

Universitat Rovira i Virgili
Escola Tècnica Superior d'Enginyeria

Tesina para la obtención del título de
Màster Interuniversitari (URV-UPC-UB) en Intel·ligència Artificial

Directora: Aïda VALLS MATEU

PROTECCIÓN DE DATOS CATEGÓRICOS QUE SIGUEN PATRONES SECUENCIALES

Cristina Gómez Alonso
<crisrina.gomez@urv.cat>

Tarragona, 5 de Septiembre de 2008

Agradecimientos

Cuando se inicia un trabajo de esta envergadura, resulta imprescindible la motivación. El tema, sin dudarlo, resultó de mi interés desde el principio. El factor de que fuese un campo poco profundizado y de especial relevancia para los proyectos con los que se colaboraba, supuso una fuente de ánimo. Aunque los inicios siempre son difíciles, a lo largo del trabajo se van alcanzando los objetivos propuestos y se va esbozando el trabajo final.

Pero me gustaría aclarar que, aunque la motivación propia resulta imprescindible, igual de importante resulta la motivación externa. Por este motivo, valoro enormemente, las directrices propuestas por mi directora, sus pasos firmes antes mis pasos dubitativos, que me han orientado hacia la consecución de este trabajo. Así como sus constantes preocupaciones y su ánimo para entregarme al máximo y realizar un trabajo exitoso. Sin duda, sin ella, este esfuerzo no sería una realidad, una experiencia que me ha formado como investigadora, informática y persona. Una experiencia que recordaré como excelente. Una experiencia que me motiva a seguir investigando en un área de interés para lograr nuevos avances. Por su conocimiento, por su preocupación, por su colaboración, por su disponibilidad, por su transmisión de energía, gracias. Muchas gracias, Aïda.

Además, me gustaría agradecer la colaboración del Dr. Vicenç Torra desde el CSIC por sus orientaciones y valoraciones sobre el trabajo que se estaba realizando en la URV. Igualmente, al Dr. Shoval por facilitarnos datos de estudio para nuestro objetivo. Y, al enmarcarse este trabajo en los proyectos E-AEGIS y ARES, por brindarme la oportunidad de colaborar en ellos.

En lo personal, me gustaría destacar la ayuda incondicional de mis padres, tanto económica, para poder formarme en esta escuela, como personal, para minorar mis responsabilidades y poder dedicarme plenamente a mi formación como profesional. Gracias, por vuestra confianza y preocupación.

Por otro lado, a mi hermana, a la que a pesar de la distancia, siento

aquí conmigo y me ayuda siempre a reenfocar las situaciones y valorar los factores positivos. Gracias, Isa.

Y también, sin duda, a Ángel, que me enseña diariamente a seguir luchando por mis sueños.

A todos, muchas gracias.

Índice general

1. Introducción	9
1.1. Origen de la tesis	9
1.2. Evolución de los patrones secuenciales	10
1.3. Objetivos básicos de la tesis	11
1.4. Estructura de la documentación	12
2. Clustering	13
2.1. Definición	13
2.2. Tipos de datos	13
2.3. Función de similitud	14
2.3.1. Distancia	14
2.3.2. Coeficientes de asociación	18
2.3.3. Coeficientes angulares	18
2.3.4. Coeficientes de similitud probabilística	19
2.4. Clasificación de técnicas de clustering	19
2.4.1. Jerárquico	20
2.4.2. Particional	22
2.4.3. Otras técnicas de clustering	24
2.5. Herramientas existentes	24

3. Control de revelación estadística	27
3.1. Conceptos	27
3.2. Formatos de las bases de datos estadísticas	28
3.3. Clasificación	28
3.4. Factores en el riesgo de revelación	29
3.5. Técnicas	30
3.5.1. Perturbativas	30
3.5.2. No-Perturbativas	31
3.5.3. Métodos de Enmascaramiento vs. Tipos de Datos . . .	32
3.6. Medidas de pérdida de información	32
3.6.1. Medidas para datos continuos	33
3.6.2. Medidas para datos categóricos	33
3.7. Propiedad de k-Anonimato	34
4. Microagregación	35
4.1. Concepto	35
4.2. Clasificación	36
4.2.1. Fixed-Sized vs. Variable-Sized	37
4.2.2. Uni-Variable vs. Multi-Variable	37
4.3. Algoritmos de microagregación	37
4.3.1. Algoritmo k-Ward	37
4.3.2. Algoritmo MDAV	39
4.3.3. Otras tendencias	41
5. Agregación de datos secuenciales	43
5.1. Protección de datos secuenciales	44

5.2.	Aproximaciones de técnicas de clustering con datos secuenciales	44
5.3.	Creación de prototipos de clusters con datos secuenciales . . .	45
5.3.1.	Funciones promedio	46
5.3.2.	Prototipos de secuencias	47
6.	Nueva propuesta para la agregación de datos secuenciales	49
6.1.	Patrones identificados en datos secuenciales	49
6.2.	Función de similitud: OSS	50
6.3.	Construcción de clusters: Método KSHC	52
6.4.	Cálculo de prototipos: Método OESP	58
7.	Experimentos	61
7.1.	Aplicación software de testeo	61
7.2.	Pruebas	63
7.2.1.	Descripción de los datos	63
7.2.2.	Función OSS	64
7.2.3.	Método KSHC	68
7.2.4.	Método OESP	69
7.2.5.	Estudio de rutas turísticas	72
8.	Conclusiones	79
9.	Trabajo futuro	81

Capítulo 1

Introducción

1.1. Origen de la tesis

Actualmente la protección de la privacidad es un aspecto muy importante en nuestra sociedad porque es un derecho fundamental. Para garantizar y proteger las libertades públicas y los derechos de las personas físicas que proporcionan sus datos se necesitan desarrollar nuevas herramientas que permitan asegurar la privacidad de los individuos para que posteriormente se puedan utilizar para realizar estudios públicos. Normalmente estos datos son recogidos por las Oficinas Nacionales de Estadística, que nunca publican directamente datos que puedan revelar la identidad de las personas (como el DNI o el nombre completo), aunque, a veces se puede deducir la identidad de la persona a partir de una combinación de otros datos que sí que se publican. Por ejemplo, publicando el lugar de nacimiento, el año y el trabajo, en pueblos pequeños se podría reidentificar al individuo en cuestión, si además hemos publicado datos como los ingresos, los tipos de inversiones que realizada, etc., no se habrá preservado correctamente la privacidad de datos completamente confidenciales.

En los últimos años, el grupo iTAKA colabora con investigadores del Instituto de Investigación en Inteligencia Artificial (IIIA) y con el grupo CRISES de la URV para estudiar este tema. Concretamente, nos interesa mejorar los métodos de enmascaramiento de datos, que consisten en no publicar los datos reales (por ejemplo, el sueldo de un individuo), sino datos aproximados obtenidos a partir de métodos de microagregación de valores similares. Esta aproximación se estudia por la inferencia estadística, la cual ha lidiado en profundidad con esta cuestión para variables numéricas. No obstante, para variables categóricas se presenta un nuevo reto que intenta resolver la Inteligencia Artificial.

Actualmente, en el marco de los proyectos E-AEGIS¹ y ARES² (financiados por el Ministerio de Educación y Ciencia Español), se quiere estudiar la aplicación de métodos basados en microagregación de datos para valores no independientes, sino relacionados secuencialmente, por ejemplo, la evolución de las ventas de una empresa a lo largo de los meses.

1.2. Evolución de los patrones secuenciales

En los últimos años ha surgido un interés creciente en el desarrollo de técnicas para el análisis de secuencias de datos. Nuevos algoritmos de minería de datos temporales han sido propuestos para tratar este tipo de datos [4, 15]. La comprensión de datos secuenciales está llegando a ser muy importante y el tratamiento de estas secuencias se espera que sea empleada para nuevas aplicaciones en los próximos años[1].

El caso que nos interesa en este trabajo es el de secuencias de datos categóricos. Este tipo de datos es cada vez más habitual y su protección no ha sido apenas estudiada. Por ejemplo, las compañías de telecomunicaciones almacenan datos espacio-temporales diariamente. Estas secuencias contienen información detallada sobre el comportamiento de los individuos o del tráfico que puede permitir la detección de patrones interesantes para ser empleados en diversas aplicaciones, como por ejemplo, el control de la circulación.

De forma similar, la gente navega en Internet. Ésta es otra gran fuente de secuencias de acciones de los usuarios (por ejemplo, las páginas webs visitadas). El estudio del comportamiento en la red también lidera aplicaciones interesantes, como por ejemplo, la detección de intrusiones. Existen otros ámbitos que también producen secuencias [11]: secuencias de proteínas que describen su composición de aminoácidos y representan su estructura y función, la información genética (ADN) que codifican la genética, historiales de salud electrónicos que almacenan el historial clínico de pacientes, etc.

Sin embargo, este tipo de datos requiere una adaptación de los algoritmos aplicados a los datos estáticos. Los datos son estáticos si todas sus características no cambian en el tiempo o de forma insignificante. No obstante, el análisis de secuencias de datos se interesa en el estudio de los cambios en los valores con el fin de identificar patrones secuenciales.

¹<http://crises-deim.urv.cat/e-aegis>

²<http://crises-deim.urv.cat/ares>

En [19] se presentan tres enfoques diferentes para manejar series: (1) trabajar directamente con datos primarios, (2) convertir una serie de datos primarios en un vector de características de menor dimensión y (3) representar la secuencia con un cierto número de los parámetros del modelo. Los enfoques basados en características y modelos permiten la aplicación de algoritmos convencionales ya que su modificación no es necesaria para tratar secuencias de datos. Sin embargo, no siempre es posible construir vectores de características o modelos. En este trabajo estamos interesados en el primer enfoque, que requiere una modificación de las técnicas clásicas con el fin de poder tratar las particularidades de los datos secuenciales categóricos.

1.3. Objetivos básicos de la tesis

Esta tesina resulta especialmente interesante y motivadora por el hecho de combinar los conceptos de clustering y protección de datos (microagregación). Su originalidad radica en orientar el estudio hacia la localización de patrones de datos en series temporales que permitan su enmascaramiento, concretamente sobre variables categóricas, ya que inicialmente, la microagregación se orientó exclusivamente hacia atributos numéricos.

El objetivo general es conocer y comprender el estado actual de las técnicas de microagregación para protección de datos, en especial para el caso de secuencias de valores, y diseñar una metodología para el caso de secuencias de datos categóricos.

Este objetivo se divide en los siguientes sub-objetivos:

- Estudio de la literatura relacionada con los métodos de clustering.
- Estudiar las funciones de similitud para datos numéricos y no numéricos en los métodos de clustering.
- Estudio de la literatura relacionada con los métodos de enmascaramiento mediante microagregación.
- Diseño de un algoritmo de microagregación para datos temporales no numéricos.

1.4. Estructura de la documentación

La documentación de esta tesina se presenta dividida en:

- **Estudio de las técnicas de clustering y microagregación** (Secciones 2-4): analiza el estado actual de las técnicas de clustering, concretamente las relacionadas con el control del descubrimiento de información en bases de datos estadísticas gracias a la técnica de microagregación. La finalidad de esta fase es la introducción en el área concreta de estudio.
- **Diseño e implementación de un Algoritmo de Protección de datos secuenciales** (Secciones 5-7): introduce los conceptos y clasificación de agregación de datos secuenciales, para presentar posteriormente, la construcción de un algoritmo que permita la publicación de volúmenes de información categórica caracterizada por su continuidad secuencial, minimizando el riesgo de revelación de datos privados. El algoritmo se acompaña de ejemplos simples y de los resultados obtenidos con datos reales.
- **Conclusiones y trabajo futuro** (Secciones 8 y 9): comenta los resultados, los avances conseguidos y posibles ampliaciones en el área.

Capítulo 2

Clustering

2.1. Definición

El clustering es un método estadístico multivariante de agrupamiento automático que a partir de una tabla de datos (casos-variables) trata de posicionarlos en grupos homogéneos, conglomerados o clusters. Los clusters no son conocidos previamente, sino que son creados en función de la propia naturaleza de los datos, de manera que los individuos que puedan ser considerados más similares sean asignados a un mismo cluster, siendo a su vez lo más diferentes (disimilares) de los que se localicen en clusters distintos.

La solución del clustering no tiene por qué ser única, pero no deben encontrarse soluciones contradictorias por distintos métodos.

Ciertas complejidades en la clasificación pueden surgir con individuos que posean valores atípicos o desaparecidos.

2.2. Tipos de datos

En las técnicas de clustering se distinguen dos tipos de datos (o variables) a manipular:

- **Cuantitativos (numéricos):** representan siempre conceptos medibles o que se expresan en forma numérica. Se pueden efectuar operaciones aritméticas sobre ellos. Se dividen en:

- **Discretos:** se corresponden con números enteros.

- **Continuos:** se corresponden con números reales.
- **Cualitativos (categóricos o simbólicos):** representan características expresadas mediante adjetivos. Toman valores sobre un conjunto finito y las operaciones aritméticas sobre ellos no tienen sentido. Se dividen en:
 - **Ordinales:** aquellos en los que existe una relación de orden definido sobre el conjunto de modalidades.
 - **Nominales:** aquellos que no presentan una ordenación de sus valores. Se incluyen dentro de este grupo las variables binarias (0/1) que representan la ausencia y presencia de una cierta propiedad.

2.3. Función de similitud

Una función de disimilitud d sobre dos objetos i y j debe satisfacer las siguientes condiciones:

1. Simetría: $d(i, j) = d(j, i)$
2. Positividad: $d(i, j) \geq 0$ para todo i, j

Existen cuatro grandes medidas de similitud: **distancia**, **coeficientes de asociación**, **coeficientes angulares** y **coeficientes de similitud probabilística** [28].

Esta clasificación no resulta exclusiva debido al gran número de métricas existentes, pero sí representativa y estructural.

2.3.1. Distancia

Una función de distancia o métrica d sobre dos objetos i y j debe satisfacer las siguientes condiciones:

1. Simetría: $d(i, j) = d(j, i)$
2. Positividad: $d(i, j) \geq 0$ para todo i, j
3. Desigualdad triangular: $d(i, j) \leq d(i, k) + d(k, j)$ para todo i, j, k ; y
4. Reflexividad: $d(i, j) = 0$ si $i = j$

Además, d es una función de distancia *normalizada* si cumple $0 \leq d(i, j) \leq 1$ para todas las secuencias i y j .

Esta definición se refiere a la distancia medida entre los puntos del espacio definido por los individuos (registros). En la actualidad, existe gran cantidad de funciones de distancia que se clasifican según el tipo de variable sobre el que se desean calcular. Por tanto, se pueden distinguir:

- Distancias aplicadas a variables numéricas (o cuantitativas)
- Distancias aplicadas a variables categóricas (o cualitativas)
- Distancias mixtas (aplicadas a variables numéricas y categóricas simultáneamente)

Nuestro caso de estudio concreto se orienta hacia variables categóricas, pero debido al gran punto de referencia que son las variables numéricas, también se detallarán brevemente [28, 36]. Para más información sobre las funciones de distancia mixtas se recomienda la consulta del documento [13].

Las principales medidas de distancia para variables numéricas son las siguientes:

- **Distancia Euclídea:** raíz cuadrada positiva de la distancia anterior.

$$d(i, j) = \sqrt{\sum_{k=1}^K (x_{ik} - x_{jk})^2} \quad (2.1)$$

- **Distancia de Minkowski:** distancia genérica que se define como la raíz q -ésima de la suma de las potencias q -ésimas de las diferencias (en valor absoluto) de las coordenadas a los dos puntos considerados. Da lugar a otras distancias.

$$d_q(i, j) = \left(\sum_{k=1}^K |x_{ik} - x_{jk}|^q \right)^{\frac{1}{q}} \quad (2.2)$$

- **Distancia de City-Block o Manhattan:** caso particular de la distancia de Minkowski para $q=1$. Es la suma de las diferencias (en valor absoluto) de todas las coordenadas de los dos individuos cuya distancia se calcula.

$$d_1(i, j) = \sum_{k=1}^K |x_{ik} - x_{jk}| \quad (2.3)$$

- **Distancia Euclídea al cuadrado:** suma de los cuadrados de las diferencias de todas las coordenadas entre dos puntos. Es un caso particular de la distancia de Minkowski, para $q=2$.

$$d_2(i, j) = \sum_{k=1}^K (x_{ik} - x_{jk})^2 \quad (2.4)$$

- **Distancia de Chebychev:** caso límite de la distancia de Minkowski. Es el máximo de las diferencias absolutas de los valores de todas las coordenadas.

$$d_{\infty}(i, j) = \max_{k=1}^K (|x_{ik} - x_{jk}|) \quad (2.5)$$

- **Distancia de Canberra:** modificación de la distancia de Manhattan que es sensible a proporciones y no sólo a valores absolutos.

$$d_{CANB}(i, j) = \sum_{k=1}^K \frac{|x_{ik} - x_{jk}|}{(x_{ik} + x_{jk})} \quad (2.6)$$

Existen ciertas funciones de distancia para valores numéricos que consideran patrones secuenciales. En [19] se presentan varias aproximaciones considerando las más comunes:

- **Distancia STS (Short Time Series):** suma de los cuadrados de las diferencias de los intervalos de dos series de tiempo.

$$d_{STS}(i, j) = \sqrt{\sum_{k=1}^K \left(\frac{x_{j(k+1)} - x_{j(k)}}{t_{(k+1)} - t_{(k)}} - \frac{x_{i(k+1)} - x_{i(k)}}{t_{(k+1)} - t_{(k)}} \right)^2} \quad (2.7)$$

donde t_k es el punto temporal para los valores x_{ik} y x_{jk} .

- **Distancia DTW (Dynamic time warping):** alineación de dos series $Q = (q_1, q_2, \dots, q_n)$ y $R = (r_1, r_2, \dots, r_m)$ para minimizar su diferencia. Devuelve como resultado una matriz de $n \cdot m$ donde el elemento (i, j) de la matriz contiene la distancia $d(q_i, r_j)$ (generalmente *distancia Euclídea*). Entonces, se calcula un *warping path* (camino serpeneante) $W = w_1, w_2, \dots, w_K$, donde $\max(m, n) \leq K \leq m + n - 1$. La mínima distancia entre las dos series se calcula mediante:

$$d_{DTW}(i, j) = \min \left(\frac{\sum_{k=1}^K w_k}{K} \right) \quad (2.8)$$

Para variables categóricas las principales medidas de distancia son:

- **Distancia de Hamming generalizada:** se define sobre cadenas, de forma que se suman del número de posiciones con valores discordes. Se considera distancia 0 cuando existe igualdad y 1 cuando no.

$$d_H(i, j) = \sum_{k=1}^K d_k(i, j) \quad (2.9)$$

- **Distancia de Chi-squared** (χ^2): transformación de los datos categóricos en forma disjuntiva completa (conjunto de variables binarias). Se basa en el número de objetos de un conjunto que tienen el mismo valor que el objeto i para la k -th variable, I_{ki} .

$$\chi^2(i, j) = \sum_{k=1}^K d_k(i, j) \quad (2.10)$$

donde $d_k(i, j)$ es 0 cuando $x_{ik} = x_{jk}$ y $\left(\frac{1}{I_{ki}} + \frac{1}{I_{kj}}\right)$ en otro caso.

En el caso de secuencias de valores categóricos, existen tres enfoques: uso de la distancia de Hamming, distancias de cadenas y distancias basadas en la alineación.

Con respecto a las distancias basadas en cadenas de caracteres destacan [11]:

- **Distancia Edit o Levenshtein**: cálculo del mínimo número de operaciones de edición para transformar S_1 en S_2 , donde una operación de edición es una inserción, supresión o sustitución de un carácter individual.
- **Distancia Damerau-Levenshtein**: adición a la distancia Edit de la operación de transposición, donde se intercambian dos elementos de una secuencia.
- **Divergencia Kullback-Liebler**: mide la diferencia entre dos distribuciones de probabilidad:

$$d_{KL}(i, j) = \sum_{k=1}^K (P_i(x|X) - P_j(x|X)) \log \left(\frac{P_i(x|X)}{P_j(x|X)} \right) \quad (2.11)$$

donde P_i es la *probabilidad condicional* de la distribución para S_i

Por otro lado, otra aproximación está basada en la alineación de secuencias. Proviene de estudios de las estructuras de las moléculas de ADN, RNA o proteínas. Su principal característica es que los elementos de las secuencias son caracteres. Estos métodos están basados en la distancia DTW (Dynamic Time Warping, eq. 2.8). Un análisis más detallado se puede encontrar en [22, 25, 35].

Al margen de las funciones de distancia, para facilitar los cálculos necesarios para alinear las secuencias, se han desarrollado algunas aplicaciones como FASTA [34] o BLAST [23], que están orientados a secuencias de nucleótidos o aminoácidos.

2.3.2. Coeficientes de asociación

Se utilizan con datos cualitativos, aunque también se podrían utilizar con datos cuantitativos si se sacrifica cierta información proporcionada. Miden la concordancia o conformidad entre los dos registros de datos. Son los más adecuados para variables binarias.

Variable 1 \ Variable 2	Presencia	Ausencia
	Presencia	Ausencia
Presencia	a	b
Ausencia	c	d

Cuadro 2.1: Relación coeficientes de asociación

Conforme al Cuadro 2.3.2 se han definido las siguientes métricas:

- **Coeficiente de Jaccard-Sneath:** cociente entre el número de emparejamientos positivos y la suma de emparejamientos positivos y negativos.

$$S_j = \frac{a}{(a + b + c)} \quad (2.12)$$

- **Coeficiente de emparejamiento simple:** cociente entre el número de emparejamientos y el número total de casos considerados.

$$S_{SM} = \frac{(a + d)}{(a + b + c + d)} \quad (2.13)$$

- **Coeficiente de Yule:** cociente entre la diferencia de presencias y ausencias conjuntas y la suma de coincidencias conjuntas.

$$S_Y = \frac{(ad - bc)}{(ad + bc)} \quad (2.14)$$

2.3.3. Coeficientes angulares

Miden la proporcionalidad e independencia entre los vectores que definen los registros. Los valores próximos a 0 indican disimilitud entre los individuos y los valores próximos a +1 o -1 indican similitud positiva o negativa respectivamente.

Destaca el **coeficiente de correlación**:

$$r_{i,j} = \frac{\sum_{k=1}^K (x_{ik} - X_i)(x_{jk} - X_j)}{\left(\sum_{k=1}^K (x_{ik} - X_i)^2 \sum_{k=1}^K (x_{jk} - X_j)^2\right)^{\frac{1}{2}}} \quad (2.15)$$

2.3.4. Coeficientes de similitud probabilística

Calculan la homogeneidad del sistema por particiones o subparticiones del conjunto de los registros e incluye información estadística.

2.4. Clasificación de técnicas de clustering

Los métodos de análisis de clusters han sido estudiados desde hace muchos años [12, 16, 36]. Existen diferentes métodos según las diversas formas de llevar a cabo la agrupación de los individuos o grupos de individuos. Una posible clasificación de los métodos es la siguiente:

- **Métodos Aglomerativos-Divisivos:**

- *Aglomerativo*: parte de tantos grupos como individuos y sucesivamente fusiona los más similares.
- *Divisivo*: parte de un único grupo formado por todos los individuos y en cada etapa efectúa divisiones del conjunto.

- **Métodos Jerárquicos-Particionales:**

- *Jerárquico*: consiste en una secuencia de $g + 1$ clusters $(G_0 \dots G_n)$ en la que G_n es la partición disjunta de todos los individuos y G_g es el conjunto partición. El número de partes de cada una de las particiones disminuye progresivamente, lo que hace que éstas sean cada vez más amplias y menos homogéneas.
- *Particional (o no jerárquico)*: construye grupos homogéneos sin establecer relaciones jerárquicas o de orden entre dichos grupos.

- **Métodos Solapados-Exclusivos:**

- *Solapado*: admite que un individuo puede pertenecer a dos grupos simultáneamente.
- *Exclusivo*: no admite que ningún individuo pueda pertenecer simultáneamente a dos grupos en la misma etapa.

- **Métodos Secuenciales-Simultáneos:**
 - *Secuencial*: aplica el mismo algoritmo recursivamente a cada grupo.
 - *Simultáneo*: efectúa la segmentación mediante una operación simple y no reiterada.
- **Métodos Monotéticos-Politéticos:**
 - *Monotético*: clasifica los objetos en base a una característica única.
 - *Politético*: clasifica los objetos en base a varias características suficientes (mas sin exigir que todos los objetos las posean).
- **Métodos Directos-Iterativos:**
 - *Directo*: realiza una única asignación de los individuos a los grupos.
 - *Iterativo*: corrige las asignaciones de los individuos a los grupos para conseguir la clasificación óptima en varias iteraciones.
- **Métodos Ponderados-No ponderados:**
 - *Ponderado*: atribuye diferentes pesos a las características de los individuos a clasificar según su importancia.
 - *No ponderado*: establece el mismo peso a todas las características.
- **Métodos Adaptativos-No adaptativos:**
 - *Adaptativo*: aprende durante el proceso de formación de los grupos y modifican su criterio de optimización o medida de similitud.
 - *No adaptativo*: es fijo y predeterminado.

En términos generales, se puede decir que la principal característica que se usa para distinguir entre métodos de clustering es la relación de jerarquía. En el siguiente apartado se presentan los métodos clasificados con este criterio.

2.4.1. Jerárquico

Aglomerativo Este método construye la jerarquía tomando elementos individuales y fusionándolos progresivamente según la medida de similitud. Este tipo de método de clustering es el más empleado, cumpliendo las propiedades de secuencialidad y exclusividad (también llamado SAHN (Sequential, Agglomerative, Hierarchic and Nonoverlapping)).

Según como se calcule la similitud de enlace entre clusters se pueden distinguir los siguientes métodos:

- **Single Linkage Method:** método del mínimo o vecino más cercano.
- **Complete Linkage Method:** método del máximo o distancia máxima.
- **Average Linkage Method:** método de la media o distancia promedio. Ponderado o no ponderado.
- **Centroid Method:** método del centroide o distancia prototipo.
- **Ward's Method:** método de la mínima varianza.

El algoritmo genérico para los métodos SAHN es:

1. Considerar cada elemento (o registro) representante de un cluster que solamente contiene dicho elemento.
2. Calcular las distancias entre todos los clusters existentes dos a dos.
3. Elegir los cluster cuya distancia sea menor.
4. Mezclar los clusters elegidos en el paso anterior según la medida de similitud.
5. Si existe más de un cluster, volver al paso 2.

Divisivo Este método construye la jerarquía tomando el conjunto de elementos y separándolos en grupos progresivamente según la medida de similitud.

La representación gráfica básica de las técnicas de clustering jerárquicas son los **dendogramas**. Estos gráficos muestran la formación de grupos jerárquicos a modo de árbol invertido, así como la distancia entre los clusters.

Aunque el algoritmo de clustering finaliza cuando todos los elementos se encuentran integrados en un mismo cluster, el dendograma permite conocer visualmente la composición de los clusters en etapas intermedias. Observando la Figura 2.1 la línea horizontal gruesa superior realiza una división del conjunto en 5 clusters (a nivel 2.5), mientras que la línea gruesa inferior la realiza en 8 clusters (a nivel 2).

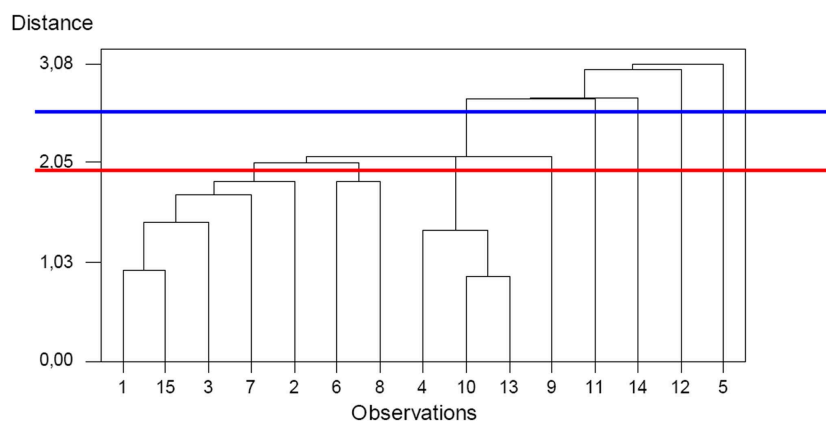


Figura 2.1: Gráfico dendograma

2.4.2. Particional

Las técnicas de clustering particionales no identifican la existencia de una estructura vertical de dependencia entre los grupos formados. En este análisis se precisa determinar previamente el número de clusters en que se desea agrupar los datos. Esta exigencia supone la necesidad de repetición de las pruebas a fin de tantear la clasificación que mejor se ajuste al objetivo del problema o sea más clara de interpretación. Existen cuatro grandes familias de métodos particionales:

- **Reasignación:** ubica a los individuos en su grupo más adecuado tras repetidas iteraciones. Finaliza cuando no se detectan reasignaciones que optimicen el resultado.
- **Búsqueda de la densidad:**
 - *Aproximación tipológica:* localiza las zonas con mayores concentraciones de individuos.
 - *Aproximación probabilística:* localiza los individuos que pertenecen a la misma distribución.
- **Directo:** clasifica simultáneamente a individuos y variables.
- **Reducción de dimensiones:** busca factores en el espacio de individuos. Cada factor se corresponde a un grupo.

El cálculo de clusters en técnicas no jerárquicas se basa en el **criterio de varianza**. Este criterio orienta la identificación de la clasificación óptima hacia la minimización de la dispersión dentro de cada grupo (suma de varianzas mínima).

Los dos métodos particionales más utilizados son:

K-Means

Este método es el más importante de las técnicas de clustering particionales.

Consta de las siguientes etapas:

(Previa: Determinar el número de clusters k)

- Aleatoriamente se generan k clusters y se determinan sus centroides (o directamente se generan k puntos aleatorios que se consideran centroides).
- Se asigna a cada punto el centroide más próximo.
- Se recalculan los nuevos centroides para cada cluster.
- Se repiten los dos pasos anteriores hasta un criterio de convergencia (Generalmente que la asignación no se haya modificado).

Las principales ventajas de este algoritmo son su simplicidad y velocidad, que permiten su ejecución sobre grandes conjuntos de datos. Por el contrario, debido a su componente de aleatoriedad inicial, no puede garantizar la obtención del mismo resultado en todas las ejecuciones, ni la mínima varianza global (aunque sí la mínima varianza dentro de cada cluster).

K Vecinos más próximos

Este algoritmo es una variante del método vecino más próximo, donde a cada punto se le asigna la clase mayoritaria entre los k vecinos más próximos (kNN , k-Nearest Neighbors). Por lo que el valor de k será clave para el resultado final del algoritmo. Otra opción, más común para solventar este problema, es determinar una región de cercanía. Las etapas de este método para la clasificación de un nuevo punto son:

(Previa: Determinar la distancia máxima r de vecindad (cercanía))

- Dado el punto x , generar el conjunto x' de ejemplos que están a una distancia menor r de x .
- Se calculan todos los $k_h(x')$, es decir, el número de ejemplos en x' que pertenecen a la clase A_h .
- Sea $k_g(x')$ el de mayor valor de todos los $k_h(x')$: $k_g = \max_{h=1\dots q}(k_h)$

- Se le asigna a x la clase A_g .

Su principal ventaja es que permite el empleo de funciones de distancias más complejas, ya que su cálculo sólo afecta a la selección de los ejemplos, no a la ubicación de los centroides.

2.4.3. Otras técnicas de clustering

Al margen de las técnicas de jerárquicas y partitivas, existen otras técnicas destacables:

- **Métodos basados en funciones de densidad:** la unión de los elementos próximos viene determinada por una medida de densidad local.
- **Métodos basados en modelos:** cada cluster es modelado (por ejemplo, mediante una función simple de distribución) y se determinan los datos que mejor se ajustan a cada modelo.
- **Métodos basados en redes neuronales:** donde neuronas activas refuerzan su vecindad en ciertas regiones y reducen la actividad de otras neuronas. Ejemplos: *SOFMs* (Self-Organized Feature Maps, o Mapas Auto-Organizativos de Características) y *ART* (Adaptive Resonance Theory, o Teoría de Resonancia Adaptativa).
- **Métodos basados en teoría de grafos:** donde los nodos del grafo se corresponden con los elementos del cluster y las aristas con las proximidades entre cada par de elementos. Se pueden aplicar tanto a grafos jerárquicos como particionales.

Para más información sobre éstas y otras técnicas consultar el documento [36].

2.5. Herramientas existentes

Existen gran cantidad de herramientas software para la minería de datos que también cubren métodos de análisis de clustering. Un ejemplo de las más destacables serían las siguientes:

■ Privadas:

- Enterprise Miner de SAS Institute
<http://www.sas.com/technologies/analytics/datamining/miner/>
- Clementine de SPSS
<http://www.spss.com/la/productos/clementine/clementine.htm>
- Intelligent Miner de IBM
<http://www-306.ibm.com/software/data/iminer/>

■ De libre distribución:

- WEKA (Waikato Environment for Knowledge Analysis)
<http://www.cs.waikato.ac.nz/ml/weka/>
- RapidMiner
<http://rapid-i.com/>
- YALE
Yet Another Learning Environment)<http://rapid-i.com/>
- Rattle de R
<http://rattle.togaware.com/>
- Pentaho
<http://www.pentaho.com/>

Capítulo 3

Control de revelación estadística

El control de revelación estadística (*Statistical Disclosure Control* o *SDC*), también llamado control de inferencia, tiene como objetivo el intercambio y difusión de datos preservando la confidencialidad. Para ello, las técnicas de control de revelación estadística transforman la base de datos original en una nueva base de datos, teniendo en cuenta que los datos protegidos satisfagan simultáneamente la condición de utilidad y de seguridad. Los datos serán útiles si son representativos del conjunto inicial y seguros si no permiten la re-identificación de los datos originales.

3.1. Conceptos

La confidencialidad de la información se garantiza al minimizar su riesgo de revelación. Ambos conceptos, confidencialidad y revelación, se definen formalmente en [31] de la siguiente forma:

Confidencialidad (*Confidentiality*) tratamiento de la información que un individuo ha facilitado en una relación de confianza y con la condición de que no será divulgada a terceros sin su consentimiento siendo congruente con la revelación original. La confidencialidad se diferencia de la privacidad porque se aplica tanto a datos de organizaciones y empresas como de individuos, mientras que la privacidad es un derecho individual.

Revelación (*Disclosure*) atribución inapropiada de la información a un sujeto, siendo este propiamente un individuo o una organización.

3.2. Formatos de las bases de datos estadísticas

Las bases de datos estadísticas son aquellas que contienen información estadística. Se dividen en los siguientes formatos:

- **Datos tabulares:** tablas de frecuencia o magnitudes. Son el *output* clásico de las estadísticas oficiales.
- **Bases de datos de consulta:** bases de datos *online* a las cuales el usuario hacer solicitar consultas estadísticas (sumas, medias ...).
- **Microdatos:** ficheros donde cada registro se corresponde con información de un sujeto (ciudadano o empresa).

A lo largo de este trabajo, se hará referencia exclusivamente a bases de datos de microdatos y a sus métodos concretos de protección y enmascaramiento, ya que su riesgo de descubrimiento es superior al de las dos primeras. Los datos tabulares publican información estática agregada y su objetivo es no contener información confidencial que pueda ser inferida. Las bases de datos de consulta también deben evitar que las consultas sucesivas de información puedan inferir información específica. Pero son los microdatos, al referirse a información individual los que implican un mayor riesgo de descubrimiento de los sujetos originales, a la vez que la información que publican ha de ser analíticamente útil.

3.3. Clasificación

Existen tres tipos de posibles riesgos de revelación:

- **Revelación de identidad** (*Identity disclosure*): sucede cuando se puede identificar al sujeto (o registro) de unos datos publicados.
- **Revelación de atributos** (*Attribute disclosure*): sucede cuando información confidencial sobre un sujeto es revelada o estimada y puede ser atribuida unívocamente.
- **Revelación de inferencia** (*Inferential disclosure*): sucede cuando información individual puede ser inferida con un grado alto de confianza mediante propiedades estadísticas de datos publicados. Generalmente, son predictores de valores individuales de baja calidad.

3.4. Factores en el riesgo de revelación

El riesgo de revelación puede verse aumentado o reducido según factores que afecten el fichero de microdatos original.

Factores contribuyentes

Existen dos fuentes principales de riesgo de revelación en un fichero de microdatos:

- **Existencia de campos de alto riesgo:**
 - Ciertos registros de un fichero pueden representar sujetos con características únicas que los identifican unívocamente, por ejemplo, trabajos poco comunes (actor, juez) o ingresos muy elevados (alrededor de un millón de euros).
 - Múltiples registros en un fichero pueden ser conocidos por pertenecer a un mismo cluster, por ejemplo, familia o colegio.
 - Una dimensión de los datos es publicada en un nivel de detalle demasiado fino, por ejemplo, si se publica el código postal.
- **Posibilidad de concordancia de un fichero de microdatos con ficheros externos:** existen ciertos individuos o empresas que poseen una única combinación de sus campos. Intrusos podrían utilizar ficheros externos con los mismos campos e identificadores para vincular los sujetos únicos con sus registros del fichero de microdatos origen.

Factores reductores

Existen diversas circunstancias que afectan positivamente en la prevención de la revelación:

- **Edad de los datos del fichero de microdatos.** Las características de los individuos y empresas pueden cambiar considerablemente a lo largo del tiempo. La edad de los ficheros externos con los que se intenta vincular el fichero original posiblemente no coincidirá con el original.
- **Ruido en la información** del fichero de microdatos y de los ficheros externos.
- **Diferente definición de variables** del fichero de microdatos y de los ficheros externos.
- **Otros factores:** tiempo, esfuerzo y recursos económicos.

3.5. Técnicas

Los métodos de control del riesgo de revelación para microdatos son conocidos como Métodos de Enmascaramiento (en inglés, *Masking Methods*) [9, 33]. Su idea es variar los datos originales para conseguir evitar los riesgos antes mencionados.

En base a sus principios operacionales, estos métodos se presentan divididos en dos categorías: **perturbativas** y **no-perturbativas**.

3.5.1. Perturbativas

El conjunto de microdatos se distorsiona antes de su publicación. Pueden incluirse nuevos datos, suprimir y/o modificar los existentes beneficiando la confidencialidad estadística.

Las principales técnicas perturbativas son:

- **Ruido aditivo** (*Additive noise*): añade ruido con la misma estructura de correlación que los datos originales. Adecuado para datos numéricos.
- **Distorsión de los datos mediante distribuciones de probabilidad** (*Data distortion by probability distribution*): distorsiona los datos con series estimadas en la función de densidad de las variables.
- **Microagregación** (*Microaggregation*): construye pequeños microclusters de datos muy similares, después sustituye los datos originales por los valores de la media del grupo (prototipo). El tamaño de los conjuntos (o clusters) debe ser igual o superior a una variable k para garantizar la no revelación.
- **Re-muestreo** (*Re-sampling*): Se toman t muestras independientes $X_1 \dots X_t$ del mismo tamaño n de la variable original V_i . Se ordenan los datos dentro de cada muestra. Se hace la media de los primeros valores de cada muestra, y estos se sustituyen por esta media. Se hace lo mismo con los $n - 1$ valores de las siguientes posiciones.
- **Pérdida de compresión** (*Lossy compresion*): considera el archivo de datos como una imagen y aplica algoritmos de compresión (por ejemplo JPEG).
- **Imputación múltiple** (*Multiple imputation*): Se genera una nueva version con datos simulados creados a partir de múltiples técnicas de

imputación a partir de los datos originales. Por ejemplo, un método de imputación consiste en hacer regresiones con una distribución aleatoria del error, para imputar valores «desconocidos» a una variable continua.

- **Camuflaje** (*Camouflage*): camufla la información original en un intervalo (conjunto infinito). Es un método adecuado para datos numéricos. Supone una pérdida de información elevada.
- **PRAM** (*Post-Randomization Method*): sustituye los datos originales por otra información diferente de acuerdo a mecanismo de probabilidad, siguiendo los modelos de cadenas de Markov. Fusiona adición de ruido, supresión de datos y recodificación.
- **Intercambio ordenado** (*Rank swapping*): intercambia aleatoriamente un porcentaje de los datos originales.
- **Redondeo** (*Rounding*): sustituye los valores originales con valores redondeados. Se elige valores pertenecientes a un conjunto predefinido de redondeo. Adecuado para variables numéricas.

3.5.2. No-Perturbativas

Estas técnicas no alteran los datos, sino que producen supresiones parciales o reducciones del nivel de detalle del conjunto original.

Los principales métodos no-perturbativos son:

- **Muestreado** (*Sampling*): publica una muestra de los datos originales. Adecuado para variables categóricas. En variables continuas, riesgo de revelación elevado.
- **Recodificación global** (*Global recoding*): combina categorías para formar nuevas categorías menos específicas. En variables continuas, discretiza valores (paso de infinito a finito).
- **Codificación superior e inferior** (*Top and bottom coding*): determina un threshold superior e inferior y forma nuevas categorías con los valores extremos. Caso concreto de 'Recodificación global'.
- **Supresión local** (*Local suppression*): elimina ciertos valores con el objetivo de incrementar el conjunto de registros concordantes. Orientado a variables categóricas.

3.5.3. Métodos de Enmascaramiento vs. Tipos de Datos

A continuación se muestra una tabla comparativa de los diferentes métodos de enmascaramiento con respecto a los diferentes datos sobre los que se pueden aplicar:

Método	Tipo	Datos continuos	Datos categóricos
Ruido aditivo	P	X	
Distorsión probabilidad	P	X	X
Microagregación	P	X	X
Re-muestreado	P	X	
Pérdida de compresión	P	X	
Imputación múltiple	P	X	
Camuflaje	P	X	
PRAM	P		X
Intercambio ordenado	P	X	X
Redondeo	P	X	
Muestreado	NP		X
Recodificación global	NP	X	X
Codificación sup/inf	NP	X	X
Supresión local	NP		X

Notación: P: Perturbativa; NP: No-Perturbativa

Cuadro 3.1: Relación Métodos de Enmascaramiento vs. Tipos de Datos

Como se puede observar, sobre los datos categóricos solamente se pueden aplicar un conjunto reducido de técnicas.

3.6. Medidas de pérdida de información

Para poder preservar la información contenida en un fichero de microdatos, resulta imprescindible un cierto índice de pérdida de los datos originales. En [33], se determinan dos propiedades para garantizar el detrimento de información:

- Un fichero de microdatos protegido es *analíticamente válido* si las siguientes condiciones son aproximadamente preservadas con respecto al fichero original (algunas condiciones son exclusivamente aplicables a atributos continuos):
 - Medias y covarianzas sobre un conjunto reducido de subdominios

(subconjunto de registros y/o variables).

- Valores marginales para ciertas tabulaciones de los datos.
 - Al menos una característica de la distribución.
- Un fichero de microdatos es *analíticamente interesante* si se proporcionan seis variables de importantes subdominios que puedan ser validadas analíticamente.

3.6.1. Medidas para datos continuos

Si se desea medir la pérdida de información para datos continuos, se consideran los ficheros de microdatos original y modificado (protegido) como matrices X y X' donde las filas son los registros y las columnas atributos y se comparan las diferencias entre la información en X y X' . Para caracterizar la información contenida en los ficheros se pueden realizar los siguientes cálculos:

- Matrices de covarianza
- Matrices de correlación
- Vectores de valores comunes
- Matrices de coeficientes de factores resultado

Dados los resultados de alguno de los métodos anteriores, la discrepancia entre matrices se suele calcular mediante:

- Error medio al cuadrado
- Error medio absoluto
- Variación media (independiente de cambios de escala de las variables)

3.6.2. Medidas para datos categóricos

Se consideran las siguientes alternativas para la medición de la pérdida de información sobre datos categóricos:

- Comparación directa de valores categóricos (tomando una medida de similitud)

- Comparación de tablas de contingencia
- Medidas basadas en entropía (si se considera el proceso de enmascaramiento como una adición de ruido a los datos originales)

3.7. Propiedad de k-Anonimato

Para poder definir el concepto de k-Anonimato (o k-Anonymity, en inglés), previamente es necesario conocer la clasificación de los tipos de atributos en un conjunto de microdatos [10, 5]:

- **Identificadores:** identifican al sujeto de forma unívoca.
- **Quasi-Identificadores:** combinación de atributos que pueden ser vinculados con información externa para reidentificar a algunos sujetos a los que se refieren registros del fichero.
- **Atributos confidenciales:** contienen información comprometida del sujeto.
- **Atributos no confidenciales:** no contienen información comprometida del sujeto.

Identificados los diferentes tipos de atributos que pueden estar contenidos en un fichero de microdatos, la k-Anonimato intenta establecer el equilibrio entre la pérdida de información y el riesgo de revelación mediante la siguiente condición:

*Un conjunto se dice que satisface la **k-Anonimato** para $k > 1$ si, para cada combinación de valores de quasi-identificadores, existen al menos k registros en el conjunto que comparten la combinación.*

En [5] se define una nueva propiedad que refina la k-Anonimato:

*Un conjunto de datos se dice que satisface la **p-Sensitive-k-Anonimato** si para $k > 1$ y $p \leq k$ si satisface la k-Anonimato y, para cada grupo de tuplas con la misma combinación de valores de atributos clave que existen en el conjunto, el número de valores distintos para cada atributo confidencial es al menos p dentro del mismo grupo.*

Capítulo 4

Microagregación

El concepto de microagregación en SDC toma como base la técnica de clustering, pero con la de necesidad de establecer un número mínimo de elementos para cada uno de los grupos (clusters) creados, de forma que la aplicación de la microagregación pueda garantizar la protección de los datos a publicar. Por otro lado, los conjuntos deben contener elementos muy similares para que su fusión sea útil posteriormente (es decir, representativa del conjunto inicial), lo que implica que, recomendablemente su tamaño debe ser reducido.

4.1. Concepto

La **microagregación** es una técnica de control de revelación estadística para microdatos ¹ que sigue una tendencia perturbativa/sustitutiva. Su procedimiento general consiste en:

- Construcción de clusters con los datos originales según un criterio de máxima similitud (donde cada cluster debe tener al menos k elementos).
- Construcción de un prototipo representativo de cada cluster.
- Sustitución de los registros originales por sus correspondientes prototipos.

¹Ficheros donde cada registro contiene información sobre un individuo (ciudadano o compañía)

Por tanto, para cualquier tipo de dato, la microagregación puede ser definida en términos de dos operaciones: **partición** del conjunto de datos original en clusters y una **agregación** de todos los registros de un cluster, que son sustituidos por su prototipo.

En [10] se afirma la adecuación de la microagregación para salvaguardar la k-Anonymity resultando más adecuada que otros métodos perturbativos de protección.

La **microagregación óptima** consiste en encontrar una k partición $P = \{G_1, \dots, G_g\}$ tal que la suma de los cuadrados de las distancias euclídeas (*SSE*: Sum of Squared Euclidean distances) para cada objeto x_{ij} a los centroides sea minimizada (donde G_i es el grupo al cual x_{ij} pertenece).

$$SSE(P) = \sum_{i=1}^g \sum_{j=1}^{|G_i|} (x_{ij} - c(G_i))'(x_{ij} - c(G_i)) \quad (4.1)$$

donde:

- g es el número de grupos (clusters) existentes
- $|G_i|$ es la cardinalidad del grupo i
- x_{ij} es el registro j del grupo i
- $c(G_i)$ es el centroide del grupo i

Según [26], el problema de la microagregación es *NP* (es decir, no puede ser resuelto en tiempo polinomial), por lo que todas las propuestas actuales de solución son de naturaleza heurística.

4.2. Clasificación

En [20] se presentan dos tendencias de clasificación de los métodos de microagregación en base a:

- Tamaño de los clusters (número de elementos contenidos): fixed-sized o variable-sized.
- Número de variables a considerar para establecer los criterios de similitud entre los elementos: uni-variable o multi-variable.

4.2.1. Fixed-Sized vs. Variable-Sized

Inicialmente, en el método de la microagregación clásica se propuso la definición de grupos de tamaño k o **fixed-sized** para la ocultación de los datos.

En 1999, en [7, 20] se demuestra la mejora en la agregación determinando el tamaño de los clusters igual o mayor que k , pero inferior a $2k - 1$. Esta tendencia recibe el nombre de **variable-sized** o **data-oriented micro-aggregation** y consigue una agrupación más homogénea de los datos y, por tanto, una pérdida de información menor.

4.2.2. Uni-Variable vs. Multi-Variable

Los métodos **uni-variables** (o *individual ranking* o *blurring*) manejan conjuntos de datos multi-variables por microagregación mediante una variable a cada paso, es decir, las variables son tratadas de forma secuencial e independientemente microagregadas. Consiguen ratios muy bajos de pérdida de información, pero su riesgo de revelación es elevado.

Los métodos **multi-variables** proyectan los datos de varias variables sobre un mismo eje o utilizan directamente los datos sin proyectar. Estas técnicas son más complejas, pero incrementan el control de revelación.

4.3. Algoritmos de microagregación

Los primeros algoritmos para microagregación fueron el **k-Ward** y el **MDAV**. El primero se orientó inicialmente hacia la tendencia uni-variable, pero modificando la consideración del criterio de similitud, puede también ser aplicado a cálculos multi-variable. El segundo, MDAV, es exclusivamente multi-variable, de tamaño fijo (aunque se ha publicado una mejora que permite que sea de tamaño variable). A continuación se presentan estos algoritmos.

4.3.1. Algoritmo k-Ward

El algoritmo de Ward es una técnica de clustering jerárquico aglomerativo cuyo objetivo es minimizar la pérdida de información dentro de cada cluster y cuantificar dicha pérdida para que pueda ser interpretable. En cada iteración del algoritmo, se considera la posible fusión de todos los pares de

grupos posibles y se escogen aquellos elementos cuyo incremento de pérdida de información en su fusión es mínimo. Esta pérdida se define en base a la suma de cuadrados mínimos (SSE, Sum of Squared Errors) dentro de cada cluster.

Inicialmente, cuando todos los elementos son considerados individualmente, $SSE = 0$. La distancia $d(i, j)$ entre dos datos univariados i e j es:

$$d(i, j) = \left(x - \frac{x+y}{2}\right)^2 + \left(y - \frac{x+y}{2}\right)^2 = \frac{(x-y)^2}{2} \quad (4.2)$$

De forma similar, la distancia entre dos clusters G_i y G_j con n_i y n_j elementos respectivamente es:

$$d(G_i, G_j) = \frac{n_i n_j}{n_i + n_j} (\tilde{x}_i - \tilde{x}_j)^2 \quad (4.3)$$

donde $\tilde{x}_i$ es la media de los elementos de G_i y $\tilde{x}_j$ es la media de los elementos de G_j .

En cada iteración del algoritmo, los grupos elegidos para fusionarse son aquellos que tienen mínima distancia entre ellos. Cuando dos grupos se fusionan, las distancias del grupo resultante con respecto al resto de grupos, se deben de recalcular.

En [20, 7], se presenta una propuesta para la microagregación basada en el algoritmo de Ward. Este método, denominado *k-Ward*, para poder limitar el número de elementos de los clusters, se estructura en las siguientes etapas:

1. Formar un grupo con los k primeros (menores) elementos del conjunto de datos y otro grupo con los k elementos últimos (mayores) del conjunto.
2. Usar el método de Ward hasta que todos los elementos del conjunto pertenezcan a un grupo conteniendo k o más elementos. Durante el proceso, nunca unir dos grupos que tengan ambos un tamaño igual o superior a k .
3. Para cada grupo de la partición final que contenga $2k$ o más elementos, aplicar el algoritmo recursivamente (el conjunto inicial ahora se restringe a grupos particulares que tengan $2k$ o más elementos).

En 2002, se publica [18] una nueva versión (*secure-k-Ward*) que preserva la seguridad a nivel individual en base a los criterios de *nivel de tolerancia* y *ratio de seguridad*. Se incluyen al algoritmo dos nuevas etapas de optimización después del paso 2 (*intra-grupo* e *inter-grupo*). En la primera se

intenta minimizar la pérdida de información del cluster y en la segunda conseguir una mayor homogeneidad (menor desviación típica).

4.3.2. Algoritmo MDAV

El algoritmo de MDAV (Maximum Distance to Average Vector)[7] consiste en las siguientes etapas:

1. Se consideran los elementos más distantes al registro media (prototipo global), x_r , x_s , y se forman dos grupos alrededor de ellos. Un grupo contiene a x_r y a los $k - 1$ elementos más próximos a x_r (utilizando la distancia Euclídea). El otro grupo contiene a x_s y a los $k - 1$ elementos más próximos a x_s .
2. Si existen al menos $2k$ vectores de datos que no pertenecen a los dos grupos formados en el paso 1, se vuelve al paso 1 tomando como conjunto de datos los datos originales menos los contenidos en los grupos creados en el paso 1.
3. Si hay entre k y $2k - 1$ elementos que no pertenecen a los grupos formados en el paso 1, formar un nuevo grupo con estos elementos y acabar el algoritmo.
4. Si hay menos de k vectores de datos que no pertenecen a los grupos formados en el paso 1, añadirlos a los grupos más próximos respectivamente.

En [30] se presenta una mejora posterior del algoritmo denominada V-MDAV (*Variable -MDAV*) para conjuntos de elementos entre k y $(2k-1)$.

La primera definición del método consideraba datos numéricos continuos. Posteriormente, en [10] el procedimiento del algoritmo MDAV se generaliza para poder trabajar con cualquier tipo de atributo (continuo, ordinal, nominal) redefiniendo los operadores de cálculo de distancias y de medias.

Algoritmo (MDAV-generico) (R: dataset, k: integer) **is**

Mientras ($|R| > k$) **hacer**

Calcular el registro medio $\tilde{x}$ de todos los registros de R

Considerar el registro más distante x_r al registro media $\tilde{x}$ usando una distancia apropiada

Formar un cluster alrededor de x_r . El cluster contiene a x_r y a los k-1 registros más próximos a x_r

Eliminar estos registros del conjunto R

Si ($|R| > k$) **entonces**

Encontrar el registro más distante x_s al registro x_r (del paso 1.b)

Formar un cluster alrededor de x_s . El cluster contiene a x_s y a los k-1 registros más próximos a x_s

Eliminar estos registros del conjunto R

Fin Si

Fin Mientras

Formar un cluster con el resto de registros

Algoritmo 1: MDAV-Genérico

En [24] se orienta este algoritmo hacia la protección de datos que siguen temporales numéricas. Tomando como base el algoritmo de MDAV-genérico, se presentan modificaciones en los cálculos de los criterios:

- **Distancia:** para su simplificación, las secuencias son consideradas alineadas y de la misma longitud, por ello, su componente temporal es exactamente la misma para ambas secuencias. La distancia se emplea para poder identificar los registros más semejantes y dispares. La distancia o disimilitud de las secuencias en esta propuesta se calcula mediante:
 - *Distancia Euclídea:* basada en la distancia entre los componentes de datos.
 - *STS* (Short Time Series): basada en la forma de las series temporales.
- **Media:** necesaria para la obtención de los centroides de los registros de un cluster. Se realiza un tipo de media aritmética punto a punto.

4.3.3. Otras tendencias

En la actualidad, han ido surgiendo nuevas preferencias que difieren de las dos tendencias anteriores:

- **MHM algorithm** (Multivariate version of the Hansen-Mukherjee algorithm) [6] (2006): intenta optimizar el cálculo de la microagregación minimizando la pérdida de información.
- **μ -Approx algorithm** [8] (2008): aproximación a la microagregación óptima basado en la descomposición de grafos.
- **k-Means variation** [32] (2004): una modificación del algoritmo de k-means para datos categóricos.

Capítulo 5

Agregación de datos secuenciales

En determinados campos de aplicación, los atributos de los registros sobre los cuales se desean aplicar ciertas técnicas de protección están relacionados entre sí. Este hecho supone la necesidad de un cambio en el cálculo de las similitudes entre registros, es decir, una reorientación del enfoque clásico centrado en los valores concretos de los atributos hacia la identificación de la similitud en su conexión, consiguiendo así, la construcción de *clusters basados en datos secuenciales (o serializados)*, tal como se ha comentado en la introducción de esta tesina.

Resulta común que en los estudios sobre agregación de datos serializados se confundan los términos de «patrones secuenciales» con «series temporales». En [4, 27] se aclara que las técnicas de clustering sobre datos secuenciales se diferencian del análisis de las series temporales porque el momento preciso no es tan relevante y no se basan en la predicción, sino que se centran en la relación de secuencialidad de los datos. Por ello, en las técnicas de clustering, las secuencias son facilitadas de forma completa, mientras que en las predicciones temporales solamente se dispone de un prefijo de la secuencia. En la actualidad, gran cantidad de datos pueden considerarse datos secuenciales: transacciones comerciales, secuencias de proteínas, accesos a páginas web, recorridos de un cliente por un supermercado, etc.

5.1. Protección de datos secuenciales

La protección sobre datos que siguen secuencias o series se puede clasificar según el tipo de los datos sobre los que se aplique la protección:

- **Números:** variables numéricas continuas y discretas.
- **Símbolos:** letras o palabras.
- **Secuencias de patrones:** secuencias de secuencias o conjuntos de items.

5.2. Aproximaciones de técnicas de clustering con datos secuenciales

Según [19] se distinguen tres enfoques de métodos de clustering para series temporales (Figura 5.1):

- **Métodos basados en datos primarios** (raw-data-based): el requerimiento de estos métodos es poder diseñar una medida de similaridad directamente considerando los valores que toman un par de secuencias.
- **Métodos basados en características** (feature-based): convierten una serie de datos primarios en un vector de características de menor dimensión, normalmente numérico, lo que permite poder usar las técnicas clásicas posteriormente.
- **Métodos basados en modelos** (model-based): representan la secuencia de datos con un cierto número de los parámetros del modelo. Ésto implica la necesidad del diseño de un modelo adecuado para secuencias de datos, para posteriormente aplicar o no técnicas convencionales de clustering. Por ejemplo: modelos probabilísticos, modelos de Markov o modelos ocultos de Markov.

El estudio de esta tesina se ha orientado hacia la primera de las técnicas basada en datos primarios, que requiere adaptar las técnicas clásicas para poder tratar las particularidades de las secuencias. Para ello, ha sido necesario focalizar el trabajo en los siguientes aspectos:

- Definición de una medida de similaridad/disimilaridad entre los registros que contienen datos secuenciales.

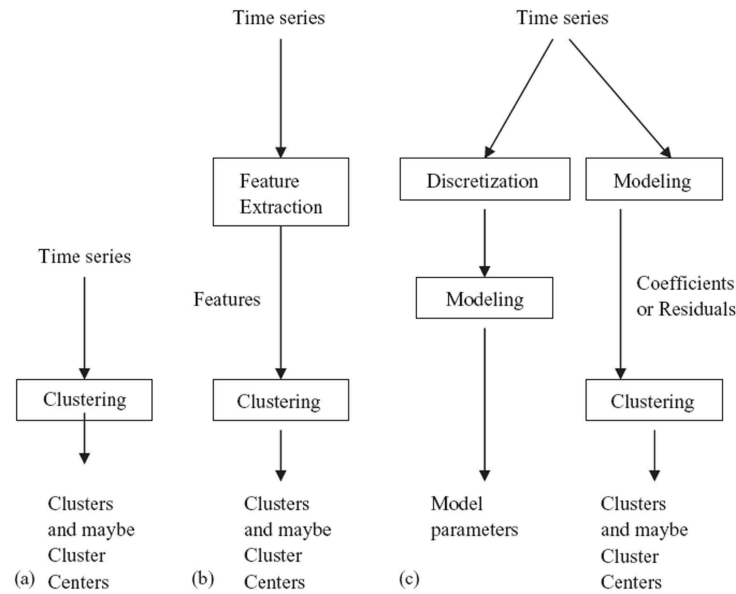


Figura 5.1: Tres enfoques de técnicas de clustering para series temporales: (a) raw-data-based, (b) feature-based, (c) model-based [19]

- Definición de un algoritmo de cálculo de prototipos (o centroides) específico para secuencias categóricas.

En la sección 2.3.1 ya se ha presentado el estado del arte del cálculo de dicha medida de similaridad para datos numéricos y categóricos. Por otro lado, el cálculo del prototipo (o elemento representativo del cluster) no siempre es necesario para aplicar clustering. Sin embargo, si lo es en el caso de la microagregación para protección de datos, tal y como se ha descrito en el apartado 4.1. La dificultad radica en que no existen apenas métodos para calcular prototipos de secuencias de eventos. En el siguiente apartado se presentan las principales características que debería de cumplir el prototipo.

5.3. Creación de prototipos de clusters con datos secuenciales

Un prototipo puede definirse como un ejemplo típico, base, o estándar de los elementos de un mismo cluster. Los prototipos (o proto-instancias) combinan los valores más representativos de atributos de los elementos que pertenecen al cluster. Consecuentemente, los prototipos son los casos típicos que representan el contenido del cluster.

Los prototipos se denominan «centroides» en técnicas de minería de datos. Resultan útiles en métodos clásicos de clustering, como, por ejemplo, el método Ward o en técnicas jerárquicas basadas en centroides [16]. En este ámbito, un centroide se define como el centro (o vector media) del cluster y se calcula mediante una función promedio.

5.3.1. Funciones promedio

Considerando su definición formal, se dice que:

Definición 1. *Una función de agregación f tiene un comportamiento promedio si por todo x se encuentra limitada por:*

$$\min(x) \leq f(x) \leq \max(x) \quad (5.1)$$

Una función promedio cumple las siguientes propiedades:

Definición 2. (*Idempotencia*). *Una función de agregación f es idempotente si para toda entrada $x = (t, t, \dots, t), t \in [0, 1]$, su salida cumple $f(t, t, \dots, t) = t$.*

Definición 3. (*Monotonidad estricta*). *Una función de agregación f es estrictamente monótona si $x \leq y$ pero $x \neq y$ implica $f(x) < f(y)$ para todo x, y .*

La media es uno de los operadores promedio más comunes, como por ejemplo, la media aritmética, media geométrica o media harmónica. Otros tipos se pueden encontrar en [3].

En el contexto de los datos categóricos, ambas propiedades de las funciones promedio se pueden aplicar si se puede establecer un orden sobre las categorías (datos ordinales). Por otro lado, en el caso de datos nominales (donde no se puede definir un orden sobre el conjunto de categorías posibles), solamente se puede requerir la idempotencia. En el caso de datos secuenciales, de forma similar, exclusivamente se exige la idempotencia ya que la otra propiedad no se puede aplicar.

En este contexto, sin embargo, se puede requerir que el valor promedio sea próximo al conjunto de datos que han sido agregados. Es decir, que el objeto c se encuentre a una mínima distancia de todos los datos x_i . Formalmente, dada una distancia d sobre el espacio de datos, la agregación de los datos $x_1, \dots, x_n$ se define como:

$$f(x_1, \dots, x_n) = \operatorname{argmin}_c \left\{ \sum_{i=1}^n d(c, x_i) \right\} \quad (5.2)$$

5.3.2. Prototipos de secuencias

En [11], se argumenta la necesidad de construir patrones de secuencias distintivas para un conjunto de secuencias relacionadas, denominándose *Sequence Motif*. El *Sequence Motif* se estudia principalmente en aplicaciones de análisis de secuencias biológicas. En estas aplicaciones, los *motifs* representan el resultado de la construcción durante la evolución. El análisis de *motifs* también predice la concordancia de las secuencias con un cierto *motif* y la posición donde se produce.

Aunque un *motif* no tiene la misma finalidad que un prototipo, se han estudiado los métodos propuestos para la construcción de *motifs*, con el fin de comprobar si son aplicables para la generación de prototipos.

Una de las principales representaciones (y más simples) de los *motifs* es una matrix de posiciones, denominada, **Position Weight Matrix** (PWM). Siendo A el alfabeto y $w > 0$ el tamaño de una ventana de una secuencia, la PWM es una matriz M de $A \times w$, donde $M(x, p)$ es un número proporcional a la frecuencia de un símbolo x en la posición p , aunque también puede verse influido por otros factores. Para usar este método, todas las secuencias deben tener una ventana de longitud w para ser comparadas. Algunos procesos previos de alineación se pueden realizar antes de calcular la PWM.

El método más simple para generar un PWM es el **Método de Frecuencias Directas**, donde cada posición de la matriz M es la frecuencia relativa de x en la posición p de la secuencia. Un ejemplo de un PWM se indica en el Cuadro 5.1.

	1	2	3	4	5	6
A	0	1	0	3/6	4/6	0
C	0	0	1/6	0	1/6	0
G	1/6	0	0	3/6	0	0
T	5/6	0	5/6	0	1/6	1

Cuadro 5.1: Ejemplo de matriz PWM

Aunque los *motifs* no se utilizan para el cálculo de prototipos, una secuencia de consenso se puede extraer de una PWM, mediante la inclusión de símbolos con valores elevados para cada posición. En el ejemplo del Cuadro 5.1, la secuencia de consenso obtenida es TAT(AG)AT, donde los símbolos entre paréntesis indican los elementos que no pueden ser resueltos ya que coinciden con la misma frecuencia para la misma posición.

Otros enfoques con supuestos más complejos se explican en [11], tales

como los métodos basados en modelos de Markov para cadenas.

Capítulo 6

Nueva propuesta para la agregación de datos secuenciales

Debido a la laguna existente en el área de la microagregación sobre datos que siguen patrones secuenciales y, concretamente, sobre atributos categóricos, éste ha sido el objetivo de esta tesina. A continuación se presentan los criterios que se han valorado y los algoritmos propuestos.

Siguiendo el esquema presentado en la sección 4.1, se ha estudiado cada uno de los siguientes aspectos de los métodos de microagregación para tipos de datos secuenciales categóricos:

- Función de similitud entre elementos del dominio.
- Método de construcción de los clusters.
- Cálculo de los prototipos para sustituir los datos originales.

En este capítulo se propone un método adecuado para cada fase del proceso.

6.1. Patrones identificados en datos secuenciales

Todos los ejemplos que se han añadido para facilitar la comprensión del cálculo consideran una letra como un elemento categórico independiente, donde *acp* podría representar una ruta de lugares de interés cultural en Tarragona siendo *a* = *anfiteatro*, *c* = *catedral* y *p* = *pretorio*. Así mismo,

se ha contrastado con el cálculo para pares de elementos, donde el ejemplo anterior acp tendría como elementos ac y cp .

6.2. Función de similitud: OSS

En [19] se presenta un resumen de medidas de similitud/distancia para datos secuenciales. Se definen nueve medidas y la mayoría de ellas sólo pueden aplicarse a valores numéricos. En los ejemplos de secuencias presentados anteriormente, los elementos de la secuencia no son números, pero sí valores categóricos (lugares, páginas web, proteínas, etc.). Aunque las secuencias de valores categóricos son muy importantes en la actualidad, todavía existen pocas propuestas de trabajo con ellos debido a la inherente complejidad del tratamiento de valores no numéricos.

En esta tesina se ha definido una nueva medida de similitud para dos secuencias categóricas que se basa en la comparación de los elementos comunes de las dos secuencias y las posiciones donde aparecen, denominada *Ordering-based Sequence Similarity*. Este trabajo a dado lugar a una publicación aceptada en el congreso internacional “Modeling Decisions for Artificial Intelligence” (MDAI) [14].

Definición 4. Siendo i y j dos secuencias de elementos de diferentes longitudes, $i = (x_{i,1}, \dots, x_{i,card(i)})$ y $j = (x_{j,1}, \dots, x_{j,card(j)})$, y siendo $L = \{l_1, \dots, l_n\}$ un conjunto de n símbolos para representar todos los posibles elementos de dichas secuencias (llamándose L el lenguaje), la función de similitud *Ordering-based Sequence Similarity* (OSS) se define como:

$$d_{OSS}(i, j) = \frac{g(i, j) + f(i, j)}{card(i) + card(j)} \quad (6.1)$$

donde

$$g(i, j) = card(\{x_{ik} | x_{ik} \notin j\}) + card(\{x_{jk} | x_{jk} \notin i\}) \quad (6.2)$$

y

$$f(i, j) = \frac{\sum_{k=1..n} (\sum_{p=1..\Delta} |i_{(l_k)}(p) - j_{(l_k)}(p)|)}{\max\{card(i), card(j)\}} \quad (6.3)$$

donde $i_{(l_k)} = \{t | i(t) = l_k\}$ y $\Delta = \min(card(i_{(l_k)}), card(j_{(l_k)}))$.

Esta función consta de dos partes:

- g que contabiliza el número de elementos no comunes entre las dos secuencias i y j .

- f que mide la similitud entre las posiciones de los elementos en las secuencias (su orden).

Para realizar el cálculo de la similitud f se transforma el espacio de representación de los elementos tomando el espacio de símbolos posibles L . Se proyecta la secuencia i sobre cada símbolo de L , obteniendo $i_{(l_1)} \dots i_{(l_n)}$. Cada uno de estos vectores contiene las posiciones de todos los símbolos en la secuencia i . Se realiza lo mismo con la secuencia j , obteniendo $j_{(l_1)} \dots j_{(l_n)}$. Puesto que estos vectores tienen la misma longitud, $\text{card}(L) = n$, las proyecciones de ambas secuencias i y j pueden ser comparadas. La similitud entre ambas se basa en sumar las diferencias entre las posiciones de cada símbolo. El valor obtenido se normaliza en base a la cardinalidad máxima de ambas secuencias i y j .

Si dos secuencias son iguales, el resultado de d_{OSS} es cero porque las posiciones son siempre iguales ($f = 0$) y no existen elementos no comunes ($g = 0$). Por otro lado, si las dos secuencias no tienen ningún elemento en común, entonces $g = \text{card}(i) + \text{card}(j)$ y $f = 0$ y, por tanto, d_{OSS} es igual a 1 cuando se divide por $\text{card}(i) + \text{card}(j)$. La función *Ordering-based Sequence Similarity* siempre retorna valores entre 0 y 1.

La función posee las siguientes propiedades:

- Simetría: $d_{OSS}(i, j) = d_{OSS}(j, i)$
- Positividad: $d_{OSS}(i, j) \geq 0$ for all i, j
- Reflexividad: $d_{OSS}(i, j) = 0$ iff $i = j$

Sin embargo no se cumple la Desigualdad Triangular: $d_{OSS}(i, j) \leq d_{OSS}(i, k) + d_{OSS}(k, j)$ para todo i, j, k .

De estas propiedades se desprende que d_{OSS} es una dissimilitud pero no una distancia.

Demostración. La demostración de Simetría, Positividad y Reflexividad es trivial según la Definición de OSS (eq. 6.1). La Desigualdad Triangular no se cumple y se demuestra en el siguiente contraejemplo.

Sean A , B y C tres secuencias definidas por $A = \{b, c\}$, $B = \{d, a\}$ y $C = \{d, a, b, c\}$. En este caso, $d_{OSS}(A, B) = 1,0$ porque no comparten ningún elemento y $d_{OSS}(A, C) = 0,5$ porque tienen dos elementos en común. B y C también tienen otros dos elementos en común (y además en la misma posición), así que, $d_{OSS}(B, C) = 0,33$. Consecuentemente, $d_{OSS}(A, C) +$

$d_{OSS}(B, C) = 0,83$ que es inferior que $d_{OSS}(A, B)$ que es 1.0, lo cuál demuestra que no se cumple la desigualdad triangular. $\square$

Cabe destacar, que esta medida puede ser aplicada a diferentes elementos: eventos individuales o grupos de eventos. Dada una secuencia $\{a, b, a, c, d\}$, en el primer caso, $i = (a, b, a, c, d)$, entonces x_{ij} es un evento individual de la secuencia. En el segundo caso, $i = (ab, ba, ac, cd)$, entonces x_{ij} es un par consecutivo de elementos, e $i = (aba, bac, acd)$ para tríos de elementos, y así consecuentemente.

El siguiente ejemplo muestra como se calcula d_{OSS} para eventos individuales (d_{OSS-1}) y para pares de eventos (d_{OSS-2}). Se dispone de las dos siguientes secuencias: $A = \{a, b, c, a\}$, $B = \{c, a, d, b, c, a, c\}$, con $card_A = 4$ y $card_B = 7$.

La similaridad para elementos individuales es $d_{OSS-1}(A, B) = 0,36$. Este resultado se obtiene de la siguiente forma: los símbolos a , b y c son comunes en ambas secuencias A y B . La proyección del símbolo a es: $A_{(a)} = \{0, 3\}$ y $B_{(a)} = \{1, 5\}$, así que $f_a(A, B) = |0-1| + |3-5| = 3$. Para el símbolo b : $A_{(b)} = \{1\}$, $B_{(b)} = \{3\}$ y $f_b(A, B) = |1-3| = 2$. Para c : $A_{(c)} = \{2\}$, $B_{(c)} = \{0, 4, 6\}$ y $f_c(A, B) = |2-0| = 2$. De esta forma, $f(A, B) = \frac{f_a(A, B) + f_b(A, B) + f_c(A, B)}{7} = 1$. Por otro lado, al calcular los elementos no comunes, tenemos $g(A, B) = 3$. Finalmente, $d_{OSS-1}(A, B) = \frac{f(A, B) + g(A, B)}{4+7} = \frac{1+3}{11} = 0,36$.

Considerando el mismo ejemplo con patrones de longitud 2, la disimilaridad aumenta $d_{OSS-2}(A, B) = 0,629$. En este caso, las secuencias quedan divididas $A' = \{ab, bc, ca\}$ y $B' = \{ca, ad, db, bc, ca, ac\}$ con cardinalidades 3 y 6. Ambas comparten 2 elementos. Para el par bc se obtiene $A_{(bc)} = \{1\}$, $B_{(bc)} = \{3\}$ y $f_{bc}(A, B) = |1-3| = 2$, mientras que para el par ca : $A_{(ca)} = \{2\}$, $B_{(ca)} = \{0, 4\}$ y $f_{ca}(A, B) = |2-0| = 2$. De modo que $f(A, B) = \frac{f_{bc}(A, B) + f_{ca}(A, B)}{6} = 0,66$. Calculando los elementos no comunes, se obtiene $g(A, B) = 5$. Finalmente, $d_{OSS-2}(A, B) = \frac{f(A, B) + g(A, B)}{3+6} = \frac{0,66+5}{9} = 0,629$.

6.3. Construcción de clusters: Método KSHC

Anteriormente se han presentado los algoritmos para microagregación k-Ward y MDAV (sección 4.3). Estos algoritmos han sido considerados para su utilización en el caso de secuencias de valores categóricos. Sin embargo, las características de estos métodos no son adecuadas para el caso que nos ocupa.

A continuación se detallan los inconvenientes de ambos métodos:

- *k-Ward*: este método tiene dos aspectos que impiden su uso para datos secuenciales categóricos: (1) se necesita un orden total entre los elementos del conjunto, y (2) se usa la medida de la variancia para decidir los elementos a juntar. En el primer punto, no se puede definir una función de orden entre secuencias de datos categóricos, que permita determinar qué secuencia es la menor, y cuál la mayor, puesto que no hay un criterio de ordenación entre ellas. En cuanto al segundo aspecto, el cálculo de la variancia con datos categóricos no es factible.
- *MDAV*: este algoritmo parte de un prototipo global inicial de todo el conjunto de datos, sin embargo, no es viable calcular un prototipo que resuma todas las secuencias de valores categóricos del conjunto de datos.

Debido a estos inconvenientes, estos dos algoritmos y sus variantes han sido descartados. Así pues, en esta tesina se ha diseñado un nuevo algoritmo que preserve el *k*-anonimato y que no requiera del cálculo de prototipos durante la formación de los clusters.

Para ello se estudiaron los algoritmos de clustering presentados en la sección 2.4, para ver cuáles cumplían las características necesarias y podían ser adaptados para ser usados en microagregación. Se seleccionaron los algoritmos de clasificación jerárquica aglomerativa, por ser muy conocidos y estudiados. De entre los diferentes métodos se descartaron el método de Ward (ya comentado anteriormente) y el método del Centroide, puesto que queremos evitar el cálculo de prototipos en etapas intermedias. Así pues, a continuación se presenta un algoritmo jerárquico aglomerativo que puede aplicar el método del mínimo (Single Linkage), o el método del máximo (Complete Linkage) o el de la distancia promedio (Average Linkage).

El algoritmo que se propone es una adaptación del método clásico, para el caso de microagregación, que asegura obtener clusters de tamaño limitado entre k y $2k - 1$. A este algoritmo se le ha denominado **K-Sized Hierarchical Clustering** (KSHC).

Antes de explicar el algoritmo es necesario definir unos conceptos previos:

Definición 5. *Se denomina **Cluster Válido** a aquel que tiene entre k y $2k - 1$ elementos.*

Definición 6. *El número de Clusters Válidos dado un conjunto de datos de tamaño n , está comprendido entre un mínimo $minVC$ y un máximo $maxVC$, de la siguiente forma:*

$$minVC = \left\lceil \frac{n}{(2k-1)} \right\rceil \quad (6.4)$$

$$maxVC = \left\lfloor \frac{n}{k} \right\rfloor \quad (6.5)$$

El algoritmo KSHC está estructurado en dos partes diferenciadas:

• **Step 1:** Se identifica el número mínimo de clusters válidos. En cada iteración, se localizan los dos registros más similares de entre todos los elementos individuales o clusters a agrupar (mínimo valor matriz de disimilitud). Su agregación se realizará según el tipo de método aglomerativo que se haya escogido (ver Sección 2.4.1), habiéndose implementado para este estudio los dos siguientes:

- *Single Linkage Method:* el resultado de la fusión de los registros se corresponde con la disimilitud mínima.
- *Complete Linkage Method:* el resultado de la fusión de los registros se corresponde con la disimilitud máxima.

Si después de crear un cluste, éste resulta ser un Cluster Válido, se reserva para el Step 2, y se continúa la microagregación sin este cluster.

• **Step 2:** En este punto se dispone de un conjunto de Clusters Válidos y unos elementos restantes. Estos elementos se distribuyen por los Clusters Válidos o, si fuese necesario, se crean nuevos clusters resultado de la agregación de elementos restantes, comprobando siempre que al final se obtenga una partición de Clusters Válidos. En cada iteración, se identifican los dos registros más similares de entre todos los elementos individuales o clusters a agrupar (mínimo valor de la matriz de disimilitud). Al igual que en el paso anterior, su agregación puede efectuarse de acuerdo a dos métodos aglomerativos (single/complete linkage). Si en alguna agregación de registros, un cluster alcanza k elementos, entonces se incrementa el número de Clusters Válidos. En cada paso, se pueden dar los siguientes casos: si la unión es igual o inferior a $2k-1$, se fusionan sin problemas; en cambio, si la unión iguala o supera el tamaño de $2k$ entonces:

- Si se quieren unir dos Clusters Válidos, no se permite su agrupación puesto que superaría el valor máximo permitido.

- Si uno de los dos clusters es no Válido, se le añade el elemento más similar del Cluster Válido, sabiendo que este seguirá siendo válido.
- Si uno de los componentes de la unión es un elemento simple, se crea un nuevo cluster con dicho elemento y el elemento más similar a éste del Cluster Válido.

El algoritmo KSHC tiene un coste de $O(n^2)$, siendo n el número de elementos en el conjunto de datos N .

El siguiente ejemplo ilustra el funcionamiento de este método. Dados los siguientes nueve registros (Cuadro 6.1) y tomando como base la matriz de disimilaridad construida mediante la función $d_{OSS} - 1$ (Cuadro 6.3), se desea realizar una agregación de elementos de tamaño $k = 3$.

Id	Sequence	Id	Sequence
0	a b e	5	c f f i
1	b c d e	6	f i
2	f f c	7	f i c
3	f i f c	8	f f b i
4	c b d e		

Cuadro 6.1: Ejemplo de secuencias de datos

Id.Reg.	0	1	2	3	4	5	6	7	8
0	0.0	0.5	1.0	1.0	0.46	1.0	1.0	1.0	0.75
1	0.5	0.0	0.75	0.81	0.06	0.78	1.0	0.75	0.81
2	1.0	0.75	0.0	0.21	0.79	0.29	0.6	0.3	0.43
3	1.0	0.81	0.21	0.0	0.84	0.19	0.3	0.18	0.34
4	0.46	0.06	0.78	0.84	0.0	0.75	1.0	0.79	0.78
5	1.0	0.78	0.29	0.19	0.75	0.0	0.46	0.32	0.31
6	1.0	1.0	0.6	0.3	1.0	0.46	0.0	0.2	0.42
7	1.0	0.75	0.3	0.18	0.79	0.32	0.2	0.0	0.5
8	0.79	0.81	0.43	0.34	0.78	0.31	0.42	0.5	0.0

Cuadro 6.2: Matriz de disimilaridad

Al finalizar el *Step 1* se obtienen los siguientes clusters:

- Cluster 9: 1,4 (Inválido)
- Cluster 11: 3,5,7 (Válido)
- Cluster 13: 2,6,8 (Válido)

quedando el elemento 0 sin fusionar.

En el *Step 2*, se consideran los Clusters 11, 12 y 13 y el elemento 0, como elementos a clasificar. La matriz tiene el mínimo 0.2 de estos elementos en la fusión del Cluster 11 con el 13. Sin embargo, su fusión implicaría un cluster de tamaño igual a $2k$. Por dicho motivo, se invalida la agregación. El siguiente mínimo de la tabla, 0.46, se corresponde con la fusión del Cluster 9 con el elemento 0. Esta agregación es válida. En este punto, el método finaliza porque no quedan elementos sin agregar a los clusters.

A continuación se muestra el pseudo-código del algoritmo que se ha diseñado y utilizado en esta tesina para la construcción de los clusters.

Algoritmo KSHC (N: dataset, k: integer) **is**

Construir la matriz de disimilitudes D para |N|elementos ; Nclust=0 ;

Mientras (Nclust < minVC) **hacer** // *Step 1*

Buscar el mínimo de la matriz

Crear un nuevo cluster con los dos objetos (o clusters) más próximos

Eliminar los dos objetos (o clusters) de la matriz D

Si tamaño(cluster) < k **entonces**

Añadir el nuevo cluster a la matriz D, calculando sus disimilitudes

Sino

Reservar el cluster

Nclust = Nclust + 1

Fin Si

Fin Mientras

Añadir los clusters reservados a la matriz D, calculando sus disimilitudes

Algoritmo 2: KSHC (K-sized Hierarchical Clustering) (I)

//Continuación

Mientras (tamaño(D)>0) **hacer** // Step 2

 Buscar el mínimo de la matriz

Si tamaño(elementos a fusionar) $\geq 2k$ **entonces**

Si fusion de un objeto con un cluster válido

 Crear nuevo cluster con objeto y elemento más próximo

 Eliminar objeto de la matriz D

 Eliminar elemento del cluster válido más próximo al objeto

 Añadir el nuevo cluster a la matriz D

 Recalcular disimilitudes

Sino 4

Si fusión de un cluster válido con uno no válido **entonces**

 Eliminar el elemento más próximo del cluster válido

 Añadir el elemento más próximo al cluster no válido

 Recalcular disimilitudes para los dos

Sino Cambiar disimilitud por máximo para impedir fusión

FinSi

FinSi

Si (tamaño(cluster) = k) **entonces** Nclust = Nclust + 1

Fin Si

Sino

 Crear un nuevo cluster con los dos objetos (o clusters) más próximos

 Eliminar los dos objetos (o clusters) de la matriz D

 Añadir el nuevo cluster a la matriz D, calculando sus disimilitudes

 NClust = NClust + 1 (si no contabilizado en el Step 1)

Fin Mientras

FAlgoritmo

Algoritmo 3: KSHC (K-sized Hierarchical Clustering) (II)

6.4. Cálculo de prototipos: Método OESP

Los centroides o prototipos de los clusters obtenidos en el paso anterior se utilizan para enmascarar los datos originales y generar una nueva versión que no permita la re-identificación de los individuos. En el caso que nos ocupa, todos los individuos de un mismo cluster sustituyen sus respectivas secuencias de eventos por una secuencia prototipo. En esta tesina se ha diseñado también un método para generar el prototipo de un conjunto secuencias de valores categóricos, denominado **Ordered Element Scoring Prototyping**(OESP).

Siendo $c(G_i)$ el prototipo de un cluster de secuencias categóricas G_i , se considera que el prototipo debe cumplir las siguientes propiedades:

- $c(G_i)$ es una secuencia categórica.
- $c(G_i)$ solamente contiene símbolos (por ejemplo, etiquetas categóricas) que se pueden encontrar en G_i
- Los símbolos en $c(G_i)$ aparecen en un orden similar que los encontrados en las secuencias de G_i
- La longitud de $c(G_i)$ es similar a la media de las longitudes de las secuencias de G_i

El método propuesto para la generación del prototipo se basa en el cálculo de una tabla de puntuaciones **Element Scoring Table** (EST), donde los elementos pueden ser o elementos individuales o patrones de eventos secuenciales considerados como unidades indivisibles. A este método se le ha denominado **Ordered Element Scoring Prototyping** (OESP).

Considerando cada elemento como un símbolo asociado al lenguaje L , el algoritmo *OESP* consiste en las siguientes etapas:

1. Los símbolos de $s \in L$ se transforman en $s' \in L'$, mediante la adición a cada símbolo s de un número, n , que indica su posición relativa en la secuencia. La primera vez que un símbolo aparece, se le asigna el número $n = 1$, la segunda vez $n = 2$, y así sucesivamente.

Esta recodificación permite distinguir la repetición de los elementos que se encuentren en una misma secuencia como elementos diferentes para poder evaluarlos correctamente y de forma que la secuencia prototipo también disponga de elementos repetidos.

Por ejemplo, si se dispone de un cluster con cinco registros (secuencias) $G = \{abc, ac, ab, cbac, cab\}$, cada uno de ellos se codificará de la forma siguiente:

$$r(abc) = \{a1, b1, c1\}$$

$$r(ac) = \{a1, c1\}$$

$$r(ab) = \{a1, b1\}$$

$$r(cbac) = \{c1, b1, a1, c2\}$$

$$r(cab) = \{c1, a1, b1\}$$

2. Identificación de todos los símbolos L' que aparecen en r . Estos símbolos forman el conjunto $E \subset L'$.

Siguiendo con el ejemplo, la lista de símbolos diferentes es $E = \{a1, b1, c1, c2\}$.

3. Construcción de la tabla *Element Scoring Table* (EST), con los símbolos de E y sus correspondientes puntuaciones. Para cada símbolo, se le van asignando diferentes puntuaciones según la posición en la que aparece en cada registro. Siendo p la posición del símbolo en uno de los registros, su resultado se incrementará en x puntos, proporcionalmente al tamaño del cluster G_i :

$$x = |G_i| - p - 1 \quad (6.6)$$

En el ejemplo, se obtiene el resultado en el Cuadro 6.3.

Id	Suma	Resultado
a1	4+4+4+2+3	17
b1	3+3+3+2	11
c1	2+3+4+4	13
c2	1	1

Cuadro 6.3: Resultado de la suma de las puntuaciones de los registros

4. Ordenación decreciente de todos los elementos de E en base a su puntuación (frecuencia).

Para el ejemplo, la tabla ordenada se muestra en el Cuadro 6.4.

5. Creación del prototipo concatenando los símbolos iniciales siguiendo el orden obtenido en el paso anterior. Es decir, el símbolo con mayor puntuación aparece en la primera posición del prototipo.

Posteriormente, se selecciona un segundo símbolo para añadirlo al prototipo. La selección del símbolo depende de la naturaleza de los elementos. Se distinguen dos formas:

Id	Resultado
a1	17
c1	13
b1	11
c2	1

Cuadro 6.4: Registros ordenados por puntuación

- **Elementos simples:** los símbolos se adicionan según el orden decreciente obtenido en los resultados de la tabla EST. En este ejemplo, el prototipo es *acbc*.
 - **Patrones:** el siguiente símbolo se fusiona al final del prototipo. Si el patrón tiene t eventos, el siguiente elemento de E debe compartir los $t - 1$ símbolos previos con la última parte del prototipo. Por ejemplo, considerando que E tiene 5 patrones de símbolos en el siguiente orden decreciente: *dae*, *abe*, *ebc* y *aeb*. El prototipo inicial es *dae*. El siguiente símbolo debe comenzar por *ae* para fusionarse correctamente con la secuencia. Siguiendo el orden, se elige el símbolo *aeb*, y el prototipo se convierte en *daeb*. En el siguiente paso, se selecciona *ebc*, obteniendo *daebc*. Esta operación es necesaria para poder seguir manteniendo la correlación de orden que se ha deseado valorar inicialmente. En el caso de que no exista un símbolo que comparta los elementos requeridos y E todavía disponga de elementos no adicionados al prototipo, se añade el primero con mayor puntuación. Es decir, se fuerza la creación de un patrón no existente realmente, pero resultando imprescindible esta fusión si no se dispone de patrones con elementos simples comunes o si no quedan elementos no relacionados todavía por añadir. En el anterior ejemplo, el prototipo final sería *daebcabe*.
6. Reducción del prototipo para ajustarlo a la longitud media de las secuencias del cluster. Los elementos del final de la secuencia prototipo se eliminan hasta alcanzar la longitud deseada.

Para el cluster G , la longitud media es 3, así que el prototipo candidato *acbc* se reduce al prototipo $c(G) = acb$.

En ciertas situaciones, esta longitud podría no garantizar la protección de alguno de los registros iniciales, por tanto, se debería aumentar. Es importante encontrar el equilibrio entre representatividad de los registros iniciales frente a su adecuada protección. Para poder preservar los registros iniciales es necesario que éstos no sean a su vez subregistros de otros registros del cluster.

Capítulo 7

Experimentos

Para validar la calidad de los algoritmos propuestos anteriormente, se ha diseñado una sencilla e intuitiva aplicación en Java que permite la lectura de un fichero de datos en Excel y que devuelve los resultados de la microagregación. En la sección 7.2 se describen las pruebas que se han realizado con conjuntos de datos preparados para testear los distintos aspectos del método y con un conjunto de datos real de itinerarios de turistas.

7.1. Aplicación software de testeo

En la Figura 7.1 se presenta el interfaz de usuario de la aplicación. Consta de los siguientes campos a seleccionar:

- **File** (*Archivo*): archivo de datos Excel origen. Dispone de un botón de búsqueda de facilita el hallazgo de la ruta del fichero con el que se desea trabajar.
- **Similarity** (*Similaridad*): tipo de elementos que se consideran para calcular la similaridad entre dos registros. Determina la longitud para aplicar la función de similaridad (*window*).
 - *Individual*: cada uno de los elementos de la secuencia es analizado individualmente.
 - *Sequences of length 2*: los datos son analizados en parejas. Se genera una dependencia de cada elemento con respecto a su anterior y posterior.

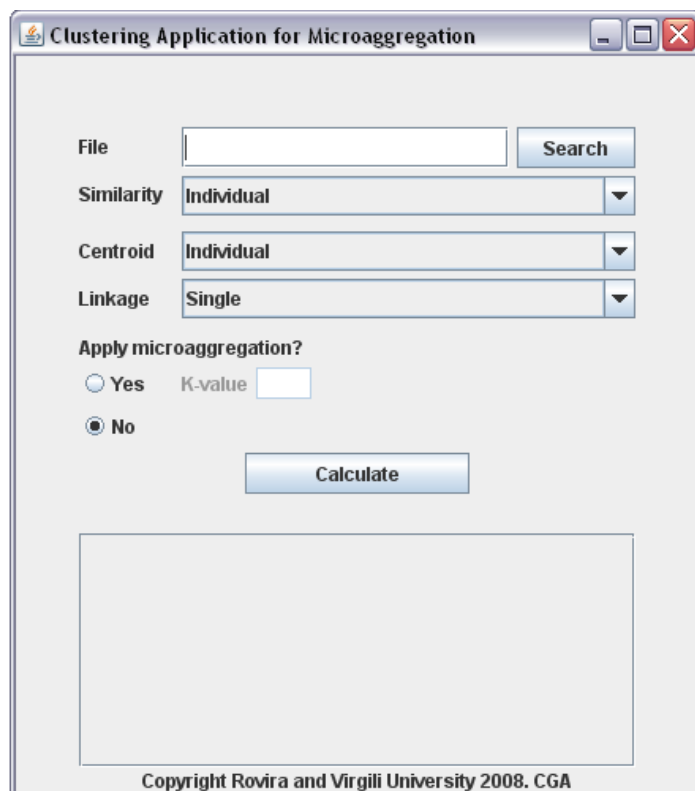


Figura 7.1: Interfaz gráfica de la aplicación desarrollada

- **Centroid** (*Centroide*): tipo de elementos a usar para la construcción del elemento prototipo o centroide.
 - *Single*: se toma la distancia mínima respecto a los elementos del cluster.
 - *Complete*: se toma la distancia máxima respecto a los elementos del cluster.
- **Linkage** (*Enlace*): método de enlace para formar nuevos clusters. Al igual que en la propiedad anterior, se puede escoger entre *Single* o *Complete*.
- **Apply microaggregation?** (*¿Aplicar microagregación?*): Permite limitar el tamaño de los clusters. El parámetro k establece un número mínimo de elementos en los clusters que hará que su tamaño oscile entre k y $2k - 1$. Para garantizar la homogeneidad de los clusters y, por tanto, la representatividad de los centroides que sustituyen a los datos originales, se recomienda que k no sea excesivamente elevada y se encuentre en coherencia con la distribución y tamaño de los datos

originales. Por otro lado, para que la protección de los datos originales sea suficiente y el riesgo de revelación no sea elevado, las secuencias de datos categóricos no deberían de ser sub-secuencias de otras secuencias y la k debería ser igual o superior a 3, ya que sino sería excesivamente débil la técnica perturbativa aplicada.

Falicitando los parámetros anteriores y pulsando el botón *Calcular*, se muestra en la parte inferior de la aplicación una tabla con los clusters definidos, donde cada fila contiene su número identificativo, su centroide (reducido y completo), el número de registros que integran el cluster, el SSE de los registros respecto al centroide y su lista de registros.

7.2. Pruebas

7.2.1. Descripción de los datos

Como sucede habitualmente en muchas técnicas de Inteligencia Artificial, la naturaleza de los valores del conjunto de datos determina las características del método que puede aplicarse, tal y como se ha explicado en la sección 2.2. La clasificación más común distingue: los valores numéricos frente a los valores categóricos. Los resultados numéricos puede ser continuos, discretos o intervalos, y pueden representar mediciones cuantitativas, ratios o escalas ordinales. Los valores categóricos representan características cualitativas con o sin orden [17]. Pero, de hecho, existen otras representaciones de datos que puede ser consideradas como textuales, espaciales, imágenes o multimedia.

Respecto a los patrones secuenciales, además se pueden identificar otras propiedades, como si las muestras de datos son uniformes o no, con valores univariados o multivariados, o si son de igual o distinta longitud [19].

En este trabajo, se tratan secuencias de eventos que representan el comportamiento del usuario en un contexto particular. Por ejemplo, los turistas que visitan una ciudad, donde las secuencias muestren el itinerario que cada persona ha seguido para la visita de los lugares más interesantes. El profesor Dr. Shoval nos ha facilitado un conjunto privado de datos reales correspondientes a los itinerarios de turistas. Este conjunto de datos consta de 25 lugares de interés (elementos) y presenta 39 itinerarios con longitudes entre 10 y 40 elementos (permitiéndose la repetición).

Otro ejemplo de este tipos de datos son las secuencias de visitas a una página de Microsoft. Esta información se encuentra pública en el repositorio UCI Machine Learning Repository [2]. Los registros se obtienen mediante

el muestreado y proceso de los logs de la página www.microsoft.com. El conjunto de registros se corresponde con el comportamiento de 38.000 usuarios anónimos. Para cada usuario, se dispone de la lista de áreas (*vroots*) que ha visitado en la web durante una semana. El número total de *vroots* disponibles es de 294, y el número medio de visitas a cada *vroot* por usuario es de 3.

Estos dos ejemplos de secuencias de eventos poseen las siguientes características:

- Los eventos son valores categóricos que pertenecen a un conjunto finito de etiquetas lingüísticas. (lugares de una ciudad o páginas web).
- Los elementos de la secuencia siguen una distribución uniforme en el tiempo, puesto que no se considera su duración.
- Las secuencias son univariantes, es decir, solamente se estudia un concepto.
- Las longitudes de las secuencias de los individuos no son iguales.
- Los eventos se pueden repetir en la secuencia (por ejemplo, un turista puede visitar una zona más de una vez durante sus vacaciones).

7.2.2. Función OSS

En este apartado se presentan los resultados del estudio de la función de similitud OSS, con independencia del resto del método. El conjunto de test consta de 14 registros, correspondientes a las secuencias de eventos presentadas en el Cuadro 7.1. Este conjunto contiene secuencias de diferentes longitudes. El Registro 9 es un caso extremo, ya que contiene un único evento en la secuencia. Las secuencias de los registros 12, 13 y 14 muestran secuencias con eventos repetidos.

Se ha evaluado la función de similaridad Ordering-based Sequence Similarity sobre el conjunto considerando los eventos individualmente, OSS-1 (Cuadro 7.2) y mediante patrones de dos eventos, OSS-2 (Cuadro 7.3).

Se observa en los resultados, que la función OSS aplicada sobre pares de eventos ofrece valores de disimilaridad superiores en muchos casos (valor 1 en el Cuadro 7.3). Esto se debe a que la existencia de pares comunes de eventos es mucho más difícil que la existencia de eventos individuales comunes.

Para OSS-2, las secuencias $\{e, d, a, b, c\}$ (id=6) y $\{f, e, d, a, b, c\}$ (id=7) son las más similares (valor 0.19), porque comparten los 4 pares (ed, da, ab, bc)

Id	Sequence	Id	Sequence
1	a b	8	c g d a b c
2	b c	9	d
3	a b c	10	d a
4	c a b	11	d b
5	d a b c	12	c b c b c b
6	e d a b c	13	b c b c b c
7	f e d a b c	14	e b e b e b

Cuadro 7.1: Ejemplo de secuencias de datos

Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0.62	0.2	0.33	0.41	0.54	0.62	0.62	1	0.62	0.5	0.75	0.77	0.75
2	0.62	0	0.33	0.4	0.5	0.59	0.66	0.60	1	1	0.62	0.54	0.5	0.77
3	0.2	0.33	0	0.22	0.25	0.4	0.5	0.48	1	0.66	0.6	0.59	0.59	0.77
4	0.33	0.4	0.22	0	0.25	0.4	0.5	0.40	1	0.6	0.66	0.57	0.61	0.79
5	0.41	0.5	0.25	0.25	0	0.19	0.33	0.35	0.6	0.33	0.37	0.66	0.66	0.81
6	0.54	0.59	0.4	0.4	0.19	0	0.16	0.37	0.7	0.48	0.51	0.72	0.72	0.66
7	0.62	0.66	0.5	0.5	0.33	0.16	0	0.40	0.76	0.58	0.60	0.77	0.77	0.72
8	0.62	0.60	0.48	0.40	0.35	0.37	0.40	0	0.76	0.58	0.60	0.58	0.59	0.87
9	1	1	1	1	0.6	0.7	0.76	0.76	0	0.33	0.33	1	1	1
10	0.62	1	0.66	0.6	0.3	0.48	0.58	0.58	0.33	0	0.5	1	1	1
11	0.5	0.62	0.6	0.66	0.37	0.51	0.60	0.60	0.33	0.5	0	0.75	0.77	0.75
12	0.75	0.54	0.59	0.57	0.66	0.72	0.77	0.58	1	1	0.75	0	0.08	0.5
13	0.77	0.5	0.59	0.61	0.66	0.72	0.77	0.59	1	1	0.77	0.08	0	0.54
14	0.75	0.77	0.77	0.79	0.81	0.66	0.72	0.87	1	1	0.75	0.5	0.54	0

Cuadro 7.2: Resultados de aplicar OSS a eventos individuales (OSS-1)

Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1	0.33	0.5	0.58	0.7	0.76	0.76	-	1	1	1	1	1
2	1	0	0.5	1	0.66	0.75	0.8	0.8	-	1	1	0.7	0.66	1
3	0.33	0.5	0	0.62	0.33	0.5	0.59	0.59	-	1	1	0.71	0.74	1
4	0.5	1	0.62	0	0.6	0.70	0.77	0.77	-	1	1	1	1	1
5	0.58	0.66	0.33	0.6	0	0.25	0.4	0.4	-	0.5	1	0.77	0.8	1
6	0.7	0.75	0.5	0.70	0.25	0	0.19	0.39	-	0.65	1	0.82	0.84	1
7	0.76	0.8	0.59	0.77	0.4	0.19	0	0.4	-	0.73	1	0.86	0.88	1
8	0.76	0.8	0.59	0.77	0.4	0.39	0.4	0	-	0.73	1	0.86	0.88	1
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	1	1	1	1	0.5	0.65	0.73	0.73	-	0	1	1	1	1
11	1	1	1	1	1	1	1	1	-	1	0	1	1	1
12	1	0.7	0.71	1	0.77	0.82	0.86	0.86	-	1	1	0	0.28	1
13	1	0.66	0.74	1	0.8	0.84	0.88	0.88	-	1	1	0.28	0	1
14	1	1	1	1	1	1	1	1	-	1	1	1	1	0

Cuadro 7.3: Resultados de aplicar OSS a pares de eventos (OSS-2)

en posiciones muy próximas y solamente difieren en el par (fe). Los siguientes, son las secuencias $\{d, a, b, c\}$ (id=5) con id=6, que tienen tres pares en común y un par no común. Y en tercera posición, se encuentran las secuencias $\{c, b, c, b, c, b\}$ (id=12) e $\{b, c, b, c, b, c\}$ (id=13), que comparten 4 pares, pero difieren en 2 no comunes.

Por otro lado, la medida OSS-1 encuentra la mínima disimilaridad entre los registros id=12 e id=13 (0.08) en lugar de entre los registros id=6 e id=7 (0.19). Ésto se debe a que, en el primer caso, los registros comparten todos sus símbolos, mientras que en el segundo, tienen en común 5 símbolos y uno no común. Desde nuestro punto de vista, OSS-1 es más útil para detectar el grado de similaridad entre dos secuencias de eventos que OSS-2, ya que prioriza que los eventos aparezcan en posiciones similares, en lugar de su aparición conjunta en pares.

Después de analizar los resultados de OSS, se ha comparado su comportamiento con la medida *Edit Distance*, muy común para la comparación de secuencias [11]. La medida *Edit Distance* contabiliza el número de cambios que son necesarios aplicar sobre una secuencia para obtener otra. Para escalar los valores en un intervalo de unidades, se ha dividido el número de cambios por la longitud de la secuencia más larga. El Cuadro 7.4 muestra los resultados de la normalización de la *Edit Distance* (ED) sobre el mismo conjunto de datos.

Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1	0.33	0.33	0.5	0.6	0.66	0.66	1	1	0.5	0.83	0.83	0.83
2	1	0	0.33	1	0.5	0.6	0.66	0.66	1	1	1	0.66	0.66	0.83
3	0.33	0.33	0	0.66	0.25	0.4	0.5	0.5	1	1	0.66	0.66	0.66	0.83
4	0.33	1	0.66	0	0.5	0.6	0.66	0.5	1	0.66	0.66	0.66	0.66	0.83
5	0.5	0.5	0.25	0.5	0	0.2	0.33	0.33	0.75	0.5	0.5	0.66	0.66	0.83
6	0.6	0.6	0.4	0.6	0.2	0	0.16	0.33	0.8	0.6	0.6	0.66	0.66	0.66
7	0.66	0.66	0.5	0.66	0.33	0.16	0	0.33	0.83	0.66	0.66	0.83	0.66	0.83
8	0.66	0.66	0.5	0.5	0.33	0.33	0.33	0	0.83	0.66	0.66	0.66	0.66	1
9	1	1	1	1	0.75	0.8	0.83	0.83	0	0.5	0.5	1	1	1
10	1	1	1	0.66	0.5	0.6	0.66	0.66	0.5	0	0.5	1	1	1
11	0.5	1	0.66	0.66	0.5	0.6	0.66	0.66	0.5	0.5	0	0.83	0.83	0.83
12	0.83	0.66	0.66	0.66	0.66	0.66	0.83	0.66	1	1	0.83	0	0.33	0.5
13	0.83	0.66	0.66	0.66	0.66	0.66	0.66	0.66	1	1	0.83	0.33	0	0.66
14	0.83	0.83	0.83	0.83	0.83	0.66	0.83	1	1	1	0.83	0.5	0.66	0

Cuadro 7.4: Resultados según la medida *Edit Distance* (ED)

Esta dificultad de distinción entre dos secuencias que tienen los mismos elementos, pero en diferente orden, es el principal inconveniente de la *Edit Distance* [37]. Un caso extremo es el resultado con secuencias como $\{a, a, b, b\}$ and $\{b, b, a, a\}$. La ED les otorga una disimilaridad de 4 cambios, es decir, 1 en el caso normalizado. Para estos registros, OSS-1 mejora claramente la

ED obteniendo un valor de disimilaridad de 0.125, el cuál entendemos que es más razonable puesto que los eventos son los mismos.

Una primera diferencia en los resultados del Cuadro comparativo 7.5 es el par de registros que consiguen la disimilaridad mínima para cada medida. La ED considera que los más similares son $\{e, d, a, b, c\}$ (id=6) y $\{f, e, d, a, b, c\}$ (id=7), porque contienen las secuencias más largas con un único cambio (la inserción de un nuevo símbolo). OSS-1 obtiene el mismo valor de disimilaridad para este par de registros (0,16), pero considera más similares las secuencias: $\{c, b, c, b, c, b\}$ (id=12) y $\{b, c, b, c, b, c\}$ (id=13), que son de la misma longitud y comparten exactamente los mismos símbolos en posiciones muy similares. De hecho, contienen la misma secuencia $\{c, b, c, b, c\}$, pero añadiendo el símbolo b antes o después de la misma.

Register		OSS-1			OSS-2			ED		
		MIN	Value	MAX	MIN	Value	MAX	MIN	Value	MAX
Id	Seq	Id	Value	Id	Id	Value	Id	Id	Value	Id
1	ab	3	0.2	9	3	0.33	2,10,11,12,13,14	3,4	0.33	2,9,10
2	bc	3	0.33	9,10	3	0.5	1,4,10,11,14	3	0.33	1,4,9,10,11
3	abc	1	0.2	9	1,5	0.33	10,11,14	5	0.25	9,10
4	cba	3	0.22	9	1	0.5	2,10,11,12,13,14	1	0.33	2,9
5	dabc	6	0.19	14	6	0.25	11,14	6	0.2	14
6	edabc	7	0.16	12,13	7	0.19	11,14	7	0.16	9
7	fedabc	6	0.16	12,13	6	0.19	11,14	6	0.16	9,12,14
8	cgdabc	5	0.35	14	6	0.39	11,14	5,6,7	0.33	14
9	d	10,11	0.33	1,2,3,4,12,13,14	-	-	-	10,11	0.5	1,2,3,4,12,13,14
10	da	5,9	0.33	2,12,13,14	5	0.5	1,2,3,4,11,12,13,14	5,9,11	0.5	1,2,3,12,13,14
11	db	9	0.33	13	all	1.0	all	1,5,9,10	0.5	2
12	cbcbcb	13	0.08	9,10	13	0.28	1,4,10,11,14	13	0.33	9,10
13	bcbcbcb	12	0.08	9,10	12	0.28	1,4,10,11,14	12	0.33	9,10
14	ebebeb	12	0.5	9,10	all	1.0	all	12	0.5	8,9,10

Cuadro 7.5: Comparación de los resultados

Si se considera la primera fila de las matrices, en la que se compara el registro $\{a, b\}$ (id=1) con el resto, se observa que el comportamiento de la ED es bastante diferente al de OSS-1. La ED considera que la secuencia $\{a, b\}$ (id=1) es igual de similar a las secuencias $\{a, b, c\}$ (id=3) y $\{c, a, b\}$ (id=4); mientras que la OSS-1 considera que la primera es más similar al registro $id = 3$ porque ambos registros comienzan su secuencia con el mismo evento a , seguido de b . La diferencia radica en que el registro $id = 1$ finaliza, mientras que el registro $id = 3$ contiene un evento más. Sin embargo, la secuencia $id = 4$ no comienza por el mismo evento. Esta diferencia debe

tenerse en cuenta si se desea valorar el orden relativo de los eventos.

Para realizar un análisis detallado del comportamiento de la función de similaridad en el contexto de los algoritmos de clasificación, interesa identificar las secuencias más próximas a una dada, puesto que son las que darán lugar a nuevos clusters. El Cuadro 7.5 muestra el identificador/es de los registros con mínima disimilaridad para cada uno de los 14 casos de estudio para las 3 medidas. También, el valor de disimilaridad que une a los pares de secuencias. Por último, también se muestra la secuencia con máxima disimilitud.

Finalmente, el análisis de la tabla sugiere que la función de disimilaridad OOS-1 es más precisa que la ED y OOS-2 para determinar los valores máximo y mínimo, ya que normalmente identifica valores únicos.

7.2.3. Método KSHC

En esta sección se demostrará el funcionamiento del método de clustering KSHC. Dadas las secuencias del Cuadro 7.6 se constata que existe una relación de similitud clara entre los 3 primeros elementos y los 6 últimos.

Id	Sequence	Id	Sequence
0	a b	5	y z x
1	b c	6	v y z x
2	a c b	7	w v y z x
3	x	8	s w v y z x
4	z x		

Cuadro 7.6: Ejemplo de secuencias de datos

Si se establece un valor de $k = 3$ en la creación de los clusters, exclusivamente se permitirá la creación de aquellos que contengan entre 3 y 5 elementos.

Si se pudieran representar las secuencias en un plano bidimensional, podrían distribuirse de una forma similar a la de la Figura 7.2.

En ella, se muestra el estado del método de clustering al finalizar el *Step1*. Se distinguen dos clusters claros: un Cluster 1 situado en la parte superior derecha que se correspondería con las 3 primeras secuencias y un Cluster 2 con los elementos más similares de entre los seis elementos situados en la parte inferior izquierda del plano.

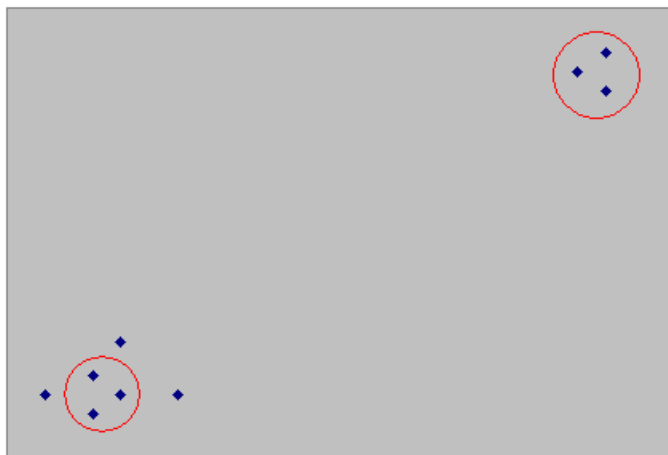


Figura 7.2: Centroides Válidos Step1

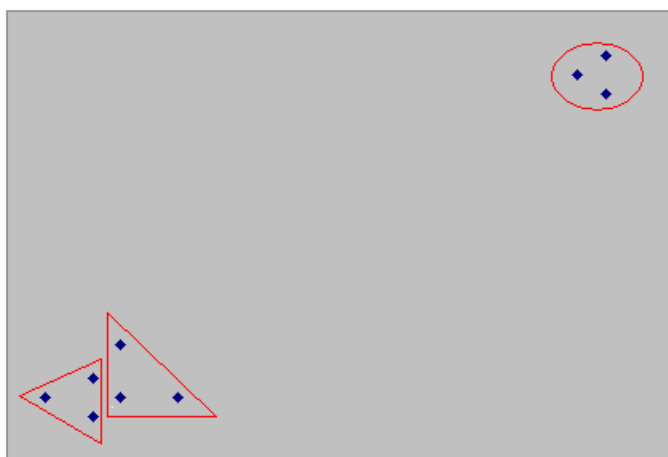


Figura 7.3: Centroides Válidos Step2. Resultado final

El resultado del *Step2* se muestra en la Figura 7.3. En la misma se demuestra que tal y como se ha diseñado el algoritmo y al alcanzar el Cluster 2 un tamaño de 5 elementos, en lugar de bloquearse y forzar la adhesión del elemento restante al Cluster 1, procede a ir cediéndole al elemento restante, los elementos pertenecientes al Cluster más similares al él, para que se pueda formar un nuevo cluster más homogéneo.

7.2.4. Método OESP

Las secuencias del Cuadro 7.7 se han utilizado para testear el método OESP propuesto en esta tesina.

Id	Sequence	Id	Sequence
0	a b c	7	d e f
1	a c	8	e d
2	a b	9	d e d e
3	c b a c	10	h f h f h f
4	c a b	11	f h f h f h
5	d e	12	g h g h f h
6	d e d	13	h f g f g h

Cuadro 7.7: Ejemplo de secuencias de datos

Para obtener los microclusters, se ha utilizado la medida de similaridad OSS-1 para comparar las secuencias y el algoritmo de clustering KSHC con un valor de $k = 3$. El resultado ha sido el siguiente

- Cluster $G_1 = \{abc, ac, ab, cbac, cab\}$ con 5 registros de diferentes longitudes.
- Cluster $G_2 = \{de, ded, def, ed, dede\}$ con 5 registros de diferentes longitudes.
- Cluster $G_3 = \{hfhfhf, fhfhfh, ghghfh, hfgfgh\}$ con 4 registros de la misma longitud.

En los siguientes cuadros (7.8, 7.9, 7.10) se muestran los prototipos creados con OESP para cada uno de los clusters y su similaridad con respecto a cada registro del cluster (calculado mediante la medida de similaridad OSS).

	<i>abc</i>	<i>ac</i>	<i>ab</i>	<i>cbac</i>	<i>cab</i>
<i>acb</i>	0.11	0.2	0.26	0.28	0.11

Cuadro 7.8: Similaridades del cluster G_1

	<i>de</i>	<i>ded</i>	<i>def</i>	<i>ed</i>	<i>dede</i>
<i>ded</i>	0.2	0.0	0.33	0.33	0.14

Cuadro 7.9: Similaridades del cluster G_2

	<i>hfhfhf</i>	<i>fhfhfh</i>	<i>ghghfh</i>	<i>hfgfgh</i>
hfhfgg	0.33	0.39	0.33	0.08

Cuadro 7.10: Similaridades del cluster G_3

Puesto que la microagregación óptima es la que minimiza el SSE (ver sección 4.1), se ha usado esta medida para evaluar los prototipos. Como no se encuentra normalizada, se ha escalado dividiendo por el número de grados de libertad, que se corresponde con el número de elementos del cluster menos 1:

$$SSE(G_i) = \frac{\sum_{j=1}^{|G_i|} (x_{ij} - c(G_i))(x_{ij} - c(G_i))}{|G_i| - 1} \quad (7.1)$$

A nivel global, se obtiene para cada cluster: $SSE(G_1) = 0,054$, $SSE(G_2) = 0,070$ and $SSE(G_3) = 0,127$. Siendo el error total de la partición P la media de los errores parciales, se obtiene que $SSE(P) = 0,0839$.

Estos resultados se han comparado con los prototipos obtenidos mediante el método PWM (presentado en la sección 5.3.2). En el caso de disponer de secuencias de diferentes longitudes, se debe añadir un símbolo en blanco al final de las secuencias cortas, hasta conseguir que todas las secuencias tengan la misma longitud. Como el método PWM no produce prototipos únicos, el Cuadro 7.11 muestra los valores de error (SSE) para cada posible prototipo. La última fila muestra la media del SSE obtenida en la aproximación del PWM.

$c(G1)$	SSE	$c(G2)$	SSE	$c(G3)$	SSE
aba	0.2381	ded	0.0706	hhghfh	0.2080
abb	0.2381	-	-	hfgghfh	0.0664
abc	0.0614	-	-	hhgffh	0.0887
ab	0.1432	-	-	hfgffh	0.1095
	0.1702		0.0706		0.1181

Cuadro 7.11: Results obtained with the PWM method

Partiendo de los valores contenidos en la tabla, se puede concluir que:

- En el cluste G_1 , todos los prototipos obtenidos con el método PWM son peores que el obtenido con el método OESP, que consigue un error de 0.054.
- En el cluster G_2 , PWM proporciona un prototipo único, que coincide

con el obtenido por el método OESP.

- En el cluster $G3$, tres de los prototipos obtenidos con el método PWM son mejores que el obtenido con método OESP, que consigue un error de 0.127, y el primero es peor. PWM no proporciona ningún mecanismo para distinguirlos.

Después de analizar los resultados del método OESP para eventos individuales, se evalúa para patrones de longitud de dos elementos. El Cuadro 7.12 muestra los resultados obtenidos en este caso.

$c(G1)$	SSE	$c(G2)$	SSE	$c(G3)$	SEE
abc	0.0605	ded	0.0706	hfhfhg	0.0827

Cuadro 7.12: Resultados obtenidos con OESP sobre patrones de dos eventos

Frente a los resultados obtenidos anteriormente, éstos son mejores al método PWM y OESP sobre elementos individuales. Principalmente se consigue una mejora en el cluster G_3 , donde se reduce el error a 0,0827.

En resumen, el método OESP sobre patrones consigue un error global de solamente 0,0723 que comparado con el obtenido con elementos individuales (0.0839), constata que la información de dos eventos consecutivos mejora la construcción de prototipos, porque mantiene las relaciones secuenciales.

7.2.5. Estudio de rutas turísticas

En las secciones anteriores se han descrito pruebas para cada una de las etapas que conforman la técnica de microagregación (función de similitud, construcción de clusters, definición de prototipos).

En este apartado se describe un análisis global con un conjunto de registros de mayor tamaño (Cuadro 7.13). Estos datos se corresponden con las rutas turísticas reales realizadas por un total de 39 visitantes de una ciudad asiática. Cada evento (carácter) de la secuencia se corresponde con una zona de la ciudad.

IdReg.	Secuencia
0	a b h e s x z c m c m d z
1	a h e s x z c m d z x s e o e b w a
2	a b j b j a h e s x z d z r x e h a
3	a b a b h e x z c m d n s e k b w a
4	a h e s x z c m d z x h b w a
5	a h b j b e h e s r z c m d z r s e k b w a
6	a h e s x z c z p v p e h b w a
7	a b a h e x z c c m d r n r e k e w a
8	a b a b j b w b k o g e s g s r n r z c m d z r n s e h b w a
9	a h e r z p x e o k e b a
10	a b e g e h e b w a
11	a b j e o e s n d z c m d z x s e k b w a
12	a h e b e o e s r z c z s e k b w a
13	a h e h a b w a
14	a h e k b w a
15	a b a h e x z c m d r x e h b e b w a
16	a h b a h e s x z c m d z x e h a b w a
17	a w b j b a h e s r z r s r d m c m c z s g s e k b j h v p x r n d n s o k w a
18	a h b h e s x z c m d r n s o k w b w a
19	a h e s x z y z r s e h e g o e o e k e b w b e b w a
20	a b a h e s x z d m z c m d z y p x h a b w a
21	a h e h e s r x z r x p x e k b w a w
22	a b a b h e s x z c m d z x s o k e b w a
23	a h e z c m d r s o k e b w a
24	a b a h p f p z c r s n g s e h e b w a
25	a b j b e k b a w a w k e h x z y p x h a b w a
26	a h e s x z r n s r x e h a b w a
27	a b j b e h x z c m d r n d n d n s o k e b w a
28	a b e k e s x z c m d s e k w a
29	a b a h a b e g s r x z c m d r s x h p v a b w a
30	a b a e o e b w a b e s x z r d z c m d r x e k b w a
31	a b j a h e s z p v a b w a
32	a b e s r n r z c m d z x s d g e k b w a
33	a h s x z m z c m d r s e w a
34	a b a h e x z c m d z x s e k b a
35	a b a h e s x z c m d n d n s g e b w a
36	a b a e s x z c m d n s o k w a
37	a b k e b a h e s r z p v h e k b w a
38	a b a h a b k e s x z c m d n s o k w a

Cuadro 7.13: Registros de rutas turísticas

En el Cuadro 7.14 se presenta un resumen de los resultados obtenidos. Para cada fila, se indica el tipo de similaridad que se ha aplicado para crear la tabla de similaridad inicial, cómo se ha creado el prototipo, el tipo de linkado en la creación de los clusters y según los resultados obtenidos, cuál ha sido el error de sumas cuadráticas (SSE) global inferior y con qué valor de k se ha producido. Indicar que las pruebas se han realizado para valores de k entre 4 y 7.

Tipo Similaridad	Tipo Centroide	Tipo Linkage	SSE	k
Individual	Individual	Single	0.0861	4
Individual	Individual	Complete	0.0764	4
Individual	Sequences of length 2	Single	0.1460	5
Individual	Sequences of length 2	Complete	0.1782	4
Sequences of length 2	Individual	Single	0.7368	5
Sequences of length 2	Individual	Complete	0.6988	4
Sequences of length 2	Sequences of length 2	Single	0.2920	5
Sequences of length 2	Sequences of length 2	Complete	0.2826	4

Cuadro 7.14: Resultado del análisis de los registros de rutas turísticas

Analizando la tabla, se obtienen las siguientes conclusiones para este estudio:

- Los mejores resultados para cada tipo de similaridad son los que se obtienen considerando el mismo tipo de datos en la creación de centroides. Es decir, si se evalúa la similaridad en base a elementos individuales, se obtiene un error inferior si el cálculo del prototipo también se realiza en base a elementos individuales. E igualmente, también para secuencias de elementos de longitud 2.
- Para todos los resultados que se han realizado en esta prueba se consigue un error inferior estableciendo la misma k si se basa la fusión de los elementos del cluster en un linkado *Complete*. Es decir, si en la construcción del cluster se evalúan las fusiones en base al valor máximo de disimilaridad entre los elementos, es probable que los elementos esten más cohexionados y se consiga formar un cluster que contenga elementos más similares y, por tanto, en la construcción de su centroide, éste se conseguirá que sea más representativo porque el grupo es más homogéneo.
- En general se consiguen mejores resultados con una k inferior.
- Para este ejemplo, el mejor resultado se obtiene con una similaridad y creación del centroide de tipo *Individual* (valor 0.0764). Por este

motivo, se van a analizar los clusters resultantes para esta clasificación.

En el Cuadro 7.15 se muestra el resultado de la microagregación para el tipo de similaridad *Individual*, el tipo de centroide *Individual* y el tipo de linkage *Complete*.

Id	SSE	Nº	Centroide	Elementos
1	0.0728	4	abebrszwacjmdnkrhsegx bzdonrsb	ababjbwbkogesgrnrzcmdzrnsehbwa awbjbahesrzsrmdmcmczsgsekbjvhvpxrmdnsokwa abesrnrzcmdzxsdgekbwa abjbehxzcmdrmdndnsokwbwa
2	0.0431	4	ahsxexcmdzbxwxs	abhesxzcmcmdz ahsxzmzcmdrsewa ahesxzcndzxseobwa ahesxzcndzxhbwa
3	0.0528	4	aebehszckrmdewsoab	ahbjbehesrzcmdzrsekbwa ahebeoesrzczsekbwa ahezcmdrsokebwa abekesxzcmdsekwa
4	0.0932	4	abaebhazpswjkbhreaxk	abjbekbawawkehxyzpxhabwa abjaheszpabwa abkebahesrzpvhekbwa ababahbegsrzcmdrsxhpvabwa
5	0.0786	5	abezxacdmehwrsobenkah	ahbhesxzcmdrnsokwbwa abjeoesndzcmdzxsekbwa abaeobwabesxzdzcndrxekbwa abahexzcndrnrekewa abahexzcndrxehbebwa
6	0.0433	6	abaezhcxbmdnaswkeo	ababhexzcndnsekbwa ababhesxzcmdzxsokebwa abaesxzcmdnsokwa ababahkesxzcmdnsokwa abahpfpczrsngsehebwa abahesxzcmdndnsgebwa
7	0.0980	5	ahexzesrhbapwekzsr	ahesxyzrsehegoeokebwbebw aherzpxeokeba ahesxzczipvehbwa ahehesrxzrxpkekbwaw ahesxzrnsrxeabwa
8	0.1295	7	abheaxwszbdhzae	abjbjahesxzdzrxeha ahbahesxzcmdzxehabwa abahexzcndzxsekba abahesxzdmcndzypxhabwa abegehebwa ahehabwa ahkebwa

Cuadro 7.15: Resultado de la microagregación con mínimo SSE

Para poder contrastar como se calculan los centroides mediante el tipo de *Sequences of length 2* se muestra el Cuadro 7.16 coincidente con los clusters mostrados en el Cuadro anterior.

Id	SSE	Nº	Centroide	Elementos
1	0.1873	4	cmdzczrsrzzsgesehxs okbjbwab	ababjbwbkogesgsrnrzcmdzrnsehbwa awbjbahestrzrsrdmcmczsgsekbjhpvrndnsokwa abesrnrzcmdzxsdgekbwa abjbehxzcmdrndndnsokebwa
2	0.1452	4	szxcmzxsahew	abhesxzcmcmdz ahsxzmzcmdrsewa ahesxzcmdzxseobwa ahesxzcmdzxhbwa
3	0.2911	4	zcmdrzsersxzrahek	ahbjbehesrzcmdzrsekbwa ahebeoesrzczsekbwa ahezcmdrsokebwa abekesxzcmdsekwa
4	0.1202	4	abahesrzpvabwawkekbjb	abjbekbawawkehxyzpxhabwa abjahesrzpvabwa abkebahesrzpvhekbwa abababegsrzcmdrsxhvpabwa
5	0.3339	5	zcmdrndzxrxesxsnoek	ahbhesxzcmdrnsokwbwa abjeoesndzcmdzxsekbwa abaeobwabesxrdzcmdrxekbwa abahexzccmdrnrekewa abahexzcmdrxehbebwa
6	0.0368	6	abahesxzcmdnsokebwa	ababhexzcmdnsekbwa ababhesxzcmdzxsokwbwa abaesxzcmdnsokwa abababkesxzcmdnsokwa ababpfpzcrsngsehebwa abahesxzcmdndnsgebwa
7	0.1031	5	ahesxrxheokebwab	ahesxyzrsehegoeokebwbebwa aherzpxeokeba ahesxzczpvehbwa ahehesrxzrxpxekbwaw ahesxzrsrxehabwa
8	0.2083	7	hehabwbahxzcm	abjbjahesxzdrxcha ahbahesxzcmdzxehabwa abahexzcmdzxsekba abahesxzdmzcmdzypxhabwa abegehebwa ahehabwa ahkbwa

Cuadro 7.16: Resultado de la microagregación con mínimo SSE - Seq.2

Para el primer cuadro, en términos generales, se puede decir, que los centroides propuestos para cada uno de los clusters son aproximadamente representativos de los elementos contenidos. Mientras que en el segundo cuadro, destaca exclusivamente la corrección del centroide del cluster $Id = 6$.

Capítulo 8

Conclusiones

La elaboración de esta tesina ha sido un reto importante por tratarse de un tema nuevo para mí.

Inicialmente, los conocimientos en el área eran escasos, focalizados en las diferentes técnicas de agrupación de datos y desde una visión más general. Tras una etapa inicial de toma de contacto y formación en el estado del arte de las técnicas actuales de clustering y su uso en la protección del riesgo de revelación estadística, se han podido realizar nuevas aportaciones en este área. Tal y como se ha demostrado en este documento, los datos categóricos presentan una mayor complejidad que los datos numéricos y su protección se encuentra poco estudiada, lo que ha supuesto un reto motivador y de gran interés para mí.

Una vez finalizado el trabajo, se puede decir que los objetivos planteados al inicio (sección 1.3) se han cumplido. Los propósitos incluían un estudio detallado de la literatura relacionada con los métodos de clustering y funciones de similitud que se ha presentado en sección 2, un estudio de los métodos de enmascaramiento mediante microagregación que se detalla en las secciones 3 y 4, y la propuesta de un nuevo algoritmo de microagregación para datos no numéricos que se ha introducido en las secciones 5-7. En relación con este último objetivo de diseño un algoritmo, las contribuciones realizadas en esta tesina han sido las siguientes:

- Definición de una nueva medida de similaridad para datos categóricos, denominada Ordering-based Sequence Similarity(OSS). Con la propuesta y los resultados de esta medida se ha realizado un artículo que ha sido aceptado en el congreso internacional MDAI 2008(“Modeling Decisions for Artificial Intelligence”[21]) y se dispondrá de una publicación en LNAI [14].

- Descripción de un nuevo algoritmo de clustering para la microagregación de datos para tamaños de clusters entre k y $2k-1$, denominado K-sized Hierarchical Clustering (KSHC). Se analizaron diferentes técnicas de microagregación, pero como se ha descrito, estaban orientadas hacia la información numérica. Fue necesario disponer de un algoritmo que tomase como base una matriz de similaridad/disimilitud para considerar la relación entre los registros y no precisar del cálculo de centroides durante el proceso de clustering.
- Definición de un método de cálculo de prototipos (o centroides) para datos categóricos. Esta aportación se presentará como artículo (actualmente en preparación) en el workshop internacional PSAI 2009 (“Privacy and Security by means of Artificial Intelligence” [29]) a finales del mes de Septiembre.

Al margen de los resultados expuestos en este documento, la realización de esta tesina me ha permitido conocer el ámbito de la investigación y sus etapas: el estudio en profundidad del estado del arte, el análisis de la dificultad de protección de los datos categóricos, la reflexión en nuevos algoritmos de resolución y la evaluación de los resultados y mejoras.

Como valoración final, destaco la gran aportación a nivel profesional y personal que ha supuesto la consecución de los objetivos propuestos, considerando especialmente relevante la publicación de (al menos) un artículo en un congreso internacional y la realización de una aplicación de escritorio que muestra el resultado de aplicar los nuevos métodos enunciados.

Capítulo 9

Trabajo futuro

Durante el desarrollo de esta tesina, y una vez finalizada la misma, se han planteado las posibles líneas de trabajo futuras:

- Inclusión de información temporal en las secuencias de datos. En esta tesina se ha trabajado con secuencias atemporales, es decir, en las que no se considera la duración de sus eventos. Sin embargo, disponemos de datos con secuencias de eventos y su duración. Sería interesante adaptar los algoritmos propuestos para hacer uso de esta información adicional.
- Modificación del método OESP de forma que garantice que el resultado final sea tanto representativo, como no idéntico a ninguno de los elementos que conforman el cluster. Este problema consideramos que tiene su origen en la existencia de elementos que se encuentran integrados en otros, es decir, subsecuencias contenidas en otras secuencias.
- Estudio de nuevos métodos de clustering recientes más eficientes temporalmente para la microagregación. Estos métodos permiten trabajar con volúmenes de datos muy elevados. Se debe estudiar si es posible aplicarlos al caso de secuencias categóricas y evaluar la calidad de sus resultados.

Además, como ya se ha comentado con anterioridad, esta tesina se sitúa dentro de los proyectos ARES y E-AEGIS. Estos proyectos, que tienen prevista su finalización en los años 2010 y 2012, respectivamente, orientan parte de sus estudios a la protección de bases de datos, donde se situaría esta tesina, marcándose como objetivo su futura aplicación en data warehouses y bases de datos públicas.

Bibliografía

- [1] O. Abul, M. Atzori, F. Bonchi, and F. Giannotti. Hiding sequences. In *ICDE Workshops*, pages 147–156. IEEE Computer Society, 2007.
- [2] A. Asuncion and D. Newman. UCI machine learning repository (<http://archive.ics.uci.edu/ml/>), 2007.
- [3] G. Beliakov, A. Pradera, and T. Calvo. *Aggregation Functions: A Guide for Practitioners*, volume 221 of *Studies in Fuzziness and Soft Computing*. Springer, 2007.
- [4] T. G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30, London, UK, 2002. Springer-Verlag.
- [5] J. Domingo-Ferrer. Microaggregation for database and location privacy. In *Next Generation Information Technologies and Systems*, pages 106–116, 2006.
- [6] J. Domingo-Ferrer, A. Martínez-Ballesté, J. M. Mateo-Sanz, and F. Sebé. Efficient multivariate data-oriented microaggregation. *The VLDB Journal*, 15(4):355–369, 2006.
- [7] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Trans. on Knowl. and Data Eng.*, 14(1):189–201, 2002.
- [8] J. Domingo-Ferrer, F. Sebé, and A. Solanas. A polynomial-time approximation to optimal multivariate microaggregation. *Comput. Math. Appl.*, 55(4):714–732, 2008.
- [9] J. Domingo-Ferrer and V. Torra. *Confidentiality, disclosure, and data access : Theory and practical applications for statistical agencies*, chapter A quantitative comparison of disclosure control methods for micro-data, pages 111–133. Elsevier, 2001.

- [10] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Min. Knowl. Discov.*, 11(2):195–212, 2005.
- [11] G. Dong and J. Pei. *Sequence Data Mining*, volume 33 of *Advances in Database Systems*. Springer US, 2007.
- [12] B. Everitt. *Cluster Analysis*. Social Science Research Council by Heinemann Educational Books, 1974.
- [13] K. Gibert and R. Nonell. Impact of mixed metrics on clustering. In *Progress in Pattern Recognition, Speech and Image Analysis*, pages 464–471, 2003.
- [14] C. Gómez-Alonso and A. Valls. A similarity measure for sequences of categorical data based on the ordering of common elements. In *Lecture Notes in Artificial Intelligence*, volume 5285, pages 134–145, 2008.
- [15] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2nd edition, 2006.
- [16] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall Advanced Series. Prentice Hall, 1988.
- [17] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [18] Y. Li, S. Zhu, L. Wang, and S. Jajodia. A privacy-enhanced microaggregation method. In *Foundations of Information and Knowledge Systems*, pages 148–159, 2002.
- [19] T. W. Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874, November 2005.
- [20] J. M. Mateo-Sanz and J. Domingo-Ferrer. A comparative study of microaggregation methods. *Questiio: Quaderns d’Estadística, Sistemes, Informàtica i Investigació Operativa*, 22(3), 1998.
- [21] Modeling Decisions for Artificial Intelligence. <http://www.mdai.cat/>, 2008.
- [22] D. W. Mount. *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press, September 2004.
- [23] E. Myers, S. F. Altschul, W. Gish, E. W. Miller, and D. J. Lipman. Blast, basic local alignment search tool (<http://www.ncbi.nlm.nih.gov/blast/blast.cgi>), 2007.

- [24] J. Nin and V. Torra. Extending microaggregation procedures for time series protection. In e. a. Salvatore Greco, editor, *Rough Sets and Current Trends in Computing*, volume 4259 of *LNCS*, pages 899–908. Springer, 2006.
- [25] C. Notredame. Recent evolutions of multiple sequence alignment algorithms. *PLoS Computational Biology*, 3(8):e123, August 2007.
- [26] A. Oganian and J. Domingo-Ferrer. The complexity of optimal microaggregation for statistical disclosure control. *Statistical Journal of the United Nations Economic Commission for Europe*, 18:4:345–354, 2001.
- [27] J. H. Orallo, C. F. Ramírez, and M. R. Quintana. *Introducción a la Minería de Datos*. Prentice Hall / Addison-Wesley, 2004.
- [28] C. Pérez and D. Santín. *Minería de Datos. Técnicas y Herramientas*. Paraninfo, 2007.
- [29] Privacy and Security by means of Artificial Intelligence. <http://crises-deim.urv.cat/psai/>, 2009.
- [30] A. Solanas and A. Martínez-Ballesté. V-mdav: a multivariate microaggregation with variable group size. In A. Rizzi and M. Vichi, editors, *Proceedings in Computational Statistics COMPSTAT 2006*, Heidelberg: Springer’s Physica Verlag, pages 917–925, 2006.
- [31] F. C. Statistical-Methodology. Report on statistical disclosure limitation methodology. Technical report, Statistical and Science Policy, Office of Information and Regulatory Affairs, Office of Management and Budget, 2005.
- [32] V. Torra. Microaggregation for categorical variables: A median based approach. In S. B. . Heidelberg, editor, *Privacy in Statistical Databases*, volume 3050/2004 of *Lecture Notes in Computer Science*, pages 162–174, 2004.
- [33] V. Torra and J. Domingo-Ferrer. *Confidentiality, disclosure, and data access : Theory and practical applications for statistical agencies*, chapter Disclosure control methods and information loss for microdata, pages 91–110. Elsevier, 2001.
- [34] U. Virginia. Fasta sequence comparison (<http://fasta.bioch.virginia.edu>), 2007.
- [35] I. M. Wallace, G. Blackshields, and D. G. Higgins. Multiple sequence alignments. *Current Opinion in Structural Biology*, 15(3):261–266, 2005.

- [36] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.
- [37] J. Yang and W. Wang. Cluseq: efficient and effective sequence clustering, 2003.