de la siguiente manera:

$$\|\phi(P) - \psi\|^2_{\mathcal{H}} = \langle\phi(P), \phi(P)\rangle_{\mathcal{H}} - 2\sum_{i=1}^{m} \omega'_i \langle\phi(P), \phi(P_i)\rangle_{\mathcal{H}} + \sum_{i=1}^{m}\sum_{j=1}^{m} \omega'_i\omega'_j \langle\phi(P_i), \phi(P_j)\rangle_{\mathcal{H}}$$

Como k es un núcleo normalizado y como los pesos ω'_i son también normalizados según la ecuación (2.7), el primer y el tercer término del miembro derecho de esta ecuación son iguales a 1. Luego esta ecuación queda de la forma:

$$\|\phi(P) - \psi\|^2_{\mathcal{H}} = 2 - 2\sum_{i=1}^{m} \omega'_i \langle\phi(P), \phi(P_i)\rangle_{\mathcal{H}}$$

Finalmente, utilizando la propiedad $k(P_i, P_j) = \langle\phi(P_i), \phi(P_j)\rangle_{\mathcal{H}}$, se obtiene que

$$\|\phi(P) - \psi\|^2_{\mathcal{H}} = 2 - 2\sum_{i=1}^{m} \omega'_i k(P, P_i) \tag{2.10}$$

La ecuación (2.10) es de gran importancia para el desarrollo del algoritmo propuesto en este capítulo. Como se puede apreciar, esta permite calcular, de manera simple, la distancia entre la imagen de cualquier partición $\phi(P), \forall P \in \mathbb{P}_X$ y el consenso ψ en el espacio de Hilbert. Esta ecuación nos brinda una medida global que permite decidir qué tan cerca o lejos se encuentra cualquier partición de la partición de consenso. El atractivo computacional de este método es que todo el trabajo en el espacio de Hilbert $\mathcal{H}$ se hace de manera implícita, es decir, no es necesario conocer ϕ, $\mathcal{H}$, ni calcular explícitamente productos internos en $\mathcal{H}$, sino que en la práctica todos los cálculos quedan en función de k. Esto es conocido en la literatura como kernel trick [94], lo cual es fundamental para el desarrollo de métodos basados en funciones núcleo para la solución de problemas de reconocimiento de patrones [2, 93, 109].

En las secciones 2.2.1 y 2.2.2 se definen dos medidas de similitud para las cuales se prueba que son funciones núcleo y en el Anexo 4 se presenta la demostración de que

el índice de Rand es una función núcleo. En la Sección 2.3, se presenta el algoritmo de combinación propuesto el cual está basado en los resultados obtenidos en esta sección.

2.2.1. Similitud basada en representación por grafos

La medida de similitud propuesta en esta sección está basada en la representación de las particiones mediante grafos. Para cada partición $P \in \mathbb{P}_X$, se define un grafo $G_P = (V, E)$, donde $V = \{v_1, \ldots, v_n\}$ tal que v_i es el nodo asociado al objeto x_i. Además, existe una arista e_{ij} entre los nodos v_i y v_j si los objetos x_i y x_j pertenecen al mismo grupo en P. De esta manera, por cada partición $P \in \mathbb{P}_X$ se obtiene un grafo donde cada grupo de P se representa por una componente conexa en el grafo, la cual es además un grafo completo.

Sea $\mathcal{G} : \mathbb{P}_X \to \mathbb{G}_X$ donde $\mathbb{G}_X$ es el espacio de todos los posibles grafos obtenidos a partir de una partición de X. No es difícil verificar que $\mathcal{G}$ es una función biyectiva, por tanto, trabajar en $\mathbb{P}_X$ es equivalente a trabajar en $\mathbb{G}_X$.

La medida de similitud propuesta está basada en la idea de utilizar variables ocultas[6] [50] para medir la similitud entre grafos. En este caso, se toman como variables ocultas los caminos (paths) sobre cada grafo. Dado un grafo $G = (V, E) \in \mathbb{G}_X$, se define como $\Sigma(G)$ el conjunto de todos los caminos del grafo G. Un camino $h \in V^l$ es una secuencia de nodos $h_1 h_2 \ldots h_l$, donde existe una arista entre todo par de nodos consecutivos (h_i, h_{i+1}) y $h_i \neq h_j$, $\forall i, j = 1, \ldots, l$ con $i \neq j$, donde l es la longitud del camino. Basado en la idea de los núcleos marginalizados para grafos [103] se define la siguiente medida de similitud entre grafos k_G:

$$k_G(G, G') = \sum_{h \in \Sigma(G)} \sum_{h' \in \Sigma(G')} \delta(h, h') \rho(h/G) \rho(h'/G') \tag{2.11}$$

donde $\delta(h, h') = 1$ si $h = h'$ y cero en otro caso. Por otra parte, $\rho(h/G)$ viene dado por:

$$\rho(h/G) = \rho_s(h_1) \left(\prod_{i=2}^{l} \rho_t(h_i/h_{i-1}) \right) \rho_e(h_l)$$

donde $\rho_s(h_1)$ representa la probabilidad de que el camino empiece en el vértice h_1, $\rho_t(h_i/h_{i-1})$ la probabilidad de estando en el vértice h_{i-1} moverse al vértice h_i y $\rho_e(h_l)$

[6] La similitud entre dos objetos se calcula como la suma de las similitudes de estructuras más simples que componen a cada objeto. Por ejemplo, la similitud entre dos grafos puede calcularse a partir de la suma de las similitudes entre los subgrafos, árboles o caminos que componen a cada grafo.

es la probabilidad de terminar en el vértice h_l. En este caso, $\rho_s(h_1) = \frac{1}{n}$ ya que se asume que todos los nodos tienen la misma probabilidad de ser tomados como primer nodo del camino y $\rho_t(h_i/h_{i-1}) = \rho_e(h_l) = \frac{1}{|C_h|}$, donde $|C_h|$ representa la cantidad de nodos en la componente conexa que contiene a h. Estos valores son calculados teniendo en cuenta que estando en un nodo h_i existen $|C_h|$ posibilidades, moverse a uno de los $|C_h| - 1$ nodos vecinos o quedarse en el mismo nodo y terminar el camino ahí mismo. Asumiendo que todas estas posibilidades son equiprobables se obtienen los valores ρ_t y ρ_e.

Proposición 2.2. *La función k_G es un núcleo definido positivo.*

Demostración. La simetría de k_G es fácil de verificar a partir de la definición de la misma en la ecuación (2.11). Por otra parte, esta medida puede escribirse de manera equivalente para dos grafos G_i, G_j de la siguiente forma:

$$k_G(G_i, G_j) = \sum_{h \in \Sigma(\mathbb{G}_X)} \delta(h/G_i)\delta(h/G_j)\rho(h/G_i)\rho(h/G_j)$$

donde $\Sigma(\mathbb{G}_X)$ es el conjunto de todos los caminos de los grafos de $\mathbb{G}_X$ y $\delta(h/G) = 1$, si $h \in \Sigma(G)$ e igual a 0 en otro caso. De esta manera, para probar que k_G es un núcleo sólo faltaría probar que para todo $t \in \mathbb{N}$, $\alpha_1, \alpha_2, \ldots, \alpha_t \in \mathbb{R}$ y para todo $G_1, G_2, \ldots, G_t \in \mathbb{G}_X$ se cumple que:

$$\sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i\alpha_j k_G(G_i, G_j) \geq 0$$

$$\sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i\alpha_j \left(\sum_{h \in \Sigma(\mathbb{G}_X)} \delta(h/G_i)\delta(h/G_j)\rho(h/G_i)\rho(h/G_j) \right) \geq 0$$

$$\sum_{h \in \Sigma(\mathbb{G}_X)} \left(\sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i\alpha_j \delta(h/G_i)\delta(h/G_j)\rho(h/G_i)\rho(h/G_j) \right) \geq 0$$

donde agrupando los términos en la suma quedaría:

$$\sum_{h \in \Sigma(\mathbb{G}_X)} \left(\sum_{i=1}^{t} \alpha_i \delta(h/G_i)\delta(h/G_j) \right)^2 \geq 0$$

con lo que queda demostrada la proposición. □

Esta similitud puede considerarse suficientemente *expresiva* como medida de simi-

litud entre este tipo de grafos (consecuentemente entre particiones), ya que se tienen en cuenta todos los caminos comunes entre los dos grafos a la hora de determinar si estos son similares o no. Por otra parte, esta similitud se puede calcular de manera eficiente (en $\mathcal{O}(n)$) ya que desde el punto de vista computacional no es necesario analizar todos los caminos de ambos grafos, pues todos los caminos con la misma longitud que están en la misma componente conexa tienen el mismo valor $\rho(h/G)$[7].

2.2.2. Similitud basada en conteo de subconjuntos

En esta sección se presenta una nueva medida de similitud entre particiones, en la cual no es necesaria la representación auxiliar de las particiones a través de grafos. Al igual que en la medida anterior, la similitud entre particiones se mide a partir del uso de variables ocultas, las cuales son una vía común para definir una función núcleo para datos estructurados tales como cadenas (strings), árboles y grafos [39, 60]. En este caso, se utilizan los subconjuntos del conjunto de objetos X como variables ocultas en las particiones.

Para cada subconjunto $S \subseteq X$, interesa medir cuán relevante es este para la partición P. Intuitivamente, se dice que S es más relevante si es más similar a un grupo C de P. Por tanto, $S \subseteq X$ es *relevante* para la partición P, si existe un grupo C de P tal que la *diferencia simétrica* entre S y C es pequeña, i.e., $|S \backslash C| + |C \backslash S|$ es pequeño[8].

Si $S \subseteq C$ para algún grupo $C \in P$, formalmente se define la *relevancia básica* de S dado P como $\mu_B(S \mid P) = \frac{|S|}{|C|}$. Esta medida es fácil de interpretar y satisface las siguientes tres propiedades:

1. $0 \leq \mu_B(S|C) \leq 1$, $\mu_B(S|C) \to 0$ cuando $S \to \emptyset$ y $\mu_B(S|C) \to 1$ cuando $S \to C$. En otras palabras, para una secuencia $S_1 \subseteq \cdots \subseteq S_t \subseteq \cdots \subseteq C$ se cumple que $\mu_B(S_1|C) \leq \cdots \leq \mu_B(S_t|C) \leq \cdots \leq 1$.

2. Si $S = S_1 \cup S_2 \subseteq C$ para algún grupo $C \in P$ y $S_1 \cap S_2 = \emptyset$, entonces:

$$\begin{aligned} \mu_B(S|P) &= \frac{|S|}{|C|} = \frac{|S_1 \cup S_2|}{|C|} = \frac{|S_1| + |S_2|}{|C|} = \\ &= \mu_B(S_1|P) + \mu_B(S_2|P) \end{aligned}$$

[7] En la próxima sección se presenta otra medida de similitud la cual también puede calcularse en $\mathcal{O}(n)$. Para esta otra similitud se hace un análisis más detallado de como se calcula y se presenta el pseudo-código del algoritmo. Las ideas utilizadas en ese análisis permiten ver con mayor claridad como es posible calcular en tiempo lineal la similitud presentada en esta sección.

[8] $|S \backslash C| = \{x \mid x \in S \;\; y \;\; x \notin C\}$.

3. Si $S_1 \subseteq C$, $S_2 \subseteq C$, $|S_1| = |S_2|$ para algún grupo $C \in P$, entonces:

$$\mu_B(S_1|P) = \mu_B(S_2|P).$$

La segunda propiedad dice que la relevancia de un conjunto $S \subseteq C$ puede ser medida como la suma de la relevancia de subconjuntos disjuntos de S tal que cada subconjunto está contenido en el mismo grupo de P. Siguiendo este razonamiento, en el caso que S no está contenido en un grupo de P, se tiene que $S = (S \cap C_1) \cup \cdots \cup (S \cap C_d)$, donde cada C_i es un grupo de P, $\forall i = 1 \ldots d$, y d es el número de grupos en P. Entonces, se estima la relevancia de un conjunto S respecto a la partición P como:

$$\mu_B(S|P) = \sum_{i=1}^{d} \mu_B(S \cap C_i|P) = \sum_{i=1}^{d} \frac{|S \cap C_i|}{|C_i|}$$

La *relevancia básica* no es la única manera de medir la relevancia de un conjunto S. En un problema particular, es posible que grandes subconjuntos sean más importantes que los pequeños o viceversa, entonces la relevancia apropiada no debe ser lineal respecto a la fracción $\frac{|S|}{|C|}$. De esta forma, se da la siguiente definición con el objetivo de generalizar este concepto:

Definición 2.3. *Sea X un conjunto de objetos. La función $\mu : 2^X \times \mathbb{P}_X \to \mathbb{R}_+$ es una medida de relevancia de un subconjunto $S \subseteq X$, si se satisfacen las siguientes condiciones:*

i) $\forall S \subseteq X \left[\mu(S|P) = \sum_{C \in P} \mu(S \cap C|P)\right]$

ii) $\forall C \in P \ [S_1 \subseteq S_2 \subseteq C \Rightarrow \mu(S_1|P) \leq \mu(S_2|P)]$

iii) $\forall S_1, S_2 \subseteq C \ [(|S_1| = |S_2|) \Rightarrow (\mu(S_1|P) = \mu(S_2|P))]$

iv) $\mu(S|P)$ *es una función acotada.*

Esta medida puede ser definida mediante la composición de cierta función $f : [0, 1] \to \mathbb{R}_+$ y la *relevancia básica* μ_B, donde f es una función que permite modificar la linealidad de μ_B respecto a $\frac{|S|}{|C|}$, y mantiene el cumplimiento de las propiedades $i) - iv)$.

Ahora, se introduce una medida de similitud entre particiones.

Definición 2.4. *Sea X un conjunto de objetos y sea $\mathbb{P}_X$ el conjunto de todas las posibles particiones de dicho conjunto. La función $k_S : \mathbb{P}_X \times \mathbb{P}_X \to \mathbb{R}_+$ tal que:*

$$k_S(P_i, P_j) = \sum_{S \subseteq X} \delta_S^{P_i} \delta_S^{P_j} \mu(S|P_i)\mu(S|P_j)$$

es una medida de similitud entre particiones, donde

$$\delta_S^P = \begin{cases} 1, & si\ \exists C \in P\ S \subseteq C \\ 0, & en\ otro\ caso \end{cases}$$

Proposición 2.3. *La función k_S definida anteriormente es una función núcleo.*

Demostración. Para probar que k_S es un núcleo, es suficiente probar que, para todo subconjunto finito de $\mathbb{P}_X$, $\{P_1, P_2, \ldots, P_t\}$ y para toda sucesión finita $\alpha_1, \alpha_2, \ldots, \alpha_t$ de números reales[9]:

$$\sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i \alpha_j k_S(P_i,\ P_j) \geq 0$$

En este caso, la parte izquierda de la expresión puede ser reescrita como:

$$\sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i \alpha_j \left(\sum_{S \subseteq X} \delta_S^{P_i} \delta_S^{P_j} \mu(S|P_i)\mu(S|P_j) \right)$$

$$= \sum_{S \subseteq X} \left(\sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i \alpha_j \delta_S^{P_i} \delta_S^{P_j} \mu(S|P_i)\mu(S|P_j) \right)$$

agrupando los términos en esta suma, se obtiene:

$$= \sum_{S \subseteq X} \left(\sum_{i=1}^{t} \alpha_i \delta_S^{P_i} \mu(S|P_i) \right)^2 \geq 0$$

y la proposición queda probada. □

A primera vista, si se tiene dos particiones P_1 y P_2, el cálculo de esta similitud parece ser muy costoso computacionalmente, debido al cálculo de la relevancia de todos los posibles conjuntos de X para ambas particiones. Sin embargo, en la definición 2.4, se puede ver que los subconjuntos S tales que $S \subseteq C_i^1$ y $S \subseteq C_j^2$ simultáneamente, para algún $C_i^1 \in P_1$ y $C_j^2 \in P_2$ son los únicos que contribuyen a la similitud con un valor mayor que cero. Si se considera $\mathrm{L} = \{L_1, L_2, \ldots, L_v\}$ el conjunto de todas las posibles intersecciones de un grupo de P_1 con un grupo de P_2 entonces, solamente es necesario analizar los elementos de L y sus subconjuntos. De la definición 2.3 puede verse que la función de relevancia $\mu(S|P)$ solamente depende de la cardinalidad de un subconjunto S y de la cardinalidad del grupo $C \in P$ tal que $S \subseteq C$, por tanto,

[9] Además, habría que probar que la función es simétrica, pero en este caso la simetría es fácilmente verificable a partir de la definición de la función.

usando la tercera propiedad definida anteriormente para funciones de relevancia, se puede ver que todos los subconjuntos de $L_i \in \mathrm{L}$ con la misma cardinalidad tienen el mismo valor de relevancia. Como hay $\binom{|C|}{|S|} = \frac{(|C|)!}{(|S|)!\ (|C|-|S|)!}$ subconjuntos diferentes con cardinalidad $|S|$ en C, todos ellos tienen el mismo valor de relevancia, y pueden ser calculados una sola vez. Por esta razón, solamente es necesario calcular n valores diferentes de relevancia para calcular la similitud entre dos particiones con n objetos. En Algoritmo 1 se encuentra el pseudo código del algoritmo para calcular la similitud entre dos particiones.

Algoritmo 1: Algoritmo para calcular la similitud entre dos particiones

Entrada: $P_1 \in \mathbb{P}_X, \quad P_2 \in \mathbb{P}_X$
Salida: Valor de similitud entre P_1 y P_2: $\Gamma = k_S\left(P_1,\ P_2\right)$
Inicialización: $\Gamma = 0$

Calcular $\mathrm{L} = \{L_1, L_2, \ldots, L_v\}$ el conjunto de todas las intersecciones de un grupo de P_1 con un grupo de P_2.
forall $L_i = C_i^1 \cap C_j^2 \in \mathrm{L}$ **do**
  for $s = 1$ *to* $s = |L_i|$ **do**
    $\Gamma = \Gamma + \binom{|L_i|}{s} \cdot \frac{s}{|C_i^1|} \frac{s}{|C_j^2|}$

A pesar de la existencia de dos ciclos anidados en este pseudo código, el costo computacional del algoritmo es $\mathcal{O}(n)$, ya que $\sum_{i=1}^{v} |L_i| = |X| = n$, esto es fácil de verificar teniendo en cuenta que L es también una partición[10] del conjunto de objetos X.

2.3. Función de consenso

El mecanismo de combinación propuesto en esta sección está basado en el hecho de que la función de similitud utilizada para definir el problema de la partición mediana pesada (2.1) es una función núcleo k cualquiera. En la práctica, pueden ser utilizadas la similitud basada en representación por grafos k_G (Sección 2.2.1), la similitud basada en conteo de subconjuntos k_S (Sección 2.2.2), el índice de Rand k_R (Anexo 4) u otra función núcleo.

El resultado obtenido en la ecuación (2.10) permite obtener una medida global de qué tan cercana está cualquier partición P de la solución del problema (2.1). Esto

[10] De hecho, es el ínfimo $(P_1 \wedge P_2)$ de las particiones P_1 y P_2 en el retículo asociado a $\mathbb{P}_X$.

permite, partiendo de una partición inicial, definir un proceso iterativo para aproximarse a la solución del problema, donde cada solución parcial es más cercana al consenso, que la anterior. Este proceso se repite hasta que no se encuentre una solución mejor o se cumpla cualquier otro criterio de parada del algoritmo. Esto significa que la búsqueda concluye en un óptimo local el cual no tiene por qué ser el óptimo global. Una variante para enfrentar el problema de una posible convergencia temprana a un óptimo local de baja calidad, es permitir movimientos a estados peores de manera controlada. Esta es la idea subyacente en la meta-heurística *recocido simulado* (simulated annealing) [61], donde estos movimientos a estados peores son controlados por una función de probabilidad, la cual toma valores más pequeños a medida que avanza la búsqueda haciendo menos frecuentes este tipo de movimientos.

Para aplicar el *recocido simulado* en un problema concreto es necesario definir un conjunto de parámetros. La *energía* y la *temperatura* son dos parámetros importantes en el *recocido simulado*. Estos términos deben su nombre al proceso de recocido en metalurgia, a partir del cual está inspirado esta meta-heurística. La energía es una medida de qué tan bueno es el estado analizado y el objetivo del proceso es encontrar un estado con la menor cantidad de energía posible. La temperatura es un parámetro global, el cual controla los movimientos a estados con un aumento de energía. A lo largo de este proceso, la temperatura debe ir decreciendo, haciendo más difícil el cambio a un estado con mayor valor de energía.

Además, es necesario definir: el *estado inicial*, la *función de costo* (la función que calcula la energía), la *función de generación de vecinos* y la *función de reducción de la temperatura*.

En este caso, los estados del sistema son particiones, y la idea consiste en comenzar con una partición inicial, y a través de un proceso iterativo, obtener particiones más cercanas al consenso. En el método propuesto, se puede definir como estado inicial P_0 la partición en $\mathbb{P}$ con un valor mínimo de distancia a ψ. En otras palabras

$$P_0 = arg\min_{P \in \mathbb{P}} \|\phi(P) - \psi\|_{\mathcal{H}}^2$$

Esto significa que se comienza con la mejor partición en el conjunto de particiones obtenidas en el paso de generación. Por otra parte, se define la *función de costo* E para cada estado P como:

$$E(P) = \|\phi(P) - \psi\|_{\mathcal{H}}^2$$

La mejor manera de evaluar un estado (partición) es calculando la distancia de su

imagen a la imagen de la *partición de consenso teórica*[11] ψ.

Se dice que dos estados (particiones) P_a y P_b son vecinos, si se puede obtener P_b moviendo un objeto de P_a de un grupo hacia otro[12] y viceversa. Luego, se define la vecindad de un estado (partición) P como

$$N(P) = \{P' \in \mathbb{P}_X \mid P' \text{ es vecino de } P\}$$

En este caso, se usa un programa de reducción de temperatura muy simple en el cual la temperatura $\mathcal{T}$ decrece de la siguiente manera: $\mathcal{T}_i{=}\beta\mathcal{T}_{i-1}$, donde $\mathcal{T}_i$ es la temperatura del i-ésimo estado del proceso y $\beta < 1$ es una constante. El pseudo código de este algoritmo para enfrentar el problema de la pre-imagen y por tanto para hallar la partición de consenso se presenta en el Algoritmo 2.

La complejidad computacional de esta función de consenso viene dada por la complejidad del *recocido simulado*, el cual es utilizado como heurística de solución. Este problema básicamente consiste en solucionar el problema (2.9) utilizando la ecuación (2.10).

Para resolver el problema (2.9) primeramente es necesario calcular los pesos normalizados ω'_i, para los cuales es necesario calcular $\left(\sum_{i=1}^{m}\sum_{j=1}^{m}\omega_i\omega_j\tilde{k}(P_i,P_j)\right)$ el cual es un valor constante y puede ser obtenido una sola vez en tiempo $\mathcal{O}(n\cdot m^2)$, ya que es necesario aplicar $\mathcal{O}(m^2)$ veces la medida de similitud k, la cual puede ser calculada en $\mathcal{O}(n)$ (ver Sección 2.2)[13].

Posteriormente, en cada iteración del algoritmo, se debe generar un nuevo estado vecino. Esto significa, crear una nueva partición a partir del estado actual, la cual difiera en solo una etiqueta de cluster para un objeto de la partición actual. Esto puede ser hecho en $\mathcal{O}(1)$. Además, se tiene que calcular la energía del estado actual (calcular la distancia de la partición actual P hasta la partición de consenso teórica) mediante la ecuación (2.10). Esto implica hacer m aplicaciones de la medida de similitud, la cual puede ser calculada en $\mathcal{O}(n)$. Por tanto, esta distancia puede ser calculada en $\mathcal{O}(n\cdot m)$. De esta manera, se puede decir que calcular cada iteración del *recocido simulado* es $\mathcal{O}(n\cdot m)$ y como se puede asumir que $m < n$, este es inferior a n^2.

El proceso de *recocido simulado* tiene como criterio de parada un máximo número de iteraciones $rMax$, luego se puede estimar el costo computacional del proceso completo en $\mathcal{O}(n\cdot m^2)+\mathcal{O}(n\cdot m\cdot rMax)$. Asumiendo que se hacen m o más iteraciones

[11] Por simplicidad, se denomina a ψ *consenso teórico*.

[12] O creando un nuevo grupo unitario con este elemento.

[13] Las tres medidas de similitud entre particiones, para las cuales se prueba en este documento que son funciones núcleos (k_G, k_S y k_R), pueden ser calculadas en tiempo $\mathcal{O}(n)$.

Algoritmo 2: Función de consenso utilizando Recocido Simulado

Notaciones: P −Estado actual.
e −Energía del estado actual.
P_{next} Próximo estado.
e_{next} −Energía del próximo estado.
$\hat{P}$ −Mejor solución actual.
$e_{\hat{P}}$ −Energía de la mejor solución actual.
r −Número de iteraciones.
$rMax$ −Máximo número de iteraciones.
$eMax$ −Umbral de energía.
$neighbor\,(P)$ −Devuelve un vecino de P

Entrada:
X -Conjunto de objetos.
$\mathbb{P}$ –Conjunto de particiones.
Ω -Conjunto de pesos asociados a las particiones.

Salida:
$\hat{P}$ -Partición de consenso obtenida.

$P_0 = \ arg\min_{P\in\mathbb{P}} \left\| \tilde{\phi}\,(P) - \tilde{\phi}\,(P^*) \right\|^2_{\mathcal{H}}$; //Se calcula el estado inicial.

$P = P_0;\ e = E\,(P) = \left\| \tilde{\phi}\,(P) - \tilde{\phi}\,(P^*) \right\|^2_{\mathcal{H}}$; //Inicializaciones.

$\hat{P} = P;\ e_{\hat{P}} = e;\ r = 0;$

while $r < \ rMax\ y\ e > eMax$ **do**
  $P_{next} = neighbor\,(P)$; //Seleccionar un vecino.
  $e_{next} = E\,(P_{next})$; //Calcular la energía.
  if $e_{next} < e_{\hat{P}}$ **then**
    $\hat{P} = P_{next};\ e_{\hat{P}} = e_{next}$; //Actualizar mejor solución.
  if $\exp\left(-\frac{e_{next}-e_{\hat{P}}}{\mathcal{T}}\right) > random$ **then**
    $P = P_{next};\ e = e_{next}$; //Cambiar estado actual.
  $\mathcal{T} = \ \beta\mathcal{T}$ //Actualizar la temperatura.
  $r = r + 1;$
  return $\hat{P}$; //Se devuelve la mejor partición.

en este proceso, la complejidad final puede ser estimada en $\mathcal{O}(n \cdot m \cdot rMax)$.

A pesar de que este algoritmo trata de encontrar el óptimo global de un problema de optimización combinatorio exponencial, y que muchos algoritmos de combinación de agrupamientos no enfrentan directamente este problema; la complejidad computacional de este algoritmo es comparable con la de los algoritmos de combinación de agrupamientos más eficientes, los cuales tienen una complejidad de $\mathcal{O}(n \cdot m \cdot d^*)$, donde

n es el número de objetos, m el número de particiones y d^* el número de grupos en la partición de consenso.

El conjunto de todas las particiones de X, $\mathbb{P}_X$ es finito, por tanto se podría utilizar un número de iteraciones suficientemente grande con el cual se asegure la convergencia al óptimo global del problema (2.1). Por supuesto, debido a que incluso para problemas donde X es pequeño la cardinalidad de $\mathbb{P}_X$ es un número considerablemente grande[14], esto conllevaría a un algoritmo computacionalmente intratable. Sin embargo, con un número *razonablemente* pequeño de iteraciones, es posible encontrar en la práctica resultados muy cercanos al consenso teórico. De hecho, en el método propuesto se comienza el *recocido simulado* con la partición en $\mathbb{P}$ más cercana a ψ, la cual en muchas aplicaciones prácticas podría ser incluso una solución suficientemente buena.

2.4. Resultados experimentales

Con el objetivo de mostrar como funciona el método propuesto en la práctica, se realizaron diversos experimentos con diferentes colecciones de datos. En estos experimentos, se discute la utilidad de cada paso del algoritmo de combinación de agrupamientos y se comparan los resultados obtenidos contra los resultados de algoritmos de agrupamiento así como contra los resultados de otros algoritmos de combinación de agrupamientos.

Para la aplicación del método propuesto, primeramente, es necesario seleccionar los algoritmos de agrupamiento (y sus configuraciones de parámetros) que se van a usar para generar el conjunto de particiones. Posteriormente, con el objetivo de aplicar el paso de Análisis de la Importancia de las Particiones, es necesario definir un conjunto de Índices de Validación de Propiedades (IVP). Además debe ser identificada la situación más apropiada para el problema en cuestión, es decir, Agrupamiento Sin Información (ASI) o Agrupamiento Con Información (ACI) y aplicar su correspondiente mecanismo de asignación de pesos (ecuación (2.2) o (2.3)). Es importante tener en cuenta que tanto los algoritmos de agrupamiento simples como los IVPs usan los objetos originales del problema, por tanto, deben ser seleccionados cuidadosamente teniendo en cuenta el tipo de datos del problema. Por ejemplo, en estos experimentos solo serán usados datos numéricos, por tanto se usa el algoritmos de agrupamiento

[14] La cardinalidad de $\mathbb{P}_X$ viene dada por el número de Bell, el cual responde a la siguiente fórmula recurrente $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$

k-Means y la distancia Euclidiana como distancia d_I en la definición de los índices (ver ecuación (2.12)).

Después de la aplicación del paso de Análisis de Relevancia de las Particiones, se tiene un conjunto de particiones $\mathbb{P}$ y un conjunto de pesos Ω. Con esta información se aplica el paso de combinación. En este último paso no se utiliza el conjunto de objetos originales. Luego, el método propuesto puede ser usado sin importar el tipo de datos originales del problema.

Además, en los experimentos presentados en este capítulo se utilizará la medida de similitud entre particiones basada en conteo de subconjuntos k_S (Sección 2.2.2), debido a que con esta se obtuvieron los mejores resultados experimentales.

En el paso de consenso, es necesario configurar algunos parámetros específicos del *recocido simulado*. En la Sección 2.3, se explicó cómo definir los parámetros genéricos de esta meta-heurística con el objetivo de adaptarla al problema de combinación de agrupamientos. Sin embargo, existen algunos parámetros específicos que deben ser configurados para un problema particular, tales como: temperatura inicial $\mathcal{T}_0$, constante de decrecimiento de temperatura β y los criterios de parada del algoritmo, por ejemplo, número máximo de iteraciones $rMax$ y un umbral de máxima energía $eMax$. En los experimentos de este capítulo, se utiliza la siguiente configuración $\mathcal{T}_0 = 100$, $\beta = 0.98$, $rMax = 10000$ y $eMax = 0$. Estos parámetros fueron ajustados empíricamente corriendo el algoritmo con diferentes configuraciones. Se utiliza un alto valor de β para obtener un proceso de enfriamiento lento, debido a que se corre solo una iteración por cada valor de temperatura. Se hace $eMax = 0$ con el objetivo de sólo usar el máximo número de iteraciones $rMax$ como criterio de parada. De esta manera, el algoritmo se ejecuta hasta alcanzar el máximo número de iteraciones. El algoritmo fue ejecutado en una PC Intel Core 2 Quad a 2.50 GHz. El tiempo de ejecución fue cercano a 1 minuto para los casos peores en las diferentes colecciones de datos utilizadas (compuestas por 1000 o menos objetos).

En los experimentos, se nombrará al algoritmo propuesto por *Combinación de Particiones Pesadas basado en Núcleos* (Weighted Partition Consensus via Kernels) (WPCK).

2.4.1. Descripción de las colecciones de datos

En los experimentos, se utilizaron 6 colecciones de datos. Una de estas es una colección de datos sintética compuesta por puntos en el plano bidimensional, la cual es utilizada para mostrar el comportamiento del método de manera visual. Las otras

Tabla 2.1: Descripción de las colecciones de datos

Nombre	#Objetos	#Atributos	#Clases	Objetos por clases
Ionosphere	351	34	2	126-225
Wine	178	13	3	59-71-48
Cassini	1000	2	3	400-300-300
Iris	150	4	3	50-50-50
Breast Cancer	699	9	2	458-241
Optical Digits	100	64	10	10-11-11-11-12-5-8-12-9-11

5 colecciones de datos son de la UCI Machine Learning Repository (ver Tabla 2.1) [36]. Para todas estas colecciones de datos se tiene disponible una estructuración de referencia (ground-truth) y los resultados son evaluados de acuerdo a su parecido con la asignación de etiquetas en la estructuración de referencia. Como es conocido, en muchos problemas de clasificación no supervisada no se conoce la estructuración de referencia (incluso esta no tiene que ser única) y la selección de un índice apropiado para un problema en particular es una tarea compleja. Por tanto, para mostrar la efectividad del método propuesto, se utilizan estas colecciones de datos de referencia internacional, en las cuales la estructuración de referencia está disponible. Los resultados son evaluados a partir del índice Tasa de Error de Agrupamiento (Clustering Error Rate; CER), el cual es obtenido para una partición P contando el número de objetos *mal clasificados* respecto a la estructuración de referencia disponible para cada colección de datos.

2.4.2. Índices de validación de propiedades utilizadas en los experimentos

Con el objetivo de aplicar el paso de Análisis de la Importancia de las Particiones (Sección 2.1) es necesario definir un conjunto de Índices de Validación de Propiedades (IVP). En estos experimentos, se usan cuatro índices de validación sencillos [49], donde cada uno mide el cumplimiento de una propiedad específica. Dado un conjunto de objetos $X = \{x_1, \ldots, x_n\}$, una partición $P = \{C_1, C_2, \ldots, C_t\}$ y una distancia $d_I : X \times X \to \mathbb{R}_+$, se define la *Varianza* de la partición de la siguiente manera:

$$\mathcal{VI}(P) = \frac{1}{\sqrt{\frac{1}{n}\sum_{C_i \in P}\sum_{x_j \in C_i} d_I(x_j,\ \eta_i)}} \tag{2.12}$$

donde η_i es el centroide del grupo C_i. La varianza $\mathcal{VI}$ es una manera de medir la compacidad de los grupos en la partición.

El segundo índice es la *Conectividad* que viene definido por:

$$\mathcal{CI}(P) = \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{nc} v_{i,j}}$$

donde $v_{i,j} = \begin{cases} \frac{1}{j}, & si\ \nexists C_h :\ x_i \in C_h \ \text{ y } nn(i,j) \in C_h \\ 0, & \text{en otro caso} \end{cases}$, y $nn(i,j)$ es el j-ésimo vecino más cercano del objeto x_i. nc es el número de vecinos que contribuyen a la medida de conectividad $v_{i,j}$. En estos experimentos se usa $nc = 5$. Este índice evalúa el grado de conectividad de los grupos en una partición midiendo cuántos vecinos de un objeto pertenecen al mismo grupo que el objeto.

El tercer índice usado en estos experimentos es el *Ancho de la Silueta*, definido a partir de la siguiente expresión:

$$\mathcal{SI}(P) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{SI}'(x_i)$$

donde $\mathcal{SI}'(x_i) = \frac{b_i - a_i}{\max(b_i,\ a_i)}$ y a_i denota la distancia promedio entre el objeto x_i y el resto de los objetos en su grupo, b_i denota la distancia promedio entre x_i y los objetos en el grupo más cercano. Este índice da una medida de la compacidad y separación de cada grupo dentro de la partición.

Además, se utiliza el *Índice de Dunn*:

$$\mathcal{DI}(P) = \min_{C_i \in P}\left\{\min_{C_j \in P} \frac{dist(C_i,\ C_j)}{\max_{C_h \in P} diam(C_h)}\right\}$$

donde $diam(C_h)$ es la máxima distancia intra-grupo en el grupo C_h y $dist(C_i,\ C_j)$ es la mínima distancia entre pares de objetos x_a y x_b tal que $x_a \in C_i$ y $x_b \in C_j$. Este índice es una medida de la proporción entre la menor distancia inter-grupo y la mayor distancia intra-grupo, por tanto es otra medida de la relación entre compacidad y separación entre los grupos en la partición.

En estos cuatro índices, mayores valores denotan mayor cumplimiento con la propiedad representada por cada índice. Por tanto, son consistentes con la definición de Índice de Validación de Propiedades (IVP) dada en la Sección 2.1.

2.4.3. Experimentación y análisis

Para mostrar el comportamiento del método propuesto desde diferentes puntos de vista, se realizan en esta sección cuatro tipos de experimentos. En el primero, se discute en detalle el funcionamiento del algoritmo sobre las colecciones de datos Ionosphere y Wine, mostrando la importancia de cada uno de los pasos del mismo y las diferencias existentes entre los dos tipos de mecanismos de asignación de pesos. En el segundo experimento, se presenta un análisis visual del desempeño del algoritmo utilizando la colección de datos Cassini. En el tercer experimento, el método propuesto es comparado contra el desempeño promedio de los algoritmos de agrupamiento, en particular, el algoritmo k-Means. Finalmente, en el cuarto experimento, se compara el algoritmo propuesto con otros algoritmos de combinación de agrupamientos reportados en la literatura. En todos los experimentos se utilizan los cuatro índices (Varianza ($\mathcal{VI}$), Conectividad ($\mathcal{CI}$), Ancho de la Silueta ($\mathcal{SI}$) y el Índice de Dunn ($\mathcal{DI}$)). También se muestra el valor d_ψ, el cual para una partición P, representa la proximidad al consenso teórico ψ (calculada a través de la ecuación 2.10).

Influencia del mecanismo de asignación de pesos

En este experimento se usan 10 particiones de la colección de datos Ionosphere para ilustrar la influencia del mecanismo de asignación de pesos en el cálculo del valor d_ψ. Las 10 particiones se obtuvieron mediante la aplicación del algoritmo de agrupamiento k-Means con diferentes inicializaciones de los parámetros y usando subconjuntos aleatorios de atributos para representar los objetos en cada corrida del algoritmo k-Means. En la Tabla 2.2, se muestran la aplicación de los cuatro índices a todas las particiones, los porcientos de CER para cada partición y los valores de d_ψ para cuatro configuraciones diferentes.

En la primera configuración, no son usados los pesos en la definición de la función de consenso, o dicho de otra manera, se le asigna el valor de peso 1 a cada partición asumiendo que todas tienen la misma importancia para el proceso de combinación. En la segunda configuración, los pesos se calculan asumiendo la situación ASI, es decir, no se tiene conocimiento acerca de los índices que se deben maximizar y los índices son utilizados para descubrir información. La tercera configuración es asumiendo la situación ACI, es decir, asumiendo que los cuatro índices definidos previamente son los que se deben maximizar. Finalmente, la cuarta configuración es también asumiendo la situación ACI pero solamente teniendo en cuenta Varianza ($\mathcal{VI}$) y el Índice de Dunn ($\mathcal{DI}$). En la Tabla 2.2, la columna $d_\psi(i)$ se refiere a la $i-$ésima configuración y

$\Omega(i)$ denota el peso asociado a esta configuración. No se muestra $\Omega(1)$ ya que este es igual a 1 para todas las particiones.

Tabla 2.2: Experimentos con 10 particiones de la colección de datos Ionosphere. Se muestran los diferentes tipos de mecanismos de asignación de pesos.

$\mathbb{P}$	$\mathcal{VI}$	$\mathcal{CI}$	$\mathcal{SI}$	$\mathcal{DI}$	$CER(\%)$	$d_\psi(1)$	$\Omega(2)$	$d_\psi(2)$	$\Omega(3)$	$d_\psi(3)$	$\Omega(4)$	$d_\psi(4)$
P_{00}	0,78	0,20	0,79	0,18	25,07	0,925	0,826	0,857	0,499	1,032	0,458	1,038
P_{01}	1	1	1	0,81	28,77	0,767	0,205	1,042	1	**0,530**	0,906	0,682
P_{02}	0,99	0,99	1	0,81	29,05	**0,760**	0,212	1,037	0,997	0,544	0,905	0,668
P_{03}	0,99	0,76	0,99	1	30,76	0,883	0,255	1,074	0,982	0,770	1	**0,654**
P_{04}	0	0,09	0,20	0,09	45,01	0,925	0,273	1,084	0,075	1,193	0	1,216
P_{05}	0,13	0,43	0,44	0,93	17,66	0,855	0,999	**0,645**	0,497	0,969	0,515	0,944
P_{06}	0,14	0,37	0,44	0,93	17,37	0,855	0,988	0,648	0,481	0,973	0,521	0,943
P_{07}	0,02	0	0	0,1	29,62	0,925	0	1,196	0	1,222	0,012	1,211
P_{08}	0,17	0,01	0,06	0	46,15	0,925	0,119	1,147	0,034	1,208	0,039	1,201
P_{09}	0,84	0,27	0,84	0,72	27,92	0,925	1	0,786	0,697	0,957	0,779	0,913

Analizando la Tabla 2.2, se observa que:

- La manera de calcular los pesos afecta el resultado final. Cada configuración produjo como resultado una partición diferente con menor valor de d_ψ asociado. Por lo tanto, es muy importante la identificación de la situación correcta, agrupamiento sin información (ASI) o con información (ACI) y la selección del conjunto de índices.

- Comparando las configuraciones 1 y 2, se puede constatar que debido al cálculo de los pesos, la partición con un valor menor de d_ψ fue una partición más cercana a la estructuración de referencia (con un menor valor CER). El análisis hecho con los índices y los pesos asignados a las particiones permite definir una partición de consenso teórica más cercana a la estructuración de referencia.

- Analizando la configuración 3, nos damos cuenta de que cuando se asignan los pesos maximizando los valores de los índices (situación ACI) la partición con menor d_ψ no es la más cercana a la estructuración de referencia. Esto corrobora que a diferencia de como puede pensarse, la partición más compacta y separada no es siempre la solución más apropiada para todo problema. En ocasiones, la *mejor* partición para un problema de agrupamiento, es aquella que satisface estas y otras propiedades con cierto grado de cumplimiento. Descubrir estas propiedades y su grado de cumplimiento óptimo es precisamente lo que se persigue con el proceso de asignación de pesos.

- Observando la configuración 4, se puede ver que cuando se usa un subconjunto de índices la partición con menor d_ψ cambia a la partición que maximizó los

valores de los índices analizados. De esta manera, en un problema de ACI, se pueden cambiar los índices en el proceso de asignación de peso para mover el consenso en la dirección de los resultados esperados.

En los siguientes experimentos, se usa el mecanismo de asignación de peso de la situación ASI (ecuación (2.2)) debido a que realmente no se conocen las propiedades que se deben maximizar para acercarse a la estructuración de referencia en las colecciones de datos utilizadas.

Tabla 2.3: Veinte particiones de la colección de datos Wine y la partición de consenso P^*. Se muestran los resultados de la aplicación de los índices, los pesos asociados a cada partición, el porciento de CER y la posición según el valor de d_ψ de cada partición.

$\mathbb{P}$	$\mathcal{VI}$	$\mathcal{CI}$	$\mathcal{SI}$	$\mathcal{DI}$	Ω	*CER (%)*	d_ψ
P_{00}	0,39	0,53	0,42	0,16	0,97	2,80	6th
P_{01}	0,34	0,57	0,47	0,16	0,95	5,16	2nd
P_{02}	0,36	0,56	0,46	0,14	0,95	3,37	3rd
P_{03}	0,42	0,61	0,44	0,08	0,97	5,02	11th
P_{04}	0,38	0,60	0,38	0,16	0,99	7,86	12th
P_{05}	0,77	0,78	0,27	0	0,68	15,73	17th
P_{06}	0,37	0,60	0,42	0,17	1	10,11	15th
P_{07}	0,40	0,67	0,38	0,04	0,92	13,48	16th
P_{08}	0,85	1	0,12	0,01	0,49	30,89	18th
P_{09}	0,38	0,55	0,45	0,19	0,95	3,37	9th
$\mathbf{P_{10}}$	**0,37**	**0,55**	**0,43**	**0,16**	**0,97**	**3,93**	**1st**
P_{11}	0,34	0,54	0,48	0,16	0,93	4,49	4th
P_{12}	0,38	0,59	0,43	0,10	0,97	6,74	8th
P_{13}	0	0	1	1	0	29,77	20th
P_{14}	1	0,98	0	0,06	0,41	38,76	19th
P_{15}	0,35	0,53	0,45	0,16	0,94	6,74	10th
P_{16}	0,37	0,53	0,43	0,16	0,96	2,80	5th
P_{17}	0,47	0,63	0,40	0,08	0,97	4,49	13th
P_{18}	0,36	0,58	0,44	0,16	0,97	6,17	7th
P_{19}	0,52	0,66	0,38	0,08	0,93	6,17	14th
Prom.	**0,44**	**0,60**	**0,45**	**0,16**	**0,84**	**10,39**	-
P^*	**0,39**	**0,55**	**0,44**	**0,16**	-	**3,88**	-

En los experimentos cuyos resultados se muestran en la Tabla 2.3, se usó la colección de datos Wine y se crearon 20 particiones de los datos $(P_{00}, P_{01}, \ldots, P_{19})$ mediante la aplicación del algoritmo k-Means con diferentes parámetros. En dicha tabla se presentan los valores de cada índice, los pesos asignados, el valor CER y la posición de cada partición organizada por su valor de d_ψ. Además, se muestra el comportamiento promedio de todas las particiones y la partición de consenso obtenida P^*. Todos los valores de los índices y pesos fueron normalizados en el intervalo $[0, 1]$ para resaltar las diferencias entre particiones.

Al analizar estos experimentos se puede observar que:

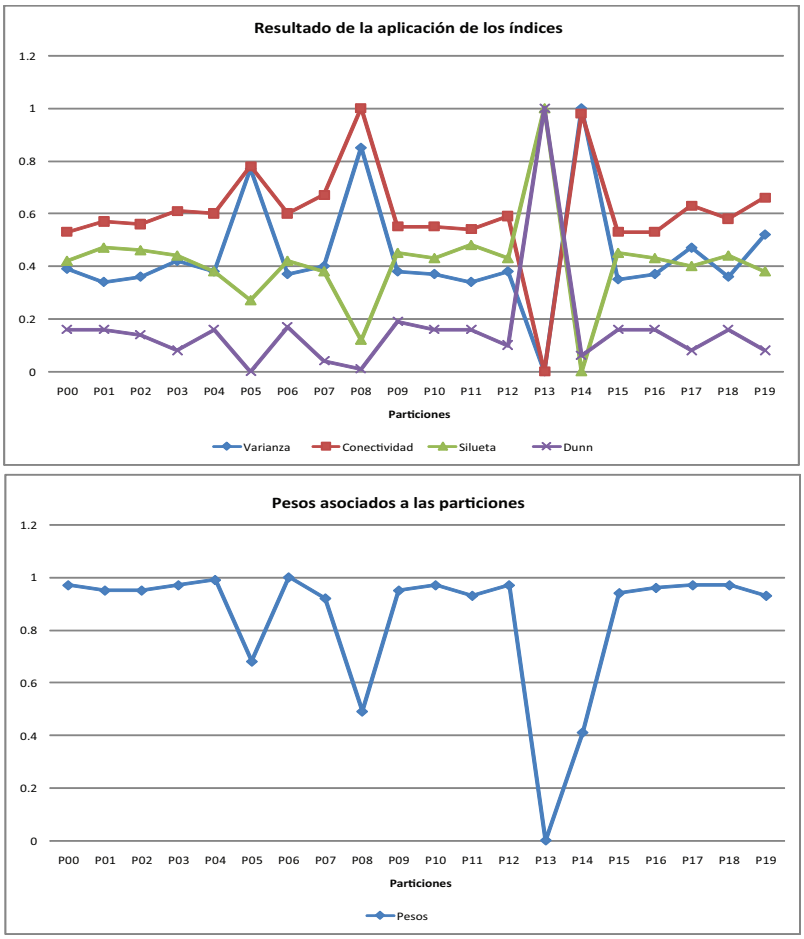


Figura 2.1: Arriba: Curvas de los índices evaluados en todas las particiones. Abajo: Los pesos asociados a cada partición.

- La partición de consenso tiene un valor CER más pequeño que el valor CER promedio entre todas las particiones.

- Sin embargo, la partición de consenso obtenida P^* no tiene el menor valor CER. Esto es esperado debido a que el algoritmo de combinación de agrupamientos no cuenta con la información de la estructuración de referencia. Ordenando las particiones por su valor CER, la partición de consenso toma el cuarto lugar de 21, con un valor muy cercano al menor valor de CER, lo cual hace a la partición de consenso una elección adecuada.

- La partición de consenso tiene un comportamiento muy similar al comportamiento promedio de las particiones para cada índice.

- La partición más cercana al consenso teórico (P_{10} en este caso) no es la que

tiene un mayor peso asociado, sin embargo sí tiene un alto valor de peso.

- Se asigna un peso pequeño a particiones bien diferentes al resto como son P_8, P_{13} y P_{14} (que a su vez tienen un alto valor de CER). Esto ocurre debido a que estas particiones son las que presentan un comportamiento más irregular con respecto al valor promedio para todos los índices. En la Figura 2.1, se puede observar que picos pronunciados en las curvas de los índices implican valores pequeños de peso y que particiones con valores de índices cercanos al promedio tienen un alto peso asociado.

- Las particiones P_8, P_{13} y P_{14}, las cuales tienen un alto valor de CER, son las más alejadas al consenso teórico, es decir, estas particiones están en las últimas tres posiciones respecto al valor d_ψ.

Análisis visual

En este experimento se utiliza la colección de datos Cassini. Esta consiste en 3 estructuras: dos externas con forma de *banana* y una con forma de círculo en el centro (ver Figura 2.2). Utilizando el algoritmo k-Means, se generaron 5 particiones de los datos $(P_0, P_1, P_2, P_3, P_4)$ (ver Tabla 2.4). Esta colección de datos está constituida por puntos en el plano, donde cada objeto viene representado por dos características: la coordenada x y la y. Las primeras 4 particiones se obtuvieron teniendo en cuenta solamente la información de la coordenada y, y la última fue generada utilizando la información de ambas coordenadas. La idea de utilizar solamente la coordenada y se debe a que utilizando esta coordenada, los objetos son fácilmente separables según la estructura de la partición de referencia (ver Fig. 2.2). Sin embargo, usando ambas coordenadas, la forma de *banana* de las dos clases externas son bien difíciles de obtener utilizando el algoritmo k-Means.

Tabla 2.4: Experimentos con la colección de datos Cassini

$\mathbb{P}$	Ω	*CER*	*CER (%)*	d_ψ
P_0	**0,98**	**16**	**1,6**	**0,27**
P_1	0,97	5	0,5	0,46
P_2	0,85	24	2,4	0,57
P_3	1	17	1,7	0,28
P_4	0	343	34,3	1
P^*	-	14	1,4	0,19

En este experimento, como en el anterior, la partición más cercana al consenso teórico (P_0) no es la que tiene un menor valor de CER, sin embargo tiene un valor de

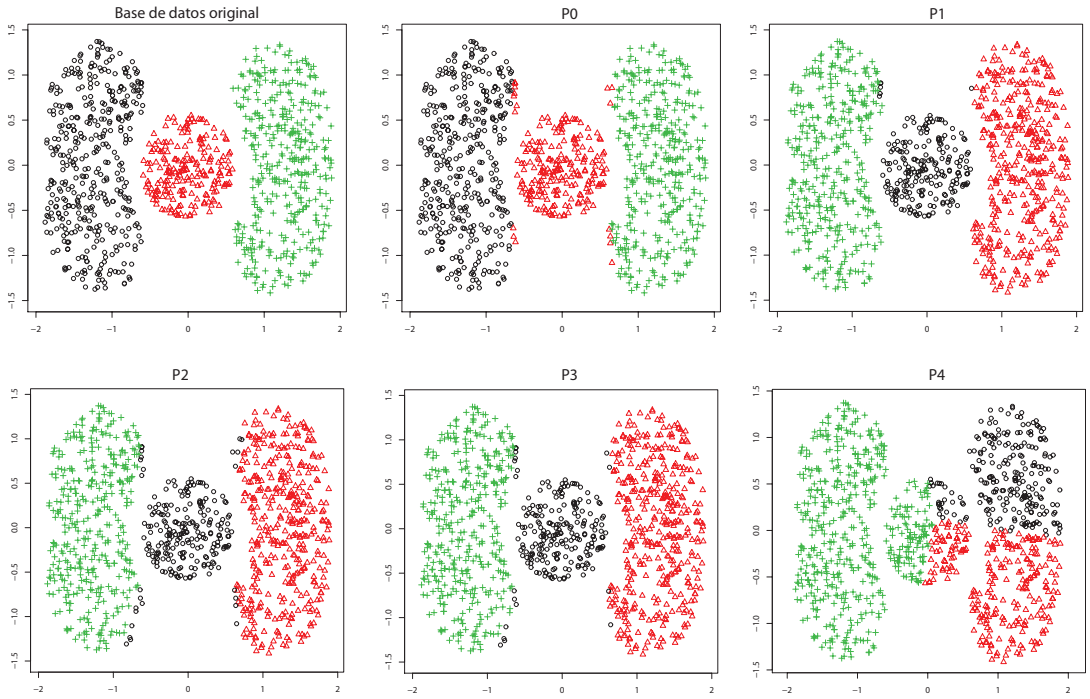


Figura 2.2: Estructuración de referencia de la colección de datos Cassini y 5 particiones obtenidas utilizando el algoritmo k-Means. Las primeras cuatro (P_0 - P_3) se obtuvieron utilizando solamente la coordenada y de cada punto y la última (P_4) utilizando ambas coordenadas, x y y.

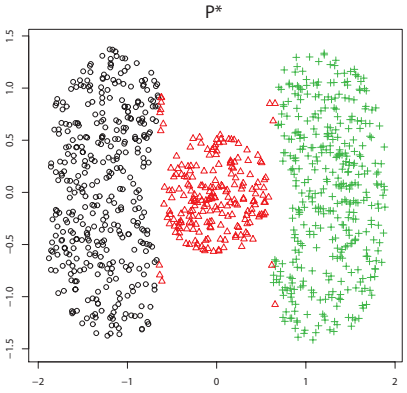


Figura 2.3: La partición de consenso obtenida.

CER muy pequeño en comparación con la peor partición (P_4). Además, la partición más diferente al resto tiene el menor peso asociado y es la que más lejos se encuentra del consenso teórico. La partición de consenso obtenida P^* (ver Figura 2.3) es la más cercana al consenso teórico y solo se diferencia de la estructuración de referencia en 14 puntos de los 1000 que componen la colección de datos.

Comparación con algoritmos de agrupamiento simples

En esta sección, se compara el algoritmo propuesto (WPCK) con el desempeño promedio del algoritmo de agrupamiento simple k-Means utilizando las colecciones de datos Ionosphere e Iris (ver Tabla 2.5). Los experimentos fueron realizados usando diferentes números de particiones (m) para la combinación. Las columnas *k-Means(min)* y *k-Means(max)* muestran el menor y mayor valor de CER entre todas las particiones. Los resultados del método propuesto fueron superiores al desempeño promedio del algoritmo k-Means, presentado en la columna *k-Means(prom)*, para todos los casos (ver Tabla 2.5). Por otra parte, son cercanos a los resultados del k-Means con menor valor de CER y lejanos de los resultados de k-Means con mayores valores de CER.

Tabla 2.5: Porciento (%) de CER del algoritmo k-Means y del algoritmo de combinación de agrupamientos propuesto (WPCK).

Colección de datos	m	k-Means(prom)	k-Means(min)	k-Means(max)	WPCK
Ionosphere	10	28,5	17,3	45,1	**17,6**
Ionosphere	20	26,3	16,2	45,1	**16,5**
Iris	10	22,3	9,3	49,3	**11,3**
Iris	20	18,6	4,0	49,3	**7,3**
Iris	30	15,1	4,0	49,3	**5,3**

Comparación con otros métodos de combinación de agrupamientos

En este último experimento, el método propuesto se compara contra otros algoritmos de combinación de agrupamientos. Primeramente, en la Tabla 2.6 se muestra una comparación del método propuesto contra métodos basados en particionamiento de (hiper)grafos CSPA, HGPA, MCLA [99] y Teoría de la Información QMI [101] sobre la colección de datos Iris, usando diferentes números de particiones m en la combinación.

Tabla 2.6: CER (%) para la colección de datos *Iris*

m	CSPA	HGPA	MCLA	QMI	WPCK
5	11,2	41,4	**10,9**	14,7	11,3
10	11,3	38,2	**10,9**	10,8	11,3
20	9,8	39,1	10,9	10,9	**7,3**
30	7,9	43,4	11,3	10,9	**5,3**
40	7,7	41,9	11,1	12,4	**5,3**

Finalmente, en la Tabla 2.7 se muestra la comparación del método propuesto contra el método basado en co-asociación EA [37] y los métodos basados en (hiper)grafos usando las colecciones de datos Breast Cancer y Optical Digits. EA-SL se refiere al

Tabla 2.7: CER(%) para las colecciónes de datos *Breast Cancer* y *Optical Digits*

Colección de datos	**EA-SL**	**EA-AL**	**CSPA**	**HPGA**	**MCLA**	**WPCK**
Breast Cancer	4,0	4,0	17,3	49,9	3,8	**3,6**
Optical Digits	56,6	23,2	**18,1**	40,7	18,5	22,1

método EA utilizando el algoritmo Single-Link para obtener la partición de consenso y el EA-AL usa el algoritmo Average-Link.

Los resultados obtenidos muestran que el método propuesto es otra alternativa para enfrentar problemas de clasificación no supervisada y que pueden obtenerse resultados mejores a los obtenidos por otros métodos.

Capítulo 3

COMBINACIÓN DE AGRUPAMIENTOS HETEROGÉNEOS

En este capítulo, se muestra cómo el uso de los objetos originales del problema pueden mejorar la calidad del proceso de combinación. Sin embargo, el uso del conjunto original de objetos después del proceso de generación de particiones no siempre es una tarea fácil. Por ejemplo, en el paso de generación pueden utilizarse diferentes representaciones de los objetos o la misma representación pero diferentes medidas de (di)similitud entre objetos en el momento de generar las particiones. Por tanto, si se desea utilizar los objetos originales después del paso de generación, se deben responder las siguientes interrogantes: ¿Qué representación de los objetos debe ser utilizada? ¿Qué medida de (di)similitud debe ser empleada?

Una posible respuesta a estas interrogantes se puede encontrar a partir de la definición de un procedimiento en el cual inmediatamente después del paso de generación se resuma la información en el conjunto de particiones a combinar. Una idea similar se encuentra en los métodos basados en co-asociación [37] (ver Sección 1.2.3). Estos métodos tratan de unificar la información de las particiones a combinar en la matriz de co-asociación, la cual es vista como una nueva matriz de similitud entre objetos. En este sentido, la matriz de co-asociación parece ser la herramienta ideal para acumular toda la información en el conjunto de particiones a combinar. Sin embargo, como se verá en más detalle en la Sección 3.1, existe información valiosa que no puede ser extraída por la matriz de co-asociación.

Por tanto, se propone en la Sección 3.1 la *matriz de asociación pesada*, la cual es

más expresiva como medida de similitud. Esta nueva matriz de similitud es el punto de partida de los dos métodos de combinación de agrupamientos que se proponen en este capítulo. En la Sección 3.2 se presenta un método de combinación de agrupamientos basado en la idea de acumulación de evidencia utilizando la matriz de asociación pesada. En la Sección 3.3 se utiliza la matriz de asociación pesada para obtener una nueva representación de los objetos originales, la cual se puede utilizar en el proceso de combinación sin importar el mecanismo de generación que haya sido aplicado. Además, se presenta un nuevo algoritmo de combinación de agrupamientos, que es una generalización del algoritmo presentado en el capítulo anterior. Finalmente, se presentan y discuten en la Sección 3.4 algunos resultados experimentales.

3.1. Matriz de asociación pesada

La matriz de co-asociación CA se definió en [37] como una nueva medida de similitud entre objetos basada en la información del conjunto de particiones a combinar. Cada celda de esta matriz tiene el siguiente valor:

$$CA(i,j) = \sum_{t=1}^{m} \delta_{ij}(P_t), \qquad \text{donde} \quad \delta_{ij}(P) = \begin{cases} 1, & \exists C \in P \ (x_i \in C \wedge x_j \in C) \\ 0, & \text{en otro caso} \end{cases} \tag{3.1}$$

De esta forma, el valor en cada posición (i,j) de la matriz es una medida de cuántas veces los objetos x_i y x_j están en el mismo grupo para todas las particiones en $\mathbb{P}$.

Siguiendo una idea similar se introduce la matriz de *p-asociación* PA [120] como una extensión de la matriz de co-asociación antes presentada. Esta matriz se define como:

$$PA(i,j) = \sum_{t=1}^{m} \gamma_{ij}(P_t), \qquad \text{donde} \quad \gamma_{ij}(P) = \begin{cases} 1, & i = j \\ \frac{1}{1+\sqrt[f]{|C|}}, & \exists C \in P \ (x_i \in C \wedge x_j \in C) \\ 0, & \text{en otro caso} \end{cases} \tag{3.2}$$

En esta nueva matriz los valores tienen una naturaleza continua en vez de binaria como en la matriz de co-asociación. Además, se tiene en cuenta para su definición, el tamaño de los grupos y la dimensión f de la tupla de características que representa a cada objeto.

Sin embargo, aún existe información valiosa en el conjunto de particiones que no se

tiene en cuenta en ninguna de las matrices definidas anteriormente. Por este motivo, se define la *matriz de asociación pesada*, la cual como se mostrará posteriormente, extrae más información del conjunto de particiones $\mathbb{P}$ con el objetivo de obtener una mejor matriz de similitud entre objetos. Primeramente, se afirma en este trabajo que el hecho de que dos objetos pertenezcan al mismo grupo en una partición no brinda la misma información para todas las particiones. Por este motivo se definirá el *valor de asociación* entre dos objetos x_i y x_j que pertenecen a un mismo grupo en cierta partición $P \in \mathbb{P}_X$. Para calcular este valor de asociación, se tendrán en cuenta tres factores: el número de elementos en el grupo al cual x_i y x_j pertenecen, el número de grupos en la partición analizada y el valor de (di)similitud entre estos dos objetos usando la misma medida que se utilizó para generar la partición P.

Ahora se muestra un ejemplo ilustrativo:

Ejemplo 3.1. *Sea $\{x_1, \ldots, x_{10}\}$ un conjunto de diez objetos y P_1, P_2, P_3, P_4 cuatro particiones de estos objetos (ver Figura 3.1). En el primer caso (a), se puede decir que el hecho de que x_1 y x_2 pertenezcan al mismo grupo brinda más información en el caso de la partición P_1 que en P_2. Esto es debido a que en P_1 los datos están más segmentados, es decir, existe un mayor número de grupos, por lo tanto se puede asumir que el algoritmo de agrupamiento utilizado para obtener P_1 fue más* discriminativo. *Sin embargo, a pesar de esto, ambos objetos se mantuvieron juntos en el mismo grupo. En el segundo caso (b), se puede decir que el hecho de que x_1 y x_2 estén agrupados juntos contribuye con más información en el caso de la partición P_4. El número de elementos en el grupo al que ambos objetos pertenecen es más pequeño en P_4 que en P_3. Por tanto, en este caso, el algoritmo aplicado para obtener P_4 fue más* discriminativo *en una vecindad de este par de objetos. Aún así, ambos permanecieron juntos en el mismo grupo.*

Del Ejemplo 3.1, se puede decir intuitivamente que dos objetos x_i y x_j, clasificados en el mismo grupo C para cierta partición P, la cual se obtiene utilizando una medida de similitud Γ_P, tienen un alto *valor de asociación* si se satisfacen las siguientes condiciones:

1. $|C|$ es pequeño($|C|$ es el número de elementos en el grupo C)

2. $|P|$ es grande ($|P|$ es el número de grupos en P)

3. $\Gamma_P(x_i, x_j)$ es grande.

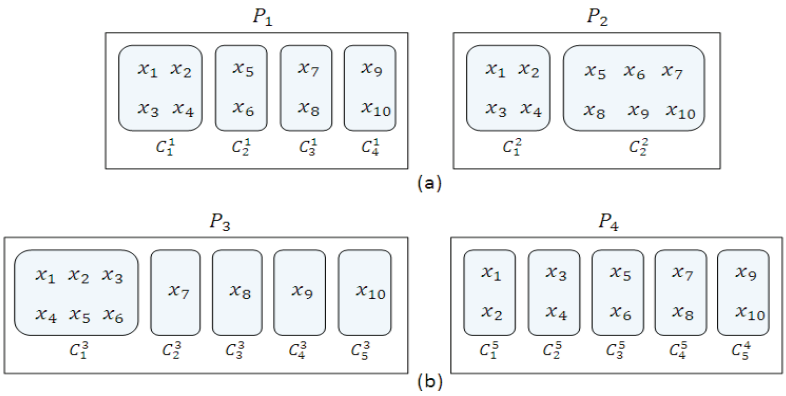


Figura 3.1: Las cuatro particiones de diez objetos en el Ejemplo 3.1

Si la partición P se obtuvo utilizando una medida de disimilitud d_P, se puede fácilmente obtener una medida de similitud equivalente utilizando el inverso o el opuesto de esta medida. Por tanto, a partir de ahora se asume que el algoritmo de agrupamiento utilizado para obtener cada partición utilizó una medida de similitud Γ_P.

Siguiendo la idea anterior, se define la matriz de asociación pesada WA como:

$$WA(i,j) = \sum_{t=1}^{m} \lambda_{ij}(P_t), \text{ donde } \lambda_{ij}(P) = \begin{cases} \frac{|P|}{|C|} \cdot \Gamma_P(x_i, x_j), & \exists C \in P \ (x_i \in C \wedge x_j \in C) \\ 0, & \text{en otro caso} \end{cases} \tag{3.3}$$

Otro ejemplo para mostrar el beneficio de la matriz WA es el siguiente:

Ejemplo 3.2. *Como se vio en la Definición 1.1, se puede definir la relación de orden parcial* "anidado en" *denotada por* $\preceq$ *sobre el conjunto de particiones* $\mathbb{P}_X$. *Los algoritmos de agrupamiento jerárquicos, como el Single-Link o Average-Link, producen jerarquías de particiones anidadas donde, si* $P \preceq P'$ *con* $P \neq P'$, *significa que el criterio utilizado para obtener* P *fue más* discriminativo *que el usado para obtener* P', *es decir,* P *se encuentra en un nivel más bajo en la jerarquía que* P'. *De esta forma, el hecho de que un par de objetos* x_i, x_j *pertenezca a un mismo grupo* C *en* P *brinda más información acerca del parecido de estos objetos que el hecho de que estos pertenezcan a un mismo grupo* C' *en* P'. *Usando las matrices de co-asociación* (3.1) *o p-asociación* (3.2), *no puede extraerse esta información. Sin embargo, utilizando la matriz de asociación pesada* (3.3) *se le da más valor a la partición* P *ya que si* $P \preceq P'$ *esto implica que* $|P| \geq |P'|$ *y* $|C| \leq |C'|$. *Entonces* $\frac{|P|}{|C|} \cdot \Gamma_P(x_i, x_j) \geq \frac{|P'|}{|C'|} \cdot \Gamma_{P'}(x_i, x_j)$ *ya que en este caso* $\Gamma_P = \Gamma_{P'}$.

3.2. Método de acumulación de evidencia pesada

En los métodos de Acumulación de Evidencia [37] y Acumulación de Probabilidad [120], el primer paso es el cálculo de la matriz de co-asociación o de p-asociación respectivamente. Posteriormente, la partición de consenso se obtiene mediante la aplicación de un algoritmo de agrupamiento jerárquico aglomerativo. La partición de consenso se obtiene utilizando el criterio de *mayor tiempo de vida* (highest lifetime) para cortar la jerarquía de particiones producida por el algoritmo jerárquico a un nivel determinado. El highest lifetime [37] es una simple pero efectiva heurística para determinar un nivel representativo en una jerarquía. El t-lifetime se define como el rango de valores con los cuales se puede cortar el dendrograma y obtener t grupos. Después de calcular el valor de lifetime para cada nivel en la jerarquía, se selecciona el que tiene un mayor valor[1].

Una primera aplicación de la matriz de asociación pesada WA propuesta en este capítulo es seguir con esta misma filosofía de los métodos basados en co-asociación. Para esto se define el algoritmo *Acumulación de Evidencia Pesada* (Weighted Evidence Accumulation; WEA), que consiste en: como primer paso, calcular la matriz WA y posteriormente aplicar un algoritmo de agrupamiento jerárquico seleccionando la partición de consenso utilizando el criterio highest lifetime.

3.3. Generalización del método de combinación de agrupamientos basado en funciones núcleo

En el capítulo anterior se presentó el algoritmo de combinación de agrupamientos WPCK, que es un claro ejemplo de cómo el uso de los datos originales del problema puede mejorar la calidad del proceso de combinación. Este método incorpora el paso Análisis de la Importancia de las Particiones (AIP) a la metodología usual de los métodos de combinación de agrupamientos, donde el uso de los objetos originales es crucial para la asignación de un peso a cada partición que representa su importancia para el proceso de combinación. Estos pesos asociados a cada partición permiten la realización de un mejor proceso de combinación de las particiones debido a que las particiones más relevantes son las que desempeñan el rol fundamental en este proceso. Sin embargo, el uso de los objetos originales después de la generación podría llevarnos

[1]En el Capítulo 4.1 se presenta un estudio más completo de los algoritmos existentes para la selección de un nivel representativo en una jerarquía de particiones.

a las siguientes situaciones:

- En el proceso de generación pueden ser usadas diferentes representaciones de los datos y diferentes medidas de (di)similitud entre objetos. ¿Cuál de estas debe ser seleccionada en el paso AIP?

- Para cualquier operación que se haga sobre los datos originales, la naturaleza de estos debe ser tenida en cuenta, por ejemplo, no debe ser aplicada una distancia Euclidiana sobre datos mezclados[2].

La búsqueda de una respuesta directa a estas interrogantes conllevaría a un proceso casuístico y muy dependiente de las particularidades de los datos del problema en cuestión. Además, la naturaleza no supervisada del proceso de combinación de agrupamientos dificulta la selección de una representación o medida de (di)similitud más apropiada para un problema práctico. En esta sección, se presenta una solución a estas preguntas, la cual a pesar del comportamiento no supervisado de la combinación de agrupamientos, es apropiada y puede ser aplicada de la misma manera independientemente de las características particulares de un problema en cuestión.

La idea consiste en introducir un nuevo paso llamado *Unificación de Información* inmediatamente después de la generación de las particiones. Este paso tiene como objetivo unificar en una nueva representación de los datos las diferentes representaciones iniciales de los datos y las medidas de (di)similitud que fueron utilizadas en el paso de generación. En este sentido la matriz de asociación pesada es una herramienta excelente para resumir la información del conjunto de particiones, por tanto esta va a ser un punto de partida para la obtención de la nueva representación unificada de los datos.

Una vez que se tiene la matriz WA, se puede construir a partir de esta un espacio de similitud basados en las ideas de la representación por (di)similitudes para el reconocimiento de patrones [83]. Originalmente, cada objeto $x \in X$ es representado por una tupla en cierto espacio de características $f-$dimensional $\mathbb{G}^f$. Entonces, con el objetivo de obtener el nuevo espacio de similitudes, se define una función $\theta(\cdot, X) : \mathbb{G}^f \rightarrow \mathbb{R}^n$, tal que para cada objeto x, $\theta(x, X) = (WA(x, x_1), WA(x, x_2), \ldots, WA(x, x_n))$.

De esta manera, se obtiene una nueva representación de los datos que resume la información acerca de las diferentes representaciones y medidas de proximidad utilizadas en el proceso de generación. Además, esta nueva representación como un vector de $\mathbb{R}^n$ permite el uso de herramientas matemáticas desarrolladas

[2]Datos compuestos por atributos numéricos y no numéricos

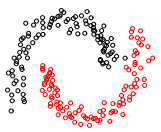
para espacios vectoriales las cuales no tienen por qué estar disponibles para las representaciones originales de los datos, por ejemplo, en caso de usar datos mezclados. Se denomina Combinación de Particiones Heterogéneas basada en Núcleos Heterogeneous Partition Consensus via Kernels (HPCK) al método que extiende el algoritmo WPCK mediante la introducción del paso *Unificación de Información*. El principal inconveniente de utilizar el paso *Unificación de Información* es que en este se calcula la similitud entre todos los pares de objetos a partir del cómputo de la matriz de asociación pesada, por tanto este paso tiene un costo computacional $\mathcal{O}(n^2)$. De esta forma, los métodos presentados en este capítulo WEA y HPCK no son apropiados en problemas con grandes volúmenes de datos.

3.4. Resultados experimentales

Para probar de manera experimental el método propuesto se utilizan siete colecciones de datos con diferentes tipos de datos (ver Tabla 4.1). Seis de estas son de la UCI Machine Learning Repository [36] y la otra es una colección de datos sintética compuesta por puntos en el plano, la cual es utilizada para probar la capacidad de los algoritmos propuestos en la determinación de estructuras complejas en los datos. Para todas estas colecciones de datos la estructuración de referencia de los datos (ground-truth) está disponible. Por lo tanto, al igual que en el capítulo anterior, en los experimentos se comparan los resultados obtenidos con la estructuración de referencia de cada colección de datos. Se utiliza el índice CER para evaluar el resultado de los algoritmos, que como se vio en el capítulo anterior, evalúa la partición resultado contando cuántos objetos son mal clasificados respecto a la estructuración de referencia. Además, se utilizan los índices F, Pureza, Rand e Información Mutua Normalizada (NMI) [3].

Tabla 3.1: Información de las colecciones de datos

Nombre	Tipo	Objetos	Atributos	Clases	Obj-por-clases	
Iris	Numérico	150	4	3	50-50-50	
Wine	Numérico	178	13	3	59-71-48	
Half-Rings	Numérico	200	2	2	100-100	
Zoo	Mezclado	101	18	7	41-20-5-13-4-8-10	
Auto	Mezclado	205	26	6	3-22-67-54-32-27	
Soybeans	Categórico	47	21	4	10-10-10-17	
Votes	Categórico	435	16	2	267-168	


Half-Rings

En estos experimentos, las particiones a combinar se obtienen mediante la aplicación del algoritmo k-Means m veces con un número aleatorio de grupos

(parámetro k del algoritmo) en el rango de 2 a 10. Además, se hizo una selección aleatoria de un conjunto de atributos de los objetos en cada corrida del algoritmo. Finalmente, se empleó una de las siguientes tres medidas de disimilitud entre pares de objetos x_i y x_j en cada corrida del k-Means.
(a) $\sqrt{\sum_{h=1}^{f} (\mathcal{CC}_{ij}(h))^2}$ (b) $\sum_{h=1}^{f} |\mathcal{CC}_{ij}(h)|$ (c) $\max_{h=1,\ldots,f} |\mathcal{CC}_{ij}(h)|$
donde $\mathcal{CC}$ es un criterio de comparación que depende del tipo de datos de cada atributo en la representación de los objetos. Este se define de la siguiente manera:

$$\mathcal{CC}_{ij}(h) = \begin{cases} \frac{x_{ih}-x_{jh}}{h_{max}}, & \text{si } h \text{ es un atributo numérico;} \\ \begin{cases} 0, & x_{ih} = x_{jh}; \\ 1, & \text{en otro caso.} \end{cases}, & \text{si } h \text{ es un atributo no numérico.} \end{cases}$$

donde x_{ih} denota el valor del h-ésimo atributo del objeto x_i y h_{max} es la máxima diferencia entre dos valores del h-ésimo atributo entre todos los objetos. Estos criterios de comparación son definidos con el objetivo de extraer el significado de la diferencia entre dos valores de atributos en cada dominio de definición particular de los atributos. Esto es una manera de mapear estas diferencias en el intervalo $[0, 1]$, con el objetivo de darle sentido al hecho de sumar estos valores. Las disimilitudes también fueron seleccionadas de manera aleatoria, tratando de obtener un conjunto de particiones bien heterogéneos.

3.4.1. Configuración de los algoritmos

En este capítulo se introdujeron dos algoritmos de combinación de agrupamientos WEA (Sección 3.2) y HPCK (Sección 3.3). Estos tienen en común el uso de la matriz de asociación pesada (3.3). En los experimentos se utiliza una versión normalizada del *valor de asociación* λ_{ij} (ver ecuación (3.3)) con el objetivo de asegurar que cada componente de la matriz WA tenga la misma relevancia. Se usa

$$\lambda_{ij}(P) = \begin{cases} \frac{|P|/|P_{max}|}{|C|/|C_{max}|} \cdot \frac{\Gamma_P(x_i,x_j)}{\Gamma_P^{max}}, & \exists C \in P \ (x_i \in C \wedge x_j \in C) \\ 0, & \text{en otro caso} \end{cases}$$

donde $|P_{max}|$ denota la cardinalidad de la partición con más grupos en $\mathbb{P}$, $|C_{max}|$ denota la cardinalidad del grupo con más elementos entre todas las particiones y Γ_P^{max} el máximo valor de similitud entre pares de objetos en X.

En el algoritmo WEA se utilizaron los siguientes algoritmos de agrupamiento

jerárquico: Single-Link, Complete-Link y Average-Link. Se denota WEA-S, WEA-C, WEA-A cada uno de estos casos. Además, se utilizaron estos mismos algoritmos jerárquicos en los métodos EA [37] y PA [120], utilizando una notación análoga.

En el algoritmo HPCK, en el paso de Análisis de la Importancia de las Particiones (AIP), se usaron cuatro índices de validación: Varianza, Conectividad, Ancho de la Silueta e índice de Dunn [49], (la definición de cada uno de ellos puede encontrarse en la Sección 2.4.2). Estos constituyen diferentes formas de medir compacidad, separación y conectividad entre grupos. A pesar de que estos fueron definidos para datos numéricos, pueden ser utilizados en el paso paso AIP del método HPCK sin importar el tipo de datos originales del problema (e.g., datos categóricos o mezclados). Esto se debe a que en el algoritmo HPCK estos índices son aplicados sobre la nueva representación de los datos obtenida en el paso Unificación de Información.

3.4.2. Experimentación y análisis

Se usan las tres primeras colecciones de datos para probar el algoritmo WEA. En la Tabla 3.2 se calcula el CER de los algoritmos EA, PA y WEA sobre las colecciones de datos Iris, Wine, Breast-Cancer y Half-Ring. También se presenta el promedio de CER para cada algoritmo en todas las colecciones de datos. El CER no es siempre una medida precisa, ya que esta impone una penalización muy fuerte si la partición obtenida no tiene el mismo número de grupos que la estructuración de referencia (ground-truth). Por lo tanto, en la Tabla 3.3 se presenta la evaluación con otros índices de validación de los resultados obtenidos para la colección de datos Iris en el caso de $m = 50$ particiones. Con este experimento, se muestra la calidad del algoritmo propuesto, en comparación con algoritmos de agrupamiento simples así como otros algoritmos de combinación de agrupamientos. De esta manera se ratifica experimentalmente, la capacidad de la matriz de asociación pesada para extraer información valiosa del conjunto de particiones.

Tabla 3.2: Porciento de CER de diferentes algoritmos en diferentes colecciones de datos

B. D.	m	Prom	EA-S	EA-C	EA-A	PA-S	PA-C	PA-A	WEA-S	WEA-C	WEA-A
Iris	25	44,9	33,3	40,0	54,6	29,3	40,0	29,3	33,3	38,6	**18,0**
Iris	50	42,2	33,3	28,0	33,3	33,3	21,3	18,0	33,3	12,6	**4,6**
Wine	25	25,5	28,6	7,30	12,9	28,6	7,30	12,9	28,6	**3,37**	12,9
Wine	50	22,7	28,6	**3,37**	12,9	28,6	**3,37**	12,9	28,6	**3,37**	7,30
H-Rings	25	43,0	67,0	67,0	20,0	67,0	39,0	25,0	20,0	67,0	**19,5**
H-Rings	50	51,0	67,0	55,5	39,5	67,0	45,0	39,5	70,0	**19,0**	45,2
Average	-	38,2	42,9	33,5	28,8	42,3	26,0	22,9	35,6	24,0	**17,9**

Tabla 3.3: Evaluación de los resultados obtenidos con la colección de datos Iris usando diferentes índices de validación externos.

CVI	Prom	EA-S	EA-C	EA-A	PA-S	PA-C	PA-A	WEA-S	WEA-C	WEA-A
F	0,81	0,59	0,62	0,59	0,59	**0,96**	0,69	0,59	0,95	0,91
Purity	0,83	0,66	0,73	0,66	0,66	**0,97**	0,82	0,66	**0,97**	0,95
Rand	0,79	0,77	0,78	0,77	0,77	0,89	0,82	0,77	0,91	**0,94**
NMI	0,64	0,73	0,68	0,73	0,73	0,70	0,73	0,73	0,78	**0,84**
CER	42,2	33,3	28,0	33,3	33,3	21,3	18,0	33,3	12,6	**4,6**

En la Tabla 3.4, se presentan los pesos asignados a diez particiones de la colección de datos Wine utilizando los algoritmos WPCK y HPCK. Los resultados similares obtenidos en este experimento muestran que la nueva representación utilizada en el método HPCK mantiene la relación entre objetos en la colección de datos. Esto justifica experimentalmente el hecho de aplicar los índices de validación internos en el paso AIP del método HPCK sobre la nueva representación de los datos. Después del paso AIP, la función de consenso fue aplicada a ambos algoritmos obteniéndose los mismos resultados finales. De esta manera, se puede apreciar que estos métodos obtienen resultados muy similares, sin embargo el HPCK puede trabajar en situaciones donde el WPCK no puede, por ejemplo: con datos categóricos y mezclados así como cuando se utilizan diferentes representaciones de los datos y diferentes medidas de (di)similitud entre objetos en el proceso de generación de las particiones.

Tabla 3.4: Pesos asignados a diez particiones de la colección de datos Wine en el paso AIP.

Algoritmo	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
WPCK (Ω)	0,67	0,73	0,85	**1,0**	0,93	0,92	0	0,84	0,82	0,43
HPCK (Ω)	0,52	0,61	0,80	**1,0**	0,82	0,80	0	0,75	0,65	0,32

Tabla 3.5: Porciento de CER de varios algoritmos de combinación de agrupamiento utilizando datos categóricos y mezclados

Colección de datos	Prom	EA-A	PA-A	WEA-A	HGPA	CSPA	MCLA	HPCK
Zoo	33,8	31,6	14,8	14,8	37,6	37,6	23,7	**13,8**
Auto	74,3	70,2	70,2	**68,3**	79,5	77,5	71,7	69,2
Votes	15,9	13,5	15,6	14,2	17,2	16,7	14,2	**12,8**
Soybeans	26,3	19,1	19,1	19,1	23,0	14,8	**8,51**	**8,51**
Promedio	37,5	33,6	29,9	29,1	39,3	36,6	29,5	**26,1**

En la Tabla 3.5 se presenta el desempeño del método propuesto HPCK con datos categóricos y mezclados. En este experimento, también se utilizan los algoritmos de combinación de agrupamientos WA, PA, WEA y los basados en particionamiento de (hiper)grafos HGPA, CSPA, MCLA [99]. Los buenos resultados obtenidos por el

algoritmo HPCK asegura una vez más la precisión de la matriz de asociación pesada. Además, el uso del paso AIP sobre la nueva representación de los datos obtenida en el paso Unificación de Información se justifica experimentalmente. Finalmente, se puede decir que el algoritmo HPCK extiende los buenos resultados del algoritmo WPCK al dominio de los datos categóricos y mezclados.

Capítulo 4

DOS PROBLEMAS DE ESTRUCTURACIÓN BAJO EL ENFOQUE DE LA COMBINACIÓN DE AGRUPAMIENTOS

Existen problemas relacionados con la estructuración de un conjunto de datos que pueden ser formulados a partir de la idea de la combinación de agrupamientos. En este capítulo se estudian dos de estos problemas: la selección del nivel representativo de una jerarquía de particiones (Sección 4.1) y la combinación de diferentes segmentaciones de una imagen (Sección 4.2).

4.1. Selección del nivel representativo de una jerarquía de particiones

Los algoritmos de agrupamiento pueden ser divididos en *Particionales* o *Jerárquicos* [55]. Los algoritmos particionales crean una partición de los datos agrupando los objetos en grupos de acuerdo a sus valores de (di)similitud. Por otra parte, los algoritmos jerárquicos construyen una jerarquía de particiones anidadas. Esta jerarquía usualmente se asocia a un *dendrograma*, el cual puede ser cortado a diferentes niveles para obtener las diferentes particiones en la jerarquía (ver Fig. 4.1).

Las jerarquías de particiones pueden ofrecer más información acerca de la estructura de los objetos en la colección de datos. Con una jerarquía, los objetos se agrupan

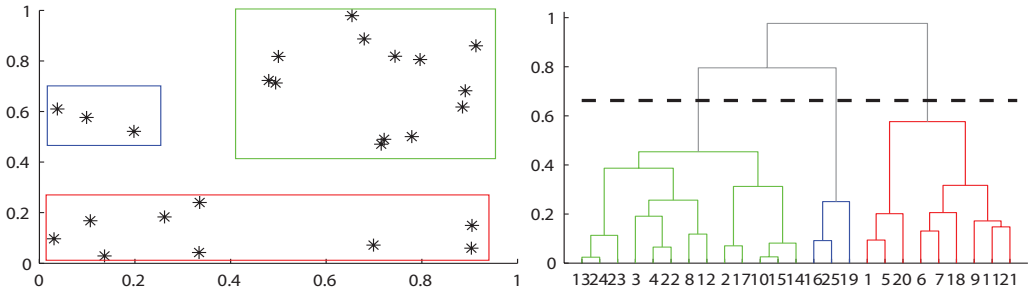


Figura 4.1: Una colección de datos formada por 25 puntos en el plano y el dendrograma producido por el algoritmo Average-Link. La línea discontinua que corta el dendrograma (derecha) produce una partición de los objetos con 3 grupos (izquierda).

a diferentes niveles; desde el nivel inferior donde cada objeto conforma un grupo hasta el nivel superior donde solo hay un grupo que contiene a todos los objetos. Sin embargo, en problemas prácticos, trabajar con la jerarquía completa es en ocasiones demasiado complejo. En estos casos, es necesaria la selección de una partición, la cual representará a la jerarquía completa y sobre la cual se realizará cualquier tipo de procesamiento requerido en el problema.

En el enfoque tradicional, esta partición *representativa*[1] se obtiene mediante la utilización de índices de validación de agrupamientos (internos) (CVIs). Cada partición en la jerarquía se evalúa con un índice y se selecciona la partición con mejores resultados. En la actualidad existe una gran cantidad de CVIs, por ejemplo, en [78] se presentaron y evaluaron experimentalmente 30 de estos índices. Los CVIs clásicos como el índice de *Calinski-Harabasz* (CH), el índice de *Hartigan* (HA) y el índice de *Dunn* [125] junto con el *Highest-Lifetime*(HL) [37] son los más utilizados para este propósito de seleccionar la partición representativa de una jerarquía.

La principal desventaja del enfoque de usar CVIs para determinar el nivel *representativo* en una jerarquía es que, como se vio en la introducción de este documento, no existe un CVI capaz de trabajar *correctamente* para todas las colecciones de datos y para todos los algoritmos de agrupamiento. En otras palabras, cada CVI evalúa una partición, de acuerdo al cumplimiento de una propiedad particular que viene dada por la definición matemática del índice. Si la propiedad no está relacionada con el criterio de agrupamiento del algoritmo utilizado para generar la jerarquía y con

[1]Se utilizarán indistintamente los términos: partición representativa y nivel representativo, ya que en una jerarquía de particiones los diferentes niveles determinan las particiones en la jerarquía.

las características de los datos del problema en cuestión, la evaluación del índice no aportará información útil para los usuarios del problema.

En este capítulo, se propone un nuevo enfoque para la selección de la partición representativa en una jerarquía basado en la filosofía de la combinación de agrupamientos. Se llamará este nuevo enfoque *Selección de la Partición Representativa basado en Combinación de Agrupamientos* (Partition Selection based on Clustering Ensemble; PSCE). Con el PSCE se define la partición representativa en una jerarquía teniendo en cuenta la evaluación de diferentes CVIs, así como las similitudes entre particiones en la jerarquía. De esta manera, se selecciona como resultado la partición que *mejor* representa las características comunes en la jerarquía.

En la Sección 4.1.1 se presenta formalmente el enfoque propuesto. En la Sección 4.1.2 se hace un análisis de la complejidad computacional y en la Sección 4.1.3 se discuten algunos resultados experimentales utilizando diferentes colecciones de datos y diferentes algoritmos de agrupamiento jerárquicos. Además, se compara el enfoque propuesto con el enfoque tradicional.

4.1.1. Determinación del nivel representativo de una jerarquía a través de la combinación de particiones

Cuando se aplica un algoritmo de agrupamiento jerárquico a un conjunto de objetos X, se obtiene un jerarquía de particiones. Una jerarquía $\mathbb{H} = \{P_1, P_2, \ldots, P_m\}$ es un conjunto de particiones anidadas[2] de X, donde $P_i \preceq P_j, \forall 1 < i < j < m$. Es fácil de verificar que $\mathbb{H} \subset \mathbb{P}_X$. Luego, se define el nivel representativo en la jerarquía como la partición que mejor resume la información en la jerarquía $\mathbb{H}$ teniendo en cuenta dos parámetros: Primero, la evaluación de diversos CVIs en todas las particiones en la jerarquía. Segundo, los valores de similitud entre cada par de particiones en la jerarquía. Formalmente, la partición representativa $\hat{P}$ en la jerarquía $\mathbb{H}$ se define como:

$$\hat{P} = \arg\max_{P \in \mathbb{H}} \sum_{i=1}^{m} (\mathcal{E}(P_i) \cdot \Gamma(P, P_i)) \tag{4.1}$$

donde $\mathcal{E}(P_i)$ es una evaluación de cada partición $P_i \in \mathbb{H}$ y Γ una medida de similitud entre particiones. Esta evaluación se puede utilizar para dar más importancia a particiones que cumplan determinadas propiedades que sean de interés para un problema determinado. Notar que en este caso, a diferencia del problema de la partición

[2] $\preceq$ es la relación de order parcial "*anidado en*", ver Definición 1.1.

mediana original (1.1), la partición buscada $\hat{P}$ es una de las particiones que se desea combinar como se muestra en la ecuación (4.1), es decir $\hat{P} \in \mathbb{H}$. Por lo tanto, este problema es más sencillo que el problema original de la partición mediana, ya que el espacio de búsqueda aquí ($\mathbb{H}$) es mucho más pequeño[3] que el espacio de búsqueda ($\mathbb{P}_X$) en (1.1).

El método de combinación de agrupamientos basado en funciones núcleo (WPCK) definido en el Capítulo 2, presenta las siguientes características que resultan convenientes para la definición del problema de selección del nivel representativo de una jerarquía, basado en la filosofía del mismo:

- Es posible calcular un valor de peso para cada partición teniendo en cuenta la evaluación de diferentes índices de validación de agrupamientos. Los valores de peso ω_i asignados a cada partición P_i pueden ser usados como valor $\mathcal{E}(P_i)$ en la ecuación (4.1).

- Es sencillo restringir la búsqueda solamente a las particiones en $\mathbb{H}$. En WPCK, se calcula como primer paso la mejor partición en el conjunto de particiones a combinar y esta solución es posteriormente mejorada mediante una búsqueda en el espacio de búsqueda completo $\mathbb{P}_X$. Como se verá más adelante, para determinar la partición representativa solo será necesario aplicar el primer paso.

- El algoritmo está teóricamente fundamentado y tiene un bajo costo computacional, $\mathcal{O}(n \cdot m \cdot rMax)$.

De esta manera, los pasos del algoritmo propuesto para encontrar la partición representativa en una jerarquía son los siguientes: *selección de una subjerarquía*, *evaluación de las particiones* y *obtención de la partición representativa.*

Selección de una subjerarquía:

Este paso consiste en la extracción de un subconjunto de particiones de la jerarquía $\mathbb{H}$. Cada partición en la jerarquía tiene un número diferente de grupos. Consecuentemente, se selecciona un subconjunto de particiones que tengan un número de grupos en un rango *razonable*. Este rango es un parámetro del algoritmo, por ejemplo, $[2, 10]$, $[2, 30]$ o $[2, \sqrt{n}]$ podrían ser utilizados. Se denota por $\mathbb{H}_{[q,t]}$ la subjerarquía de $\mathbb{H}$, donde P_q es el nivel superior, P_t es el nivel inferior, y cada partición $P_s \in \mathbb{H}$ con s grupos pertenece a $\mathbb{H}_{[q,t]}$ si y solo si $q \leq s \leq t$. Por simplicidad, se denota $v = t - q + 1$

[3] $\mathbb{H}$ está compuesto a lo sumo por n particiones, siendo n el número de objetos en X.

el número de particiones en $\mathbb{H}_{[q,t]}$. La jerarquía completa $\mathbb{H}$ puede ser utilizada, es decir, seleccionar el rango $[1, n]$ $(v = n)$. Sin embargo, se recomienda la utilización de rangos más pequeños para disminuir el costo computacional del algoritmo.

Evaluación de las particiones:

Se obtiene el valor de evaluación de cada partición a través de la aplicación de diferentes índices de validación de agrupamientos. En este caso se puede seguir la misma idea utilizada para el método de combinación de agrupamientos WPCK propuesto en el Capítulo 2. Es decir, se define para cada partición $P_i \in \mathbb{H}_{[q,t]}$ el valor $\mathcal{E}(P_i) = \omega_i$, donde ω_i es el peso calculado para la partición P_i en el proceso de Análisis de la Importancia de las Particiones (ver Sección 2.1), asumiendo que $\mathbb{H}_{[q,t]}$ es el conjunto de particiones a combinar en el método WPCK. En este caso, índices como *Calinski-Harabasz* (CH), *Hartigan* (HA) y *Highest-Lifetime* (HL) pueden ser fácilmente transformados para satisfacer la definición de Índice de Validación de Propiedades (IVP) (ver Sección 2.1) y por tanto pueden ser utilizados en este proceso.

Obtención de la partición representativa:

Primeramente, al igual que en el método WPCK, se pueden utilizar cualesquiera de las medidas de similitud entre particiones que sean funciones núcleo. Por tanto se asume que se utiliza una función núcleo k en la definición del problema (4.1). Según los resultados obtenidos en la Sección 2.3, para el problema (2.1) la distancia de cada partición al consenso teórico se puede calcular a partir de la ecuación (2.10). Estos resultados se pueden utilizar para hallar la solución exacta $\hat{P}$ del problema (4.1) de la selección del nivel representativo en una jerarquía de particiones. Esta solución viene dada como la solución del problema

$$\hat{P} = \arg \min_{P \in \mathbb{H}_{[q,t]}} \|\phi(P) - \psi\|^2_{\mathcal{H}}$$

es decir, se puede encontrar la partición representativa calculando la distancia de cada partición en la jerarquía $\mathbb{H}_{[q,t]}$ al consenso teórico ψ utilizando la ecuación (2.10), y seleccionando la partición más cercana a ψ.

4.1.2. Análisis de la complejidad computacional

El cálculo de todos los valores de pesos para todas las particiones es $\mathcal{O}(v \cdot r \cdot f(\mathbb{I}))$, donde v es el número de particiones en $\mathbb{H}_{[q,t]}$, r es el número de IVPs utilizados y $f(\mathbb{I})$ es el costo computacional del índice más costoso. En la práctica, r es un número pequeño, entonces, se puede considerar $\mathcal{O}(v \cdot f(\mathbb{I}))$ el costo computacional del mecanismo de asignación de pesos. Dado los valores de pesos, es necesario calcular la ecuación (2.10) para cada partición en $\mathbb{H}_{[q,t]}$. La cual, según el análisis en la Sección 2.3 se puede resolver en $\mathcal{O}(v \cdot n)$ para una partición, y para las v particiones en $\mathbb{H}_{[q,t]}$ puede calcularse en $\mathcal{O}(v^2 \cdot n)$. Finalmente, la complejidad computacional global de la selección del nivel representativo es $\mathcal{O}(v \cdot f(\mathbb{I})) + \mathcal{O}(v^2 \cdot n)$. Con una apropiada selección de los índices $\mathbb{I}$ y la subjerarquía $\mathbb{H}_{[q,t]}$, este costo computacional es menor que $O(n^2)$, que es la complejidad común para los algoritmos de agrupamiento jerárquicos. Sin embargo, en el peor caso (v cercano a n) la complejidad computacional del algoritmo será $\mathcal{O}(n^3)$. De aquí la importancia de una apropiada selección de la subjerarquía y de los IVPs.

4.1.3. Resultados experimentales

En los experimentos se usaron 8 colecciones de datos, 5 son de la UCI Machine Learning Repository [36] y las otras 3 son colecciones de datos sintéticas compuestas por puntos en el plano (ver Tabla 4.1). Para todas estas colecciones de datos la estructuración de referencia (ground-truth) está disponible. Por lo tanto, en los experimentos, se comparan los resultados obtenidos con la estructuración de referencia de cada colección de datos. Se utiliza la medida Información Mutua Normalizada (NMI) para evaluar los resultados de los algoritmos. Esta es una medida de similitud entre particiones muy utilizada que evalúa una partición a partir de la información que esta comparte con la estructuración de referencia.

Tabla 4.1: Descripción de las colecciones de datos

Nombre	**Obj-por-clases**	**Colecciones de datos sintéticas**		
Cassini	120-60-120			
Half-Rings	100-100			
Smiley	33-33-50-84			
Wine	59-41-78			
Opt-Digits	10-11-11-11-12-5-8-12-9-11			
Iris	50-50-50			
Glass	70-76-17-13-9-29			
Ionosphere	126-225	Cassini	Half-Rings	Smiley

En cada experimento, las jerarquías se obtuvieron utilizando 3 algoritmos de

agrupamiento jerárquicos muy conocidos: Single-Link (SL), Complete-Link (CL) y Average-Link (AL) [55]. Para cada colección de datos, se comparan los resultados obtenidos por el enfoque propuesto (PSCE) y el enfoque basado en el uso de CVIs con los siguientes índices: *Highest-Lifetime*(HL), *Calinski-Harabasz* (CH) y *Hartigan* (HA). En la Tabla 4.2, estos se denotan por CV-HL, CV-CH y CV-HA respectivamente. Además se presenta para cada algoritmo el valor "*Más Cercano al Ground-Truth*" (MCG), el cual es calculado mediante la evaluación de todas las particiones en la jerarquía con respecto a la estructuración de referencia usando la medida NMI y tomando el mayor valor. Bebe notarse que el valor MCG depende de la calidad de las jerarquías. Además, los resultados de CV-HL, CV-CH, CV-HA y PSCE están acotados superiormente por el valor MCG de cada jerarquía.

Las jerarquías fueron generadas mediante la aplicación de los algoritmos de agrupamiento jerárquicos SL, CL, y AL en las 8 colecciones de datos. En todos los casos, se usa una subjerarquía $\mathbb{H}_{[2,35]}$ compuesta por particiones con s grupos, tal que $2 \leq s \leq 35$. Para todas las jerarquías generadas, el valor MCG se obtuvo en una partición de la subjerarquía $\mathbb{H}_{[2,35]}$. Por tanto, el rango $[2, 35]$ es apropiado para estos experimentos y permite un decrecimiento considerable del costo computacional de los algoritmos. Los resultados se evaluaron utilizando la medida NMI y en cada caso, se resalta el mejor resultado.

Tabla 4.2: Comparación de los métodos CV-HL, CV-CH, CV-HA y PSCE para la selección de la partición representativa en una jerarquía. Cada casilla en la columna del extremo derecho (Prom) es el valor promedio de las casillas en su fila.

Alg	Método	Cassini	Half-R	Smiley	Wine	Opt-D	Iris	Glass	Ionosp	Prom
SL	CV-HL	**0,941**	**0,720**	0,846	**0,102**	0,706	**0,733**	**0,154**	**0,076**	0,534
	CV-CH	**0,941**	0,488	**0,863**	0,092	0,250	**0,733**	**0,154**	0,008	0,441
	CV-HA	**0,941**	0,488	**0,863**	0,092	0,250	0,545	**0,154**	**0,076**	0,426
	PSCE	**0,941**	**0,720**	0,853	**0,102**	**0,798**	0,720	**0,154**	**0,076**	**0,545**
	MCG	0,970	0,961	1,0	0,502	0,801	0,733	0,394	0,129	0,686
CL	CV-HL	0,657	0,197	0,712	**0,790**	0,789	**0,756**	0,446	0,143	0,561
	CV-CH	0,551	0,353	0,291	0,665	0,250	**0,756**	0,442	0,037	0,418
	CV-HA	0,522	0,353	0,646	0,709	0,723	**0,756**	0,442	0,037	0,523
	PSCE	**0,743**	**0,393**	**0,820**	0,709	**0,805**	**0,756**	**0,516**	**0,160**	**0,612**
	MCG	0,792	0,442	0,865	0,798	0,825	0,756	0,590	0,193	0,657
AL	CV-HL	**0,779**	0,066	**0,766**	0,693	0,730	0,643	0,452	0,082	0,526
	CV-CH	**0,779**	0,347	0,685	**0,775**	0,250	**0,685**	0,452	0,082	0,506
	CV-HA	0,513	0,347	0,623	**0,775**	0,712	0,643	0,452	0,082	0,518
	PSCE	**0,779**	**0,433**	0,728	**0,775**	**0,814**	0,661	**0,454**	**0,083**	**0,590**
	MCG	0,792	0,474	0,883	0,775	0,843	0,783	0,501	0,169	0,652

Se usan 5 índices para el cálculo de los pesos asociados a las particiones en el enfoque propuesto: Varianza, Conectividad, HL, CH y HA. Los dos primeros son índices sencillos que miden compacidad y conectividad respectivamente (ver Sección 2.4.2). Los otros 3 índices son los mismos que se utilizaron, de manera independiente, en el

enfoque tradicional (basado en el uso de índices de validación). Sin embargo, en todos los casos estos se normalizaron al rango $[0, 1]$ para satisfacer la definición de Índice de de Validación de Propiedades. No se reportan los resultados de los índices Varianza y Conectividad usados en el enfoque tradicional debido a que los resultados fueron muy malos. La simplicidad de estos índices no permite que estos puedan funcionar por si solos como criterios de selección del nivel representativo con cierto grado de eficacia. Sin embargo, en el enfoque PSCE estos pueden ser muy útiles, ya que cada índice evalúa las particiones desde una perspectiva diferente y todos estos puntos de vista se combinan para obtener el resultado final.

En la Tabla 4.2, se resumen los resultados experimentales. De la última columna de esta tabla puede verse como el PSCE tiene un mejor desempeño promedio en todos los casos. En las jerarquías obtenidas con el Single-Link (SL), CV-HL y PSCE trabajan de manera similar. Sin embargo, para el Complete-Link (CL) y el Average-Link (Al), el PSCE supera claramente a los otros métodos. Los resultados en esta tabla corroboran la capacidad del enfoque PSCE para trabajar bien en diferentes circunstancias, es decir, cuando se utilizan diferentes algoritmos de agrupamiento y diferentes colecciones de datos.

De la Tabla 4.2 también puede verse que el valor MCG casi nunca se alcanza. Esto ratifica que un simple índice no puede trabajar correctamente para todas las colecciones de datos en el caso del enfoque basado índices de validación. Además, esto significa que se puede utilizar un conjunto de índices más apropiado con el objetivo de mejorar un poco más los resultados del enfoque PSCE.

4.2. Combinación de segmentaciones de una imagen

De manera similar a como ocurre con los algoritmos de agrupamiento, no existe un algoritmo de segmentación de imágenes capaz de trabajar correctamente para todo tipo de imágenes. Diferentes algoritmos de segmentación[4] pueden producir segmentaciones muy diferentes de una misma imagen. Por tanto, si se tienen varias segmentaciones de una imagen, la idea de combinarlas siguiendo la filosofía de los algoritmos de combinación de agrupamientos parece ser una variante apropiada.

Una primera idea puede ser utilizar los algoritmos de combinación de agrupamientos directamente en problemas de segmentación de imágenes. Es decir, mirar una imagen como un conjunto de pixeles y una segmentación como una partición

[4] O el mismo algoritmo pero aplicado utilizando diferentes configuraciones de los parámetros.

de este conjunto de pixeles y de esta manera aplicar los algoritmos de combinación de agrupamientos existentes. De hecho, algunos de los algoritmos de combinación de agrupamientos han sido evaluados experimentalmente a partir de sus resultados en la combinación de segmentaciones de imágenes [37, 98]. Otros trabajos donde se explora la idea de combinación de segmentaciones, pero principalmente dedicados a aplicaciones o problemas específicos dentro de la segmentación de imágenes se encuentran en [19, 32, 122, 121, 129, 131]. Sin embargo, esta forma de enfrentar el problema de combinación de segmentaciones tiene tres desventajas fundamentales:

- Una imagen está compuesta por un gran número de pixeles. Por ejemplo, una imagen de 500×500 pixeles equivaldría a una colección de datos de 250000 objetos, lo cual dificultaría la eficiencia de los algoritmos de combinación de agrupamientos.
- Al asumir que una imagen es solamente un conjunto de pixeles se pierde la relación espacial existente entre los pixeles en una imagen.
- No todas las particiones del conjunto de pixeles forman una *segmentación válida* (Ver Figura 4.2).

En el método propuesto en este capítulo[5] se le da una solución a estos problemas. Se presenta un enfoque de combinación de segmentaciones basado en la combinación de agrupamientos utilizando funciones núcleo, en el cual se tienen en cuenta tanto la dimensionalidad de la imagen como la relación espacial de los pixeles.

Primeramente, en la Sección 4.2.1 se presenta la idea del uso de los super-pixeles [98], la cual es una variante de solución al problema de la dimensionalidad de la imagen. Sin embargo, a diferencia de la propuesta inicial de los super-pixeles en [98], en la definición propuesta, cada super-pixel estará representado por un conjunto de características que permiten la aplicación de índices de validación para evaluar cada segmentación y asignarles un peso de acuerdo a su importancia para el proceso de combinación. Por otra parte, se definen formalmente los conceptos de *imagen*, *segmentación* y se plantea el problema de la combinación de segmentaciones a partir del problema de la partición mediana pesada (2.1), pero en un espacio de búsqueda más reducido. Posteriormente en la Sección 4.2.2 se presentan los pasos del algoritmo propuesto en este capítulo y finalmente en la Sección 4.2.3 se realiza un estudio experimental utilizando diferentes imágenes y segmentaciones de las mismas.

[5]Los resultados que se presentan en este capítulo fueron obtenidos en colaboración con el grupo de investigación de Reconocimiento de Patrones y Procesamiento de Imágenes de la Universidad de Münster, Alemania, bajo la dirección del Prof. Dr. Xiaoyi Jiang.

4.2.1. Planteamiento formal del problema

Para lidiar con la gran cantidad de pixeles que se encuentran en una imagen, se presenta la idea del uso de super-pixeles. El cómputo de los super-pixeles es un procedimiento simple e intuitivo con el cual se puede disminuir de manera considerable la cantidad de objetos que representan una imagen. La idea de los super-pixeles está motivada por el hecho de que pixeles *vecinos* que estén agrupados en una misma región o grupo en todas las segmentaciones que se desean combinar, no tienen por qué ser analizados de forma separada. Es de esperar que estas *regiones conectadas* de pixeles que estuvieron juntos en el mismo grupo en todas las segmentaciones estén también en un mismo grupo en la segmentación de consenso. Por tanto, se puede seleccionar un solo representante o super-pixel por cada una de estas regiones.

Para llegar a una definición más formal de super-pixel es necesario definir un conjunto de conceptos básicos. Dada una imagen digital de dimensiones $w \times h$, donde w representa el ancho y h la altura, para cada uno de los $w \cdot h$ pixeles que la componen se asumen conocidas las siguientes 5 características: x, y, R, G y B. Las dos primeras denotan la posición o coordenadas del pixel dentro de la imagen y las tres últimas son los valores de rojo, verde y azul respectivamente que componen el color del pixel.

Dados dos pixeles p_1 y p_2 con coordenadas $(\mathsf{x}_1, \mathsf{y}_1)$ y $(\mathsf{x}_2, \mathsf{y}_2)$ respectivamente, se dice que estos son *vecinos*[6] si se satisface que $|\mathsf{x}_1 - \mathsf{x}_2| + |\mathsf{y}_1 - \mathsf{y}_2| = 1$.

Un *camino* es una secuencia de pixeles $p_1, p_2, \ldots, p_t$ en la cual todo par de pixeles consecutivos p_i, p_{i+1} son *vecinos*. Se dice que un conjunto C de pixeles es una *componente conexa* de una imagen, si para todo par de pixeles $p, p' \in C$ se cumple que existe un camino $p_1, \ldots, p_t$ tal que $p = p_1$, $p' = p_t$, donde $p_i \in C, \forall i = 1, \ldots, t$.

De esta manera, los super-pixeles son componentes conexas dentro de la imagen formadas por pixeles que estuvieron en el mismo grupo en todas las segmentaciones que se desean combinar. Igualmente, se dice que dos super-pixeles sp_1, sp_2 son *vecinos* si existe al menos un par de pixeles p, p' vecinos, de forma tal que $p \in sp_1$ y $p' \in sp_2$.

Para desarrollar el método propuesto en este capítulo, es necesario dar una definición formal de los conceptos *Imagen* y *Segmentación* a partir de los super-pixeles. Como se puede apreciar en la Figura 4.2, es posible tener un grafo asociado a una imagen de forma tal que este represente las relaciones de conectividad entre los super-pixeles que la conforman. Este grafo será de vital importancia para la definición de *Imagen* y *Segmentación* dadas a continuación.

[6] En este caso se define la 4-vecindad en una imagen (ver [62] para más información sobre Topología Digital).

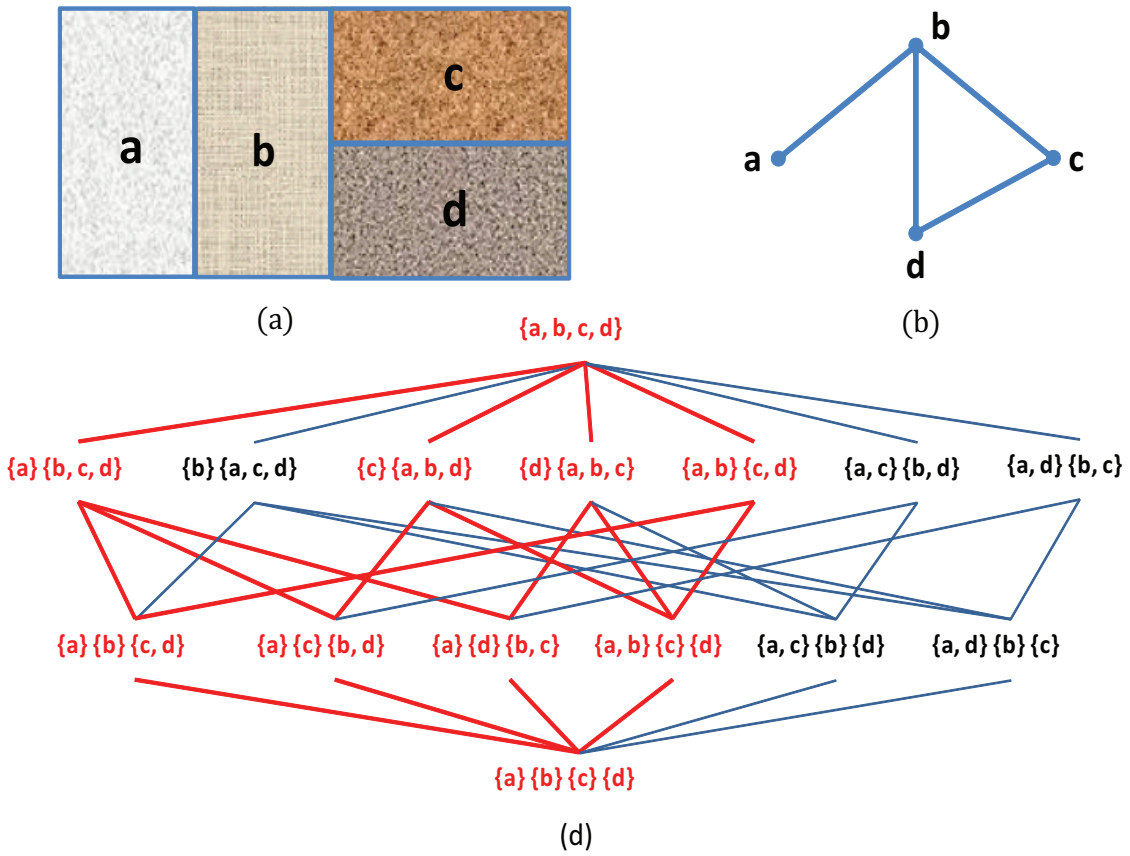


Figura 4.2: Imagen compuesta por el conjunto de super-pixeles $\{a, b, c, d\}$ como se muestra en (a). La relación de conectividad de estos super-pixeles queda expresada en el grafo presentado en (b). En (c) se muestran en rojo las segmentaciones posibles con este conjunto de super-pixeles. Se debe notar que es un subconjunto del conjunto de todas las posibles particiones del conjunto de super-pixeles.

Definición 4.1. *Una imagen es un par* $\mathbb{I} = (\mathbb{I}_S, \mathbb{I}_G)$*, donde* $\mathbb{I}_S = \{sp_1, sp_2, \ldots, sp_n\}$ *es un conjunto de super-pixeles.* $\mathbb{I}_G = (V, E)$ *es el* grafo de conectividad *de los super-pixeles, donde existe un nodo* $v_i \in V$ *asociado a cada super-pixel* sp_i *y existe una arista* $e_{ij} \in E$ *si y solo si los super-pixeles* sp_i *y* sp_j *son vecinos.*

Proposición 4.1. *El grafo* $\mathbb{I}_G$ *es un grafo planar.*

Esta proposición se enuncia sin demostración debido a que no es complicado comprobar que es cierta a partir de la construcción del grafo. En este grafo solo existen aristas entre super-pixeles vecinos, por tanto es posible dibujar todas las aristas de manera tal que ninguna de estas se cruce con otra en puntos distintos de los nodos del grafo.

Definición 4.2. *Una segmentación* $S = \{R_1, R_2, \cdots, R_d\}$ *es un conjunto de regiones*

$R_i \subseteq \mathbb{I}_S$ que satisface las cuatro propiedades siguientes:

- $R_i \neq \emptyset, \forall i = 1, \ldots, d$
- $\bigcup_{i=1}^{d} R_i = \mathbb{I}_S$
- $R_i \cap R_j = \emptyset, \forall i, j = 1, \ldots, d$, *con* $i \neq j$
- $\forall R \in S$, *se cumple que el grafo* R_G *es un grafo conexo.* R_G *es el subgrafo de* $\mathbb{I}_G$ *que se obtiene al eliminar todos los nodos que no están asociados a un super-pixel de* R.

Las tres primeras propiedades aseguran que una segmentación es una partición del conjunto de super-pixeles. Por otra parte, la última propiedad garantiza que en una segmentación, cualquier par de super-pixeles en una región estén conectados por un *camino* de super-pixeles que también pertenezcan a dicha región. En la Figura 4.2 (c) se ve cómo solo las particiones en color rojo satisfacen esta última propiedad.

A partir de las definiciones anteriores, se puede observar que, dada una imagen $\mathbb{I}$ el conjunto de todas las segmentaciones posibles $\mathbb{S}_\mathbb{I}$ es un subconjunto del conjunto de todas las posibles particiones del conjunto de super-pixeles que conforman la imagen. De esta manera, dado un conjunto de segmentaciones $\mathbb{S} = \{S_1, S_2, \ldots, S_m\}$, la segmentación de consenso S^* se define como:

$$S^* = \arg\max_{S \in \mathbb{S}_\mathbb{I}} \sum_{i=1}^{m} \omega_i \cdot k(S, S_i) \tag{4.2}$$

donde ω_i es un peso asociado a cada segmentación S_i y k es una medida de similitud entre particiones que cumple la propiedad de ser una función núcleo. Para calcular los pesos ω_i se puede utilizar el mecanismo de asignación de pesos propuesto en la Sección 2.1. La función de similitud puede ser una de las propuestas en la Sección 2.2.

4.2.2. Método propuesto

El primer paso del algoritmo propuesto es el cómputo de los super-pixeles. Sin embargo, con el objetivo de obtener una mejor representación de la imagen, cada super-pixel estará conformado por una tupla de características. Esta tupla de características debe, de alguna manera, extraer las propiedades comunes de los pixeles que son representados por dicho super-pixel. En esta representación de los super-pixeles

existen dos tipos de información que se desea extraer de cada región que representa un super-pixel. La primera es información relacionada con la forma de las regiones, es decir, sería conveniente que cada super-pixel pueda almacenar información que permita comparar dos super-pixeles de acuerdo a su forma geométrica. La segunda, es información relacionada con el color, la cual debe permitir una comparación entre super-pixeles de acuerdo a los valores de color de los pixeles que conforman cada región. En este caso, se definen 11 características para representar cada super-pixel:

1. NP: Número de pixeles en la región que representa el super-pixel.

2. R: Promedio de los valores de las componentes rojas (R) de cada pixel en la región que representa el super-pixel.

3. G: Promedio de los valores de las componentes verdes (G) de cada pixel en la región que representa el super-pixel.

4. B: Promedio de los valores de las componentes azules (B) de cada pixel en la región que representa el super-pixel.

5. R_e: Promedio de error en la componente roja (R). Este se calcula sumando los valores absolutos de la diferencia entre el valor de la componente roja de cada pixel y el valor promedio para esa componente en la región del super-pixel.

6. G_e: Promedio de error en la componente verde (G) y se calcula de manera análoga al R_e.

7. B_e: Promedio de error en la componente azul (B) y se calcula de manera análoga al R_e.

8. X: Promedio de las coordenadas x de todos los pixeles en la región del super-pixel.

9. Y: Promedio de las coordenadas y de todos los pixeles en la región del super-pixel.

10. Ra: Radio de la región del super-pixel. Este representa la mayor distancia desde un pixel en esta región hasta la posición (X, Y) calculada previamente[7].

[7] La posición (X, Y) representa la posición del centro de gravedad de la región del super-pixel.

11. Pe: Perímetro de la región del super-pixel. Este viene dado por el número de pixeles en esta región que tienen al menos un pixel vecino en otra región o se encuentran en uno de los límites de la imagen.

Todas estas características asociadas a un super-pixel pueden ser calculadas durante el mismo proceso de generación de los super-pixeles sin aumentar la complejidad computacional de este proceso.

Luego del cálculo de los super-pixeles, se aplica el mecanismo de asignación de pesos según se presentó en la Sección 2.1. Sin embargo, en este caso, se definen índices de validación capaces de extraer información valiosa de las segmentaciones a combinar. En particular en los experimentos realizados se utilizaron los cuatro índices[8] siguientes [130]: *Discrepancia*, E_{CW}, *Compacidad* y *Circularidad*. Los dos primeros son índices que permiten evaluar una segmentación de acuerdo al comportamiento del color en las regiones que la componen y los dos últimos evalúan una segmentación de acuerdo a la forma geométrica de sus regiones.

En el planteamiento del problema (4.2) se puede utilizar cualquiera de las medidas de similitud que satisfacen la propiedad de ser una función núcleo. Sin embargo, en los experimentos, al igual que en secciones anteriores, se utilizó la medida k_S (ver Sección 2.2.2). En este caso, se puede seguir la misma estrategia que en la Sección 2.3 para hallar la partición de consenso. No obstante, como el espacio de búsqueda en este problema es un subconjunto del espacio de búsqueda en el problema general de la partición mediana, es necesario modificar la meta-heurística *recocido simulado* para trabajar con las características del nuevo problema. En realidad, solo es necesario cambiar la definición de vecindad entre dos estados (en este caso segmentaciones[9]) de forma tal que el proceso de definir una segmentación vecina de cierta segmentación inicial satisfaga las siguientes propiedades:

- Solo sean generadas segmentaciones que satisfagan la definición de segmentación dada en la sección anterior. Es importante garantizar que mediante el proceso de construcción de un nuevo estado se asegure que la segmentación generada satisface la definición (sin una verificación adicional).

- Empezando por una segmentación inicial, se pueda alcanzar cualquier segmentación en $\mathbb{S}_{\mathbb{I}}$ en $\mathcal{O}(n)$ pasos del algoritmo.

[8]Estos índices fueron adaptados de su definición original para trabajar sobre la representación de la imagen a partir de los super-pixeles.

[9]Tener en cuenta que también son particiones del conjunto de super-pixeles.

Dada una imagen $\mathbb{I} = (\mathbb{I}_S, \mathbb{I}_G)$, con $\mathbb{I}_G = (V, E)$ y una segmentación de esta imagen S, se propone el siguiente proceso de generación de segmentaciones vecinas. Se seleccionan aleatoriamente un super-pixel $sp \in \mathbb{I}_\mathbb{S}$ y una arista $(sp, sp') \in E$, la cual tiene al nodo que representa el super-pixel sp como uno de los nodos que la conforman. Si el super-pixel sp' no pertenece a la misma región que sp en S, se mueve el super-pixel sp a la región del super-pixel sp'. Por otra parte, si sp y sp' están en la misma región, se crea una nueva región que solo contiene al super-pixel sp. Con este procedimiento se satisfacen las dos propiedades enunciadas previamente. La primera es fácil de verificar debido a que los super-pixeles se mueven de una región a otra de acuerdo a las aristas en el grafo $\mathbb{I}_G$. Para la segunda, se observa que mediante este proceso se puede llegar a la segmentación donde cada super-pixel es una región, en $\mathcal{O}(n)$ y de ahí a cualquier otra segmentación también en $\mathcal{O}(n)$.

Como el grafo $\mathbb{I}_G = (V, E)$ es planar se cumple que si $|V| \geq 3$ entonces $|E| \leq 3 \cdot |V| - 6$, por tanto $|E| = \mathcal{O}(|V|)$, es decir, se pueden almacenar todas las aristas del grafo en memoria con una complejidad espacial lineal respecto a la cantidad de super-pixeles en la imagen. De esta forma, el proceso de generación de segmentaciones vecinas propuesto no afecta la complejidad del algoritmo.

4.2.3. Resultados experimentales

Se denominará al método propuesto en este capítulo *Combinación de Segmentaciones basado en Combinación de Agrupamientos* (Segmentation Combination based on Clustering Ensemble; SCCE).

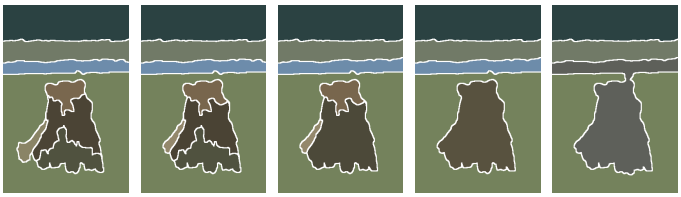
En estos experimentos se utilizan las imágenes a color de la colección de datos Bekerley [74]. Esta es una colección de datos muy utilizada para la evaluación de algoritmos de segmentación y está formada por 300 imágenes naturales de tamaño 481×321. Por cada imagen se utilizaron dos algoritmos de segmentación TBES [87] y UCM [5] para generar, por cada uno, un conjunto de 10 segmentaciones a partir de las cuales se buscará la segmentación de consenso. Las diferentes segmentaciones se obtuvieron en ambos casos variando los parámetros de los algoritmos de segmentación. El conjunto de las 10 segmentaciones obtenidas con el algoritmo TBES se denotarán por *TBES* en la Tabla 4.3, de igual manera, el conjunto de las 10 segmentaciones obtenidas con el algoritmo UCM se denota por *UCM*. Cada imagen en esta colección de datos cuenta con varias segmentaciones hechas por especialistas humanos o segmentaciones de referencia (ground-truth). En los experimentos, los resultados son comparados contra las segmentaciones de referencia utilizando las medidas Informa-

Tabla 4.3: Comparación del método propuesto (SCCE) con los algoritmos de combinación de agrupamientos BOK, BOEM, CSPA, HGPA, MCLA, EA-SL, EA-AL y QMI para la combinación de segmentaciones.

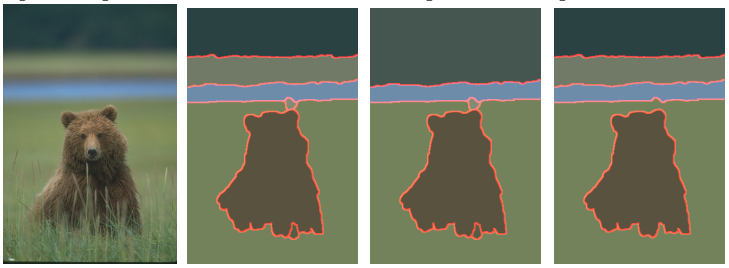
		1 - NMI		VI		1 - RI	
Segm.	**Método**	**Mejor GT**	**Todos GT**	**Mejor GT**	**Todos GT**	**Mejor GT**	**Todos GT**
TBES	BOK	0,41	0,48	1,34	1,73	0,21	0,28
	BOEM	0,35	0,42	1,52	1,82	0,16	0,22
	CSPA	0,33	0,39	1,75	1,99	**0,14**	0,21
	HGPA	0,32	0,38	1,75	1,98	**0,14**	0,21
	MCLA	0,34	0,41	1,47	1,77	0,16	0,22
	EA-SL	0,33	0,39	1,43	1,71	0,16	0,21
	EA-AL	0,32	0,39	1,51	1,78	0,15	0,21
	QMI	0,33	0,39	1,68	1,93	0,15	0,21
	SCCE	**0,30**	**0,37**	**1,41**	**1,71**	**0,14**	**0,20**
UCM	BOK	0,34	0,40	1,90	2,17	0,15	0,21
	BOEM	0,41	0,46	2,20	2,44	0,19	0,25
	CSPA	0,34	0,40	1,90	2,17	0,15	0,21
	HGPA	0,42	0,49	3,67	4,00	0,18	0,27
	MCLA	0,36	0,42	1,91	2,18	0,16	0,22
	EA-SL	0,35	0,41	1,89	2,16	0,15	0,23
	EA-AL	0,35	0,41	1,90	2,17	0,15	0,21
	QMI	0,37	0,43	2,26	2,52	0,16	0,24
	SCCE	**0,30**	**0,37**	**1,36**	**1,66**	**0,14**	**0,20**

a) Algunas segmentaciones obtenidas con el algoritmo de segmentación *UCM*.

b) Imagen original c) BOEM d) CSPA e) SCCE



f) Algunas segmentaciones obtenidas con el algoritmo de segmentación *TBES*.



g) Imagen original h) EAC_AL i) QMI j) SCCE

Figura 4.3: Segmentaciones de consenso obtenidas con diferentes algoritmos.

ción Mutua Normalizada (NMI)[10], Variación de Información (VI) y el índice de Rand (RI). En la Tabla 4.3 se presenta la comparación de los resultados contra la segmentación de referencia más parecida al resultado, denotado por *Mejor GT* y además se presenta el promedio de la comparación contra todas las segmentaciones de referencia, denotado por *Todos GT*.

En la Tabla 4.3 se compara el algoritmo propuesto contra varios algoritmos de combinación de agrupamientos aplicados al problema de la combinación de segmentaciones. Estos algoritmos son BOK y BOEM [30], CSPA, HGPA y MCLA [99], EA-SL y EA-AL [37] y QMI [101]. Los resultados que se presentan en esta Tabla, son el promedio de los resultados obtenidos para cada una de las 300 imágenes estudiadas.

En la Figura 4.3 se muestran algunos ejemplos de segmentaciones y los resultados obtenidos utilizando algunos algoritmos de combinación. En esta Figura se puede ver el tipo de imágenes utilizadas y evaluar visualmente algunos de los resultados.

A pesar de que es posible aplicar algoritmos de combinación de agrupamientos a problemas de combinación de segmentaciones, el diseño de un algoritmo que utilice las particularidades de las imágenes y las segmentaciones debe proporcionar mejores resultados. Esto queda evidenciado experimentalmente en la Tabla 4.3 donde el método propuesto obtiene mejores resultados que los algoritmos de combinación de agrupamientos en todos los experimentos.

[10]En la Tabla 4.3 se presentan los valores $1 - RI$ y $1 - NMI$ para normalizar estas evaluaciones. De esta manera el RI y la NMI son presentadas como medidas de disimilitud al igual que la VI, por tanto en todos los casos, menores valores implican mejores resultados.

CONCLUSIONES

Como conclusiones del trabajo presentado en esta tesis se tiene:

- Entre los enfoques utilizados para la definición de la partición de consenso, el enfoque basado en la búsqueda de la partición mediana es el que mayores garantías teóricas brinda. La demostración de la robustez de la partición mediana corrobora la afirmación anterior.

- Una simple combinación de todas las particiones no es siempre la solución más apropiada, es posible hacer un análisis previo del conjunto de particiones a combinar y extraer información que puede ser utilizada convenientemente para mejorar los resultados del proceso de combinación. Se demostró que a partir de los índices de validación es posible determinar pesos que representen la importancia de cada partición en el proceso de combinación.

- Las funciones núcleo son de gran utilidad para el desarrollo de algoritmos de combinación de agrupamientos. A partir del uso de estas medidas se obtiene una medida global para evaluar la cercanía de cada partición a la partición de consenso. Esta medida global puede utilizarse para desarrollar heurísticas de solución para el problema de la búsqueda de la partición mediana pesada. Se demostró que pueden desarrollarse medidas de similitud *expresivas* entre particiones que cumplen la propiedad de ser funciones núcleo y que pueden calcularse de manera eficiente ($\mathcal{O}(n)$).

- El conjunto de objetos originales X y sus valores de (di)similitud es una información adicional que puede ser convenientemente utilizada para mejorar los resultados del proceso de combinación de particiones.

- Es posible definir el nivel representativo de una jerarquía de particiones de manera fundamentada a partir de la filosofía de la combinación de agrupamientos. De esta manera, es posible hallar una partición representativa utilizando más

información del problema que siguiendo el enfoque tradicional, donde la partición representativa queda determinada por el criterio, totalmente arbitrario, de un índice de validación. Esto quedó evidenciado a partir del método propuesto en la tesis.

- En problemas de segmentación de imágenes, el uso de la relación espacial de los pixeles permite desarrollar algoritmos de combinación de segmentaciones más apropiados para resolver estos problemas que los algoritmos de combinación de agrupamiento de propósito general. Esto quedó corroborado a partir del método introducido en la tesis.

- Los diferentes estudios experimentales realizados utilizando diferentes colecciones de datos ratificaron en la práctica la eficacia de los algoritmos propuestos en esta tesis.

RECOMENDACIONES

A partir de los resultados alcanzados en este trabajo se han generado un conjunto de nuevos problemas que por su importancia deben ser abordados en un futuro inmediato.

- Los resultados y algoritmos presentados en este trabajo fueron desarrollados para dar solución al problema de combinación de particiones. En la práctica aparecen problemas en lo cuales son necesarias estructuraciones solapadas o incluso difusas. Se propone extender estos resultados para estos tipos de estructuraciones.

- Es conocido que el problema de la búsqueda de la partición mediana usando la distancia de Mirkin es $\mathcal{NP}$-duro, sin embargo se pueden encontrar medidas de (di)similitud para la cual el problema puede ser resuelto en tiempo polinomial. Pero estas medidas no tienen relevancia en la práctica. Por tanto, se puede formular la siguiente pregunta: ¿Existirá una medida de (di)similitud *suficientemente expresiva* como (di)similitud entre particiones, que permita resolver el problema de la partición mediana en tiempo polinomial? Esta pregunta no ha sido estudiada en la literatura y una respuesta positiva permitiría el desarrollo de algoritmos de combinación de agrupamientos con repercusiones teóricas y prácticas.

- El algoritmo HPCK propuesto en el Capítulo 3 tiene como principal desventaja su costo computacional cuadrático. Este se debe a la utilización de una nueva representación de los objetos a partir del enfoque de representación por (di)similitudes para problemas de reconocimiento de patrones [83]. Como trabajo futuro, se propone la utilización de las técnicas de selección de prototipos [82] utilizadas para disminuir el costo computacional de los algoritmos basados en representación por (di)similitudes.

- Recientemente, se propuso en [47] la idea de la búsqueda de la partición representativa de una jerarquía, no solo en los niveles explícitos de la jerarquía, si no en un conjunto extendido de particiones. Como trabajo futuro, se propone generalizar el enfoque introducido en la tesis (PSCE) para trabajar en este conjunto extendido de particiones donde podrían alcanzarse mejores resultados.

Bibliografía

[1] Abdala, D. D., Wattuya, P., and Jiang, X. (2010). Ensemble clustering via random walker consensus strategy, in *International Conference on Pattern Recognition*, 1051 – 4651.

[2] Aizerman, M. A., Braverman, E. M., and Rozonoer, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning, *Automation and Remote Control*, 25, 821–837.

[3] Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints, *Information Retrieval*, 12, no. 4, 461–486.

[4] Analoui, M. and Sadighian, N. (2006). Solving cluster ensemble problems by correlation's matrix & ga, in *IFIP*, 228, 227–231.

[5] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2009). From contours to regions: An empirical evaluation, in *CVPR, IEEE*, 2294 – 2301.

[6] Aronszajn, N. (1950). Theory of reproducing kernels, *Transactions of the American Mathematical Society*, 68, 337 – 404.

[7] Avogadri, R. and Valentini, G. (2008). Ensemble clustering with a fuzzy approach, *Studies in Computational Intelligence*, 126, 49–69.

[8] Ayad, H. G. and Kamel, M. S. (2008). Cumulative voting consensus method for partitions with a variable number of clusters, *IEEE Trans. Pattern Anal. Mach. Intell.*, 30, no. 1, 160–173.

[9] Bakus, J., Hussin, M. F., and Kamel, M. (2002). A som-based document clustering using phrases, in *Proceedings of the 9th International Conference on Neural Information Procesing (ICONIP'02)*, 2212 – 2216.

[10] Barthélemy, J. (1976). Sur les éloignements symétriques et le principle de pareto, *Math. Sci. Hum.*, 56, 97 – 125.

[11] Barthélemy, J. and Leclerc, B. (1995). The median procedure for partition, in *Partitioning Data Sets: DIMACS Workshop*, 19, 3–34.

[12] Barthélemy, J. and Monjardet, B. (1981). The median procedure in cluster analysis and social choice theory, *Math. Soc. Sci.*, 1, 235 – 268.

[13] Ben-Hur, A., Elisseeff, A., and Guyon, I. (2002). A stability based method for discovering structure in clustered data, in *Pacific Symposium on Biocomputing*, 6 –17.

[14] Berg, C., Christensen, J. P. R., and Ressel, P. (1984). *Harmonic Analysis on Semigroups*. Springer-Verlag, New York.

[15] Bertolacci, M. and Wirth, A. (2007). Are approximation algorithms for consensus clustering worthwhile? in *Proceedings of the Seventh SIAM ICDM*, 437 – 442.

[16] Bezdek, J. (1984). Fcm: The fuzzy c-means clustering algorithm, *Computers and Geosciences*, 10, 191 – 203.

[17] Bishop, C. (1995). *Neural networks for pattern recognition*. New York, NY: Oxford University Press.

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

[19] Chang, Y., Lee, D.-J., Hong, Y., Archibald, J., and Liang, D. (2008). A robust color image quantization algorithm based on knowledge reuse of k-means clustering ensemble, *Journal of Multimedia*, 3, 20 – 27.

[20] Cheremesina, E. and Ruiz-Shulcloper, J. (1992). Cuestiones metodológicas de la aplicación de modelos matemáticos de reconocimiento de patrones en zonas del conocimiento poco formalizadas, *Revista Ciencias Matemáticas, Cuba*, 13, no. 2, 93 – 108.

[21] Cichocki, A., Morup, M., Smaragdis, P., Wang, W., and Zdunek, R. (2008). *Advances in Nonnegative Matrix and Tensor Factorization.* Computational Intelligence & Neuroscience, Hindawi Publishing Corporation.

[22] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience.

[23] de Olivera, J. V. and Pedrycz, W. (2007). *Advances in Fuzzy Clustering and Its Applications.* Wiley.

[24] Deodhar, M. and Ghosh, J. (2006). Consensus clustering for detection of overlapping clusters in microarray data, in *Data Mining Workshops. ICDM Workshops 2006*, 104–108.

[25] Dimitriadou, E., Weingessel, A., and Hornik, K. (2001). An ensemble method for clustering, in *ICANN*, 217–224.

[26] Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification, 2nd edition.* New York, NY: John Wiley & Sons.

[27] Dudoit, S. and Fridlyand, J. (2003). Bagging to improve the accuracy of a clustering procedure, *Bioinformatics*, 19, no. 9, 1090–1099.

[28] Everitt, B., Landau, S., and Leese, M. (2001). *Cluster Analysis, 4th edition.* London: Arnold.

[29] Fern, X. Z. and Brodley, C. E. (2004). Solving cluster ensemble problems by bipartite graph partitioning, in *ICML '04: Proceedings of the Twenty-First International Conference on Machine Learning*, (New York, NY, USA), ACM, 36– 44.

[30] Filkov, V. and Skiena, S. (2004). Integrating microarray data by consensus clustering, *International Journal on Artificial Intelligence Tools*, 13(4), 863 – 880.

[31] Fischer, B. and Buhmann, J. (2003). Bagging for path-based clustering, *IEEE Trans. Pattern Anal. Mach. Intell.*, 25, no. 11, 1411 – 1415.

[32] Forestier, G., Wemmert, C., and Gançarski, P. (2008). Collaborative multi-strategical clustering for object-oriented image analysis, in *Studies in Computational Intelligence*, 126, Springer Berlin / Heidelberg, 71–88.

[33] Formin, S. and Kolmogorov, A. (1999). *Elements of Theory of Functions and Functional Analysis*. Dover Publications.

[34] Fowlkes, E. B. and Mallows, J. A. (1983). A method for comparing two hierarchical clusterings, *Journal of the American Statistical Association*, 78 (383), 553 –569.

[35] Franek, L., Abdala, D. D., Vega-Pons, S., and Jiang, X. (2010). Image segmentation fusion using general ensemble clustering methods, in *Assian Conference on Computer Vision ACCV2010*. Aceptado para su publicación.

[36] Frank, A. and Asuncion, A., UCI machine learning repository (2010).

[37] Fred, A. L. and Jain, A. K. (2005). Combining multiple clustering using evidence accumulation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 835–850.

[38] Fred, A. (2001). Finding consistent clusters in data partitions in *3rd. Int. Workshop on Multiple Classifier Systems*, 309–318.

[39] Gartner, T. (2008). *Kernel for Structured Data*. Series in Machine Perception and Artifitial Intelligence, World Scientific Press.

[40] Gionis, A., Mannila, H., and Tsaparas, P. (2007). Clustering aggregation, *ACM Trans. Knowl. Discov. Data*, 1, no. 1, 341–352.

[41] Gluck, M. and Corter, J. (1985). Information, uncertainty, and the utility of categories, in *Proc. of the Seventh Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Lawrence Erlbaum, 283–287.

[42] Goder, A. and Filkov, V. (2008). Consensus clustering algorithms: Comparison and refinement, in *ALENEX* (Munro, J. I. and Wagner, D., eds.), SIAM, 109–117.

[43] González, E. and Turmo, J. (2008). Comparing non-parametric ensemble methods for document clustering, in *Natural Language and Information Systems*, 5039 of *LNCS*, 245 – 256.

[44] Gordon, A. and Vichi, M. (2001). Fuzzy partition models for fitting a set of partitions, *Psychometrika*, 66, no. 2, 229–248.

[45] Greene, D. and Cunningham, P. (2006). Efficient ensemble method for document clustering tech. rep., Department of Computer Science, Trinity College Dublin, (2006).

[46] Grotschel, M. and Wakabayashi, Y. (1989). A cutting plane algorithm for a clustering problem, *Math. Program.*, 45, 59–96.

[47] Gurrutxaga, I., Albisua, I., Arbelaitz, O., Martín, J., Muguerza, J., Pérez, J., and Perona, I. (2010). Sep/cop: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index, *Pattern Recognition*, 43, no. 10, 3364 – 3373.

[48] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques, *Intell. Inf. Syst. J.*, 17, 107–145.

[49] Handl, J., Knowles, J., and Kell, D. (2005). Computational cluster validation in post- genomic data analysis, in *Bioinformatics*, 21, 3201–3212.

[50] Haussler, D. (1999). Convolution kernels on discrete structures Tech. Rep. UCSCCRL-99-10, University of Calfornia in Santa Cruz, (1999).

[51] Hong, Y., Kwong, S., Chang, Y., and Ren, Q. (2008). Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm, *Pattern Recognition*, 41, 2742 – 2756.

[52] Hong, Y., Kwong, S., Chang, Y., and Ren, Q. (2008). Consensus unsupervised feature ranking from multiple views, *Pattern Recogn. Lett.*, 29, 595 – 602.

[53] Hu, X., Park, E., and Zhang, X. (2009). Microarray gene cluster identification and annotation through cluster ensemble and em-based informative textual summarization., *IEEE Trans Inf Technol Biomed.*, 13, no. 5, 832–840.

[54] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data.* Prentice-Hall advanced reference series, Prentice-Hall, Inc., Upper Saddle River, NJ.

[55] Jain, A. K., Murty, M., and Flynn, P. (1999). Data clustering: A review, *ACM Computing Surveys (CSUR)*, 31, no. 3, 264–323.

[56] Jiang, X. and Bunke, H. (2010). *Pattern Recognition and Machine Vision*, ch. Learning by Generalize Median Concept, 231 – 246. River Publishers.

[57] Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM Journal of Scientific Computing*, 20, 359–392.

[58] Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S. (1997). Multilevel hypergraph partitioning: application in vlsi domain, in *DAC '97: Proceedings of the 34th Annual Conference on Design Automation*, (New York, NY, USA), ACM, 526–529.

[59] Kashef, R. and Kamel, M. (2007). Cooperative partitional-divisive clustering and its application in gene expression analysis, in *7th IEEE International Conference on Bioinformatics and Bioengineering. BIBE 2007*, 116–122.

[60] Kashima, H., Tsuda, K., and Inokuchi, A. (2003). Marginalized kernels between labeled graphs, in *Proceedings of the Twentieth International Conference on Machine Learning*, AAAI Press, 321–328.

[61] Kirkpatrick, S., Gellat, C., and Vecchi, M. (1983). Optimization by simulated annealing, in *Science*, 220, 671–680.

[62] Kong, T. Y. and Rosenfeld, A. (1989). Digital topology: Introduction and survey, *Computer Vision, Graphics, and Image Processing*, 48, no. 3, 357 – 393.

[63] Krivanek, M. and Moravek, J. (1998). Hard problems in hierarchical-tree clustering, *Acta Inform.*, 3, 311 – 323.

[64] Kuhn, H. (1955). The hungarian method for the assignment problem, *Naval Research Logistic Quarterly*, 2, 83–97.

[65] Kuncheva, L. I. (2004). *Combining Pattern Classifiers. Methods and Algorithms*. New York: John Wiley & Sons.

[66] Kuncheva, L. I., Hadjitoorov, S. T., and Todorova, L. P. (2006). Experimental comparison of cluster ensemble methods, in *9th International Conference on Information Fusion*, 1–7.

[67] Lam, L. and Suen, C. Y. (1997). Application of majority voting to pattern recognition: An analysis of its behavior and performance, *IEEE Transactions on Systems, Man, and Cybernetics*, 27, no. 5, 553 – 568.

[68] Leclerc, B. (2003). The median procedure in the semilattice of orders, *Discrete Applied Mathematics*, 127, 285 – 302.

[69] Li, T., Ding, C., and Jordan, M. I. (2007). Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization, in *ICDM '07*, (Washington, DC, USA), IEEE Computer Society, 577–582.

[70] Li, Y., Yu, J., Hao, P., and Li, Z. (2007). Clustering ensembles based on normalized edges, in *PAKDD* (Zhou, Z. H., Li, H., and Yang, Q., eds.), 4426 of *LNAI*, Springer-Verlag Berlin Heidelberg, 664–671.

[71] Luo, H., Jing, F., and Xie, X. (2006). Combining multiple clusterings using information theory based genetic algorithm, in *IEEE International Conference on Computational Intelligence and Security*, 1, 84 – 89.

[72] MacQueen, J. (1967). Some methods for classification and analysis of multivariate obstervations, in *Fifth Bekerley Symposium on Math. Stat. and Prob. University of California Press*, 281 – 297.

[73] Maronna, R., Martin, D., and Yohai, V. (2006). *Robust Statistics - Theory and Methods*. Wiley.

[74] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in *ICCV*, 2, 416 – 423.

[75] Martínez-Trinidad, J., Ruiz-Shulcloper, J., and Lazo-Cortés, M. (2000). Structuralization of universes, *Fuzzy Sets and Systems*, 112, no. 3, 485 – 500.

[76] McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. New York: John Wiley & Sons.

[77] Meilă, M. (2007). Comparing clusterings–an information based distance, *Journal of Multivariate Analysis*, 98, no. 5, 873 – 895.

[78] Milligan, G. W. and Cooper, M. C. (1985). An examination of procedures for determing the number of clusters in a data set., *Psychometrika*, 50, no. 2, 159–179.

[79] Mirkin, B. (1974). The problems of approximation in space of relations and qualitative data analysis, *Automatika i Telemechanica, translated in: Information and Remote Control*, 35, no. 9, 1424 – 1431.

[80] Mirkin, B. (2001). Reinterpreting the category utility function, *Mach. Learn.*, 45, no. 2, 219–228.

[81] Mirkin, B. G. (1996). *Mathematical Classification and Clustering.* Kluwer Academic Press, Dordrecht.

[82] Pekalska, E. and Duin, R. P. W. (2002). Prototype selection for finding efficient representations of dissimilarity data, in *International Conference on Pattern Recognition*, 3, (Quebec, Canada), 37–40.

[83] Pekalska, E. and Duin, R. P. W. (2005). *The Dissimilarity Representation For Pattern Recognition. Foundations and Applications.* World Scientific.

[84] Pfitzner, D., Leibbrandt, R., and Powers, D. (2009). Characterization and evaluation of similarity measures for pairs of clusterings, *Knowl. Inf. Syst.*, 19, 361—-394.

[85] Punera, K. and Ghosh, J. (2008). Consensus-based ensembles of soft clusterings, *Applied Artificial Intelligence*, 22, no. 7&8, 780–810.

[86] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical Association*, 66, 846 – 850.

[87] Rao, S., Mobahi, H., Yang, A., Sastry, S., and Ma, Y. (2009). Natural image segmentation with adaptive texture and boundary encoding, in *ACCV*, 1, 135 – 146.

[88] Régnier, S. (1965). Sur quelques aspects mathematiques des problemes de classification automatique., *ICC Bull.*, 4, 175–191.

[89] Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure, in *Proceedings of the 2007*

Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 410 – 420.

[90] Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, 20, 53–65.

[91] Ruiz-Shulcloper, J., Sánchez-Diaz, G., and Abidi, M. (2002). Clustering in mixed incomplete data, *Heuristics and Optimization for Knowledge Discovery*, 88 – 106.

[92] Saitoh, S. (1988). *Theory of Reproducing Kernels and its Applications*. Longman Scientific & Technical, Harlow, England.

[93] Scholkopf, B., Smola, A. J., , and Miiller, K.-R. (1999). Kernel principal component analysis, in *Advances In Kernel Methods-Support Vector Learning* (Scholkopf, B., Burges, C., and Smola, A., eds.), MIT Press, Cambridge, MA, 327 – 352.

[94] Scholkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

[95] Shen, J., Lee., P., Holden, J., and Shatkay, H. (2007). Using cluster ensemble and validation to identify subtypes of paervasive developmental disorders, in *Proccedings of the AMIA Symposium, Chicago*, 666 – 670.

[96] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22, 888 – 905.

[97] Shinnou, H. and Sasaki, M. (2007). Ensemble document clustering using weighted hypergraph generated by nmf, in *ACL '07*, 77–80.

[98] Singh, V., Mukherjee, L., Peng, J., and Xu, J. (2010). Ensemble clustering using semidefinite programming with applications, *Machine Learning*, 79, 177 – 200.

[99] Strehl, A. and Ghosh, J. (2002). Cluster ensembles: a knowledge reuse framework for combining multiple partitions, *J. Mach. Learn. Res.*, 3, 583–617.

[100] Topchy, A., Jain, A. K., and Punch, W. (2004). A mixture model of clustering ensembles, in *SIAM Int. Conference on Data Mining*, 379 – 390.

[101] Topchy, A., Jain, A. K., and Punch, W. (2003). Combining multiple weak clusterings, in *ICDM '03*, 331–338.

[102] Topchy, A. P., Jain, A. K., and Punch, W. F. (2005). Clustering ensembles: Models of consensus and weak partitions, *IEEE Trans. Pattern Anal. Mach. Intell.*, 27, no. 12, 1866–1881.

[103] Tsuda, K., Kin, T., and Asai, K. (2002). Marginalized kernels for biological sequences, *Bioinformatics*, 18, 268 – 275.

[104] Tumer, K. and Agogino, A. (2008). Ensemble clustering with voting active clusters, *Pattern Recogn. Lett.*, 29, 1947– 1953.

[105] V. Singh, L. Mukerjee, J. P. J. X. (2007). Ensemble clustering using semidefinite programming, in *Advance in Neural Information Processing Systems*, 20, 1353 – 1360.

[106] Van Dongen, S. (2000). Performance criteria for graph clustering and markov cluster experiments Tech. Rep. INS-R0012, Centre for Mathematics and Computer Science, (2000).

[107] Van Rijsbergen, C. (1974). Foundation of evaluation, *Journal of Documentation*, 30(4), 365–373.

[108] van Zuylen, A. (2005). Deterministic approximation algorithm for ranking and clustering problems Tech. Rep. 1431, OIRIE, Cornell University, (2005).

[109] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, NY.

[110] Vega-Pons, S., Correa-Morris, J., and Ruiz-Shulcloper, J. (2008). Weighted cluster ensemble using a kernel consensus function, in *CIARP '08* (Kropatch, W. and Ruiz-Shulcloper, J., eds.), 5197 of *LNCS*, 195–202.

[111] Vega-Pons, S., Correa-Morris, J., and Ruiz-Shulcloper, J. (2010). Weighted partition consensus via kernels, *Pattern Recognition*, 43(8), 2712–2724.

[112] Vega-Pons, S. and Ruiz-Shulcloper, J. (2009). Clustering ensemble method for heterogeneous partitions, in *CIARP 2009* (Bayro-Corrochano, E. and Eklundh, J.-O., eds.), 5856 of *LNCS*, 481––488.

[113] Vega-Pons, S. and Ruiz-Shulcloper, J. (2010). Combinación de agrupamientos: un estado del arte Serie Azul RT_029, Centro de Aplicación de Tecnologías de Avanzada (CENATAV), (2010).

[114] Vega-Pons, S. and Ruiz-Shulcloper, J. (2010). Partition selection approach for hierarchical clustering based on clustering ensemble, in *CIARP 2010* (Bloch, I. and Cesar, R., eds.), 6419 of *LNCS*, 525 – 532.

[115] Vega-Pons, S. and Ruiz-Shulcloper, J. (2010). A survey of clustering ensemble algorithms, *International Journal of Pattern Recognition and Artifcial Intelligence*. To appear.

[116] Vega-Pons, S., Ruiz-Shulcloper, J., and Guerra, A. (2010). Weighted association based methods for the combination of heterogeneous partitions, *Pattern Recognition Letters*. Enviado a la Revista.

[117] Verma, D. and Meila, M. (2003). A comparison of spectral clustering algorithms tech. rep., University of Washington, (2003).

[118] Wakabayashi, Y. (1986). *Aggregation of Binary Relations: Algorithmic and Polyhedral Investigations*. PhD thesis, Universitat Augsburg.

[119] Wakabayashi, Y. (1998). The complexity of computing median of relations, *Resenhas IME-USP*, 3, 311–323.

[120] Wang, X., Yang, C., and Zhou, J. (2009). Clustering aggregation by probability accumulation, *Pattern Recognition*, 42, no. 5, 668–675.

[121] Wattuya, P., Rothaus, K., Prassni, J. S., and Jiang, X. (2008). A random walker based approach to combining multiple segmentations, in *ICPR 2008*, 1–4.

[122] Wattuya, P. and Jiang, X. (2008). Ensemble combination for solving the parameter selection problem in image segmentation, in *Structural, Syntactic, and Statistical Pattern Recognition*, 5342 of *LNCS*, 392–401.

[123] Wouterse, A. and Philipse, A. P. (2006). Geometrical cluster ensemble analysis of random sphere packings, *The Journal of Chemical Physics*, 125, 194709.1––194709.10.

[124] Xu, R. and Wunsch II, D. (2005). Survey of clustering algorithms, *IEEE Transactions on Neural Networks*, 16, 645–678.

[125] Xu, S., Lu, Z., and Gu, G. (2008). An efficient spectral method for document cluster ensemble, in *The 9th International Conference for Young Computer Scientists*, 808–813.

[126] Yoon, H.-S., Ahn, S.-Y., Lee, S.-H., Cho, S.-B., and Kim, J. H. (2006). Heterogeneous clustering ensemble method for combining different cluster results, in *BioDM 2006*, 3916 of *LNBI*, 82–92.

[127] Yoon, H.-S., Lee, S.-H., Cho, S.-B., and Kim, J. H. (2006). A novel framework for discovering robust cluster results, in *DS 2006*, 4265 of *LNAI*, 373–377.

[128] Yu, Z. and Wong, H. (2009). Class discovery from gene expression data based on perturbation and cluster ensemble, *IEEE Trans Nanobioscience*, 8, no. 2, 147–160.

[129] Yu, Z., Zhang, S., Wong, H.-S., and Zhang, J. (2007). Image segmentation based on cluster ensemble, in *Advances in Neural Networks*, 4493 of *LNCS*, 894–903.

[130] Zhang, H., Fritts, J. E., and Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods, *Computer Vision and Image Understanding*, 110, no. 2, 260 – 280.

[131] Zhang, X., Jiao, L., Liu, F., Bo, L., and Gong, M. (2008). Spectral clustering ensemble applied to sar image segmentation, *IEEE Transactions on Geoscience and Remote Sensing*, 46, no. 7, 2126–2136.

[132] Zhao, Y. and Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis Tech. Rep. TR 01-40, Department of Computer Science, University of Minnesota, Minneapolis, MN., (2001).

[133] Zhou, Z.-H. and Tang, W. (2006). Clusterer ensemble, *Knowledge-Based Systems*, 19, 77–83.

ANEXOS

Anexo 1: Terminología

Tabla A. 1: Terminología

CVI	Cluster Validity Index
SL	Single-Link
CL	Complete-Link
AL	Average-Link
PV	Plurally Voting
VM	Voting Merging
VAC	Voting Active Clusters
CV	Cumulative Voting
EA	Evidence Accumulation
PA	Probability Accumulation
NMI	Normalize Mutual Information
EM	Expectation Maximization
CSPA	Cluster-based Similarity Partitioning Algorithm
HPGA	Hiper Graph Partitioning Algorithm
MCLA	Meta CLustering Algorithm
BOK	Best of K Algorithm
BOM	Best One-element Move Algorithm
QMI	Quadratic Mutual Information
HCE	Heterogeneous Clustering Ensemble
NMF	Non-negative Matrix Factorization
sCSPA	Soft Cluster-based Similarity Partitioning Algorithm
sMCLA	Soft Meta CLustering Algorithm
RKHS	Reproducing Kernel Hilbert Space
WPCK	Weighted Partition Consensus via Kernels
CER	Clustering Error Rate
WEA	Weighted Evidence Accumulation
HPCK	Heterogeneous Partition Consensus via Kernels
CH	Calinski-Harabasz index
HA	Hartigan index
HL	Highest Lifetime index
PSCE	Partition Selection based on Clustering Ensemble
SCCE	Segmentation Combination based on Clustering Ensemble

Anexo 2: Comparación de los métodos de combinación de agrupamientos

Kuncheva *et al.* [66] realizaron una comparación experimental de los algoritmos de combinación de agrupamientos, pero considerando solamente diferentes versiones de métodos basados en co-asociación (ver Sección 1.2.3) y en particionamiento de (hiper)grafos (ver Sección 1.2.4). Bertolacci y Wirth [15] y Goder y Filkov [42] hicieron una comparación experimental de los métodos basados en la distancia de Mirkin (ver Sección 1.2.7). Sin embargo, estas comparaciones fueron hechas solamente desde el punto de vista experimental, donde los resultados fueron obtenidos aplicando los diferentes métodos sobre un número fijo de colecciones de datos. Además, estas comparaciones fueron hechas entre un pequeño número de algoritmos de combinación de agrupamientos con características similares.

En este anexo, se presenta una comparación de las funciones de consenso presentadas en el Capítulo 1 teniendo en cuenta 6 propiedades. Además, se incluye en este estudio las funciones de consenso basadas en funciones núcleo presentadas en este documento. El principal objetivo de esta comparación es ayudar en la selección de una función de consenso apropiada para resolver un problema dado. Se presenta el comportamiento general de los diferentes tipos de funciones de consenso, es decir, se unifican todos los métodos basados en el mismo tipo de función de consenso en una fila de la Tabla A.2. De esta manera, para cada tipo de función de consenso, se pone en la Tabla A.2 el comportamiento general con respecto a cada una de las propiedades que son analizadas. En este proceso son resaltadas en algunas casillas de la Tabla A.2 algunas excepciones que se considera que son importante tenerlas en cuenta.

Los diferentes tipos de funciones de consenso son comparados teniendo en cuenta las siguientes propiedades:

1. *Número de grupos en cada partición (NGP).* Esta propiedad expresa si los métodos pueden combinar particiones con diferentes número de grupos o no. Un método que pueda combinar particiones con un número variable de grupos puede ser utilizado en un mayor número de situaciones prácticas. Exigir que las particiones a combinar tengan el mismo número de grupos es una restricción fuerte al problema de la combinación de agrupamientos.

2. *Dependencia del mecanismo de generación (DMG).* Esta característica se refiere a si la función de consenso es dependiente o no de un mecanismo de generación

Tabla A. 2: Comparación de las funciones de consenso

	NGP (1)	DMG (2)	CCO (3)	GPC (4)	DT (5)	CC (6)
Re-etiquetamiento y Votación	Fijo (*Cumulative Voting*[8], Variable)	No	No	Si	Co-ocurrencia de objetos	Dependiente de heurística
Matriz de Co-asociación	Variable	No (*Voting-k-means*[38], Si)	No	No	Co-ocurrencia de objetos	Alta
Particiona-miento de (hiper)grafos	Variable	No	No	Si	Co-ocurrencia de objetos	Baja (*CSPA*[99], Alta)
Distancia de Mirkin	Variable	No	No	No	Partición Mediana	Dependiente de heurística
Teoría de la Informa-ción	Variable	No	No	Si	Co-ocurrencia de objetos	Baja
Modelos de Mezclas	Variable	No	No	Si	Co-ocurrencia de objetos	Baja
Algoritmos Genéticos	Variable	No	No	No	Partición Mediana	Dependiente de heurística
Métodos NMF	Variable	No	No	No	Partición Mediana	Dependiente de heurística
Métodos difusos	Variable	No	No	Si	Co-ocurrencia de objetos	Baja
Métodos núcleo	Variable	No	Si	No	Partición Mediana	Dependiente de heurística

fijo. Una función de consenso conectada a un mecanismo de generación fijo, podría hacer uso de características particulares del mecanismo de generación para mejorar los resultados del consenso. Sin embargo, si el mecanismo de generación no es apropiado para un problema en particular los resultados no van a ser los mejores. Además, una función de consenso que pueda ser aplicada independientemente del mecanismo de generación utilizado puede ser más flexible y adaptarse mejor a diversas condiciones.

3. *Considera el conjunto de objetos originales (CCO).* La mayoría de las funciones de consenso no consideran los objetos originales y solo trabajan con el conjunto de particiones. Sin embargo, los objetos originales del problema y sus valores de (di)similitud son información adicional que podría ser utilizada para mejorar los resultados de la combinación. Por otra parte, una función de consenso estrictamente dependiente de los objetos originales no podría ser aplicada en situaciones donde estos no están disponibles. Por lo tanto, una función de consenso que puede hacer uso de los objetos originales si estos están disponibles, pero que también pueda trabajar sin estos, puede ser una opción adecuada.

4. *El número de grupos en la partición de consenso es un parámetro de la función de consenso (GPC).* Una función de consenso capaz de determinar el número óptimo de grupos en la partición de consenso es preferible generalmente. Sin embargo, si en un problema particular los usuarios conocen cuántos grupos necesitan, una función de consenso donde el número de grupos pueda ser especificado debe ser más apropiada. No obstante, los algoritmos que pueden trabajar sin una previa especificación de la cantidad de grupos a formar, normalmente pueden ser transformados, de manera sencilla, para hacer uso del número de grupos como un parámetro y restringir la solución final encontrada a este parámetro. Por otra parte, los métodos que necesitan la especificación del número de grupos en la partición de consenso usualmente no pueden ser transformados de manera sencilla para trabajar independientemente de este parámetro. De esta forma, las funciones de consenso capaces de trabajar sin la especificación del número de grupos son más flexibles y fáciles de adaptar a diferentes escenarios.

5. *Definición teórica (DT).* Las funciones de consenso pueden estar basadas en dos enfoques *co-ocurrenicia de objetos* y *partición mediana*, (ver Sección 1.2). Los métodos de combinación que enfrentan el problema a través de la búsqueda de la partición mediana tienen una mejor fundamentación teórica. Sin embargo, en la práctica, muchos de estos son heurísticas para enfrentar un problema combinatorio exponencial, por tanto, la fortaleza teórica de estos métodos esta determinada por las heurísticas particulares utilizadas en cada caso.

6. *Complejidad Computacional (CC).* En el caso de la complejidad computacional, se utilizan tres valores cualitativos (*baja*, *alta* y *dependiente de la heurística*) por la siguientes razones. En cada tipo de función de consenso pueden haber diferentes métodos con diferentes complejidades computacionales. Para todos los métodos de combinación de agrupamientos, su complejidad computacional exacta no está dada en términos de las mismas variables. La complejidad computacional de algunos métodos es difícil estimarla, debido a que muchos métodos son heurísticas y no es sencillo estimar cuántos pasos serán necesarios aplicar para alcanzar una condición de parada del algoritmo. Además, los algoritmos de combinación de agrupamientos con una complejidad computacional cuadrática o superior en el número de objetos no pueden ser aplicados en grandes colecciones de datos. Por tanto, se utiliza el costo computacional cuadrático en el número de objetos como umbral para determinar si un algoritmo tiene *alta* o *baja* complejidad computacional. Se usa el valor *dependiente de la heurística*

cuando es difícil determinar la complejidad computacional ya que depende de la heurística aplicada, del problema particular a resolver o de los criterios de convergencia utilizados en cada caso.

Anexo 3: Estudio de la robustez de la partición mediana

En estadística robusta (robust statistic) [73], el concepto *punto de quiebre* (breakdown point) es una herramienta común para medir la robustez de un estimador. Intuitivamente, el *punto de quiebre* de un estimador es el porciento de objetos atípicos (por ejemplo, valores arbitrariamente grandes) que el estimador puede manejar sin que el resultado se convierta arbitrariamente grande. De esta manera, es de esperar que el máximo valor posible de *punto de quiebre* para cualquier estimador sea 0.5, ya que no tendría sentido considerar más del 50 % de las muestras como valores atípicos. Dos casos ilustrativos son el *promedio* y la *mediana* de un conjunto de números reales $\{\alpha_1, \ldots, \alpha_t\}$. El *promedio* tiene un punto de quiebre igual a 0, ya que este puede hacerse arbitrariamente grande haciendo suficientemente grande cualquiera de los valores α_i. Por otra parte, la *mediana* tiene un punto de quiebre igual a 0.5, ya que la mitad de los objetos pueden hacerse tender a infinito sin obtener un aumento considerable de la *mediana*. En este anexo se presenta un estudio sobre la robustez del concepto mediana generalizada (generalized median concept) [56], del cual la partición mediana es un caso particular.

Notación

Sea Ω un conjunto de objetos (cualquier tipo de objetos). Este puede ser un conjunto finito o infinito.

Sea $d : \Omega \times \Omega \to \mathbb{R}_+$ una distancia (métrica) definida sobre el conjunto Ω.

Sea $X \subset \Omega$ un conjunto de n objetos $X = \{x_1, x_2, \ldots, x_n\}$.

La mediana generalizada x^* del conjunto X se define como:

$$x^* = \arg\min_{x \in \Omega} \sum_{i=1}^{n} d(x, x_i) \tag{1}$$

Para cada objeto $x \in \Omega$ la suma de distancias desde él a cada objeto en X se denota como:

$$S_X(x) = \sum_{i=1}^{n} d(x, x_i)$$

Entonces, x^* es un objeto con un valor mínimo de S_X.

Además, se define el radio del conjunto X, $r(X)$ como:

$$r(X) = \max_{x_i \in X} d(x^*, x_i)$$

.

Planteamiento del problema

Sea X un conjunto de n objetos y $x^* \in \Omega$ la mediana generalizada para el conjunto X. Se añaden m ($m < n$) nuevos objetos $Y = \{y_1, y_2, \ldots, y_m\}$ al problema. Entonces, se tienen $n + m$ objetos denotados por $Z = X \cup Y$. Sea z^* la mediana generalizada del conjunto Z. Para este problema, se formula la siguiente proposición:

Proposición A. 1. *La distancia entre x^* y z^* está acotada y esta cota no depende de la posición de los nuevos objetos Y respecto a los objetos existentes X. Específicamente, se cumple:*

$$d(x^*, z^*) \leq \left(\frac{2n}{n-m}\right) \cdot r(X) \tag{2}$$

Demostración. Como z^* es la mediana generalizada de Z, $S_Z(z^*) \leq S_Z(x^*)$. Entonces,

$$\begin{aligned} d(z^*, x_1) + \ldots + d(z^*, x_n) + d(z^*, y_1) + \ldots + d(z^*, y_m) \leq d(x^*, x_1) + \ldots + d(x^*, x_n) + \\ + d(x^*, y_1) + \ldots + d(x^*, y_m) \end{aligned} \tag{3}$$

Sin embargo, como x^* es la mediana generalizada de X, $S_X(x^*) \leq S_X(z^*)$, en otras palabras:

$$d(x^*, x_1) + \ldots + d(x^*, x_n) \leq d(z^*, x_1) + \ldots + d(z^*, x_n)$$

Por tanto, con el objetivo de satisfacer la desigualdad (3), deben cumplirse las dos siguientes propiedades:

- (i) $d(z^*, y_1) + \ldots + d(z^*, y_m) \leq d(x^*, y_1) + \ldots + d(x^*, y_m)$
- (ii) $d(z^*, x_1) + \ldots + d(z^*, x_n) - (d(x^*, x_1) + \ldots + d(x^*, x_n)) \leq d(x^*, y_1) + \ldots + d(x^*, y_m) - (d(z^*, y_1) + \ldots + d(z^*, y_m))$

Agrupando los términos en la segunda desigualdad (ii), se obtiene que:

$$\begin{aligned} (d(z^*, x_1) - d(x^*, x_1)) + \ldots + (d(z^*, x_n) - d(x^*, x_n)) \leq (d(x^*, y_1) - d(z^*, y_1)) + \ldots + \\ + (d(x^*, y_m) - d(z^*, y_m)) \end{aligned} \tag{4}$$

Usando las propiedades de simetría y desigualdad triangular de d, para cada y_i se cumple que:

$$d(x^*, y_i) - d(z^*, y_i) \leq |d(x^*, y_i) - d(z^*, y_i)| \leq d(x^*, z^*)$$

Usando esta última desigualdad en cada término del miembro derecho de la inecuación (4), se obtiene:

$$(d(z^*, x_1) - d(x^*, x_1)) + \ldots + (d(z^*, x_n) - d(x^*, x_n)) \leq m \cdot d(x^*, z^*) \tag{5}$$

En el miembro izquierdo de la inecuación (5), se tienen n términos de la forma $(d(z^*, x_i) - d(x^*, x_i))$. Para que se satisfaga la desigualdad (5), al menos uno de los términos en el miembro izquierdo debe satisfacer lo siguiente:

$$(d(z^*, x_k) - d(x^*, x_k)) \leq \frac{m}{n} \cdot d(x^*, z^*) \tag{6}$$

Esto es fácil de verificar ya que si ninguno de los términos satisface esta última propiedad, no se cumple la desigualdad (5). De esta manera, se puede asumir sin pérdida de generalidad que el k-ésimo término satisface esta última propiedad (6). Ahora, usando la simetría y la desigualdad triangular de d, se obtiene:

$$d(x^*, z^*) \leq d(x^*, x_k) + d(z^*, x_k) \tag{7}$$

Despejando $d(z^*, x_k)$ en la desigualdad (6) y sustituyendo en el miembro derecho de la desigualdad (7), se obtiene:

$$d(x^*, z^*) \leq d(x^*, x_k) + d(x^*, x_k) + \frac{m}{n} \cdot d(x^*, z^*)$$

$$\frac{n-m}{n} \cdot d(x^*, z^*) \leq 2 \cdot d(x^*, x_k)$$

$$d(x^*, z^*) \leq \frac{2n}{n-m} \cdot d(x^*, x_k)$$

Finalmente, asumiendo el peor caso, i.e., x_k es el objeto a mayor distancia de x^*, en otras palabras, asumiendo que $r(X) = d(x^*, x_k)$, se tiene:

$$d(x^*, z^*) \leq \frac{2n}{n-m} \cdot r(X)$$

y la proposición queda probada. □

Esta cota superior es totalmente independiente de las posiciones de los nuevos objetos. Es decir, dado un conjunto de $n + m$ objetos con $m < n$ se pueden hacer tender a infinito m objetos que la mediana generalizada de este conjunto se mantendrá relativamente cerca. Por tanto, se puede concluir que el punto de quiebre de la mediana generalizada y por tanto de la partición mediana es 0.5. La demostración propuesta en este anexo está basada en el hecho de que la medida utilizada para definir la mediana generalizada es una distancia (métrica). No obstante, los resultados propuestos en este anexo son también válidos para la definición del problema usando una función núcleo, ya que el problema de la partición mediana con una función núcleo puede ser reducido al problema de la partición mediana usando una distancia. Esto es posible porque una función núcleo induce una distancia entre los objetos ya que la función núcleo puede ser calculada como un producto interno en un espacio de Hilbert.

Anexo 4: Índice de Rand como función núcleo

Dado un conjunto de objetos $X = \{x_1, \ldots, x_n\}$ y el conjunto de todas las posibles particiones de los objetos X, $\mathbb{P}_X$, el índice de Rand [86] $RI : \mathbb{P}_X \times \mathbb{P}_X \to \mathbb{R}$ se define entre dos particiones $P_a = \{C_1^a, \ldots, C_{d_a}^a\}$ y $P_b = \{C_1^b, \ldots, C_{d_b}^b\}$ como:

$$RI(P_a, P_b) = \frac{N_{11}^{ab} + N_{00}^{ab}}{n(n-1)/2} \tag{8}$$

donde N_{11}^{ab} es la cantidad de pares de objetos que están agrupados en el mismo grupo en P_a y en P_b, mientras que N_{00}^{ab} es la cantidad de pares de objetos que están en diferentes grupos tanto en P_a como en P_b.

Donde estas medidas pueden ser escritas como:

$$N_{11}^{ab} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_{ij}^a \cdot \delta_{ij}^b}{2} \quad \text{donde} \quad \delta_{ij}^t = \begin{cases} 1, & \text{si } (i \neq j) \wedge (\exists C \in P_t : x_i \in C \wedge x_j \in C); \\ 0, & \text{en otro caso.} \end{cases}$$

$$N_{00}^{ab} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}^a \cdot \gamma_{ij}^b}{2} \quad \text{donde} \quad \gamma_{ij}^t = \begin{cases} 1, & \text{si } \nexists C \in P_t : x_i \in C \wedge x_j \in C; \\ 0, & \text{en otro caso.} \end{cases}$$

luego el índice de Rand puede ser reescrito como:

$$RI(P_a, P_b) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_{ij}^a \cdot \delta_{ij}^b + \sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}^a \cdot \gamma_{ij}^b}{n(n-1)} \tag{9}$$

Proposición A. 2. *En índice de Rand (RI) es una función núcleo.*

Demostración. Para probar que la función RI es un núcleo es necesario probar que:

- RI es simétrica.
- $\forall t \in \mathbb{N}$, $\forall \alpha_1, \alpha_2, \ldots, \alpha_t \in \mathbb{R}$ y $\forall P_1, P_2, \ldots, P_t \in \mathbb{P}_X$ se cumple que:

$$\sum_{a=1}^{t}\sum_{b=1}^{t}\alpha_a\alpha_b RI(P_a, P_b) \geq 0$$

La simetría es evidente de la definición del índice de Rand. Para demostrar la segunda

propiedad, se sustituye en la expresión anterior la ecuación (9).

$$\sum_{a=1}^{t}\sum_{b=1}^{t}\alpha_a\alpha_b\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\delta_{ij}^{a}\delta_{ij}^{b}+\sum_{i=1}^{n}\sum_{j=1}^{n}\gamma_{ij}^{a}\gamma_{ij}^{b}}{n(n-1)}\geq 0$$

reorganizando los términos se tiene

$$\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{a=1}^{t}\sum_{b=1}^{t}\alpha_a\alpha_b(\delta_{ij}^{a}\delta_{ij}^{b}+\gamma_{ij}^{a}\gamma_{ij}^{b})\geq 0$$

$$\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\sum_{a=1}^{t}\sum_{b=1}^{t}\alpha_a\delta_{ij}^{a}\alpha_b\delta_{ij}^{b}+\sum_{a=1}^{t}\sum_{b=1}^{t}\alpha_a\gamma_{ij}^{a}\alpha_b\gamma_{ij}^{b}\right)\geq 0$$

donde nuevamente al organizar los términos se obtiene

$$\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\left(\sum_{a=1}^{t}\alpha_a\delta_{ij}^{a}\right)^2+\left(\sum_{a=1}^{t}\alpha_a\gamma_{ij}^{a}\right)^2\right)\geq 0$$

con lo cual queda demostrada la proposición. □

Anexo 5: Plataforma de experimentación para la combinación de agrupamientos

Para apoyar el estudio realizado acerca de los algoritmos de combinación de agrupamientos se desarrolló una plataforma[11] de experimentación de algoritmos de combinación de agrupamientos llamada Clustering Ensemble Suite. En esta se realizó la implementación de diversos tipos de métodos tales como: algoritmos de agrupamiento jerárquicos: Single Link, Complete Link y Average Link, así como no jerárquicos: K-Means. También se incorporaron algoritmos basados en particionamiento de grafos (METIS y HMETIS). Para la validación de estructuraciones se implementaron índices internos y externos. Además, se adicionaron diferentes algoritmos de combinación de agrupamientos basados fundamentalmente en métodos de co-asociación (Co-association, Weigthed association y Probability association), algoritmos basados en la Teoría de la Información (QMI) y basados en particionamiento de grafos e hipergrafos (CSPA, HGPA y MCLA). Estos algoritmos son capaces de trabajar con diferentes tipos de datos: numéricos, categóricos y mezclados, además de ser robustos ante la ausencia de información (missing values). La plataforma desarrollada brinda al usuario la oportunidad de seleccionar diferentes configuraciones para los algoritmos seleccionados así como la visualización de los resultados obtenidos, facilitando el análisis y comprensión de los experimentos realizados. Esta plataforma está completamente implementada en el lenguaje de programación C# 3.0, soportado por el .NET Framework 3.5 de Microsoft Corporation. El entorno visual está implementado en Windows Presentation Fundation (WPF), una tecnología también soportada por el .NET Framework 3.5. La programación de dicha plataforma es lo suficientemente flexible para admitir la incorporación de nuevos algoritmos relacionados con el tema, que permitan hacer de la misma una herramienta más completa. Una parte considerable de los experimentos presentados en esta tesis fueron realizados en esta plataforma. Actualmente, se continúa en el proceso de incorporación de nuevos algoritmos a dicha plataforma, con el fin de convertirla en una herramienta de gran utilidad para futuras investigaciones en temas relacionados con la combinación de agrupamientos.

[11] Esta plataforma computacional fue desarrollada por los licenciados en Ciencias de la Computación, Joan Sosa García y Alejandro Guerra Gandón de la facultad de Matemática y Computación de la Universidad de la Habana como parte de su trabajo de tesis de diploma bajo la tutoría del autor de este documento.

RT_044, diciembre 2011
Aprobado por el Consejo Científico CENATAV

Editor: Lic. Lucía González Bayona
Diseño de Portada: Di. Alejandro Pérez Abraham
RNPS No. 2142
ISSN 2072-6287
Indicaciones para los Autores:
Seguir la plantilla que aparece en www.cenatav.co.cu
C E N A T A V
7ma. No. 21812 e/218 y 222, Rpto. Siboney, Playa;
La Habana. Cuba. C.P. 12200
Impreso en Cuba