

# El algoritmo de microagregación KSHC para anonimización de secuencias de datos categóricos

Aida Valls, Cristina Gómez-Alonso

ITAKA Research Group - Intelligent Tech. for Advanced Knowledge Acquisition

Department of Computer Science and Mathematics

Universitat Rovira i Virgili

43007 Tarragona, Catalonia, Spain

Email: aida.valls@urv.cat

## 1. Introducción

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 [2, 10]. 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 [7]: 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.

En [14] 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.

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. 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 [12].

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 [14].

El número total de *vroots* disponibles es de 294, y el número medio de visitas a cada *vroot* por usuario es de 3.

Las secuencias de eventos que se van a considerar se caracterizan por:

- 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 univariadas, es decir, solamente se estudia un concepto.
- Las longitudes de las secuencias pueden ser diferentes.
- Los eventos se pueden repetir en la secuencia (por ejemplo, un turista puede visitar una zona más de una vez durante sus vacaciones).

El objetivo de este trabajo es el diseño e implementación de un algoritmo clustering que pueda ser usado para la protección de secuencias de eventos. El uso de técnicas de clustering para generar una versión protegida de los datos se conoce como *microagregación* [4]. La **microagregación** es una técnica de control de revelación estadística para microdatos <sup>1</sup> que sigue una tendencia perturbativa/sustitutiva. Su procedimiento general consiste en:

- Construcción de clusters con los datos originales según un criterio de máxima similaridad (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.

Dadas las características intrínsecas de las secuencias categóricas, los algoritmos clásicos de microagregación no son aplicables y por ello se ha estudiado un nuevo método. En este documento se propone una solución para la primera etapa, la construcción de clusters. En [20] se propone una solución para la segunda etapa, la construcción de prototipos, proceso que también requiere el diseño de nuevos algoritmos.

Este documento se estructura de la siguiente forma: en la sección 2 se revisan las técnicas tradicionales de clasificación automática (data mining), en la sección 3 se explican y evalúan los métodos tradicionales de microagregación para protección de datos, finalmente en la sección 4 se propone un nuevo algoritmo de microagregación adaptado a los requerimientos de los datos categóricos secuenciales, que hemos denominado KSHC.

---

<sup>1</sup>Ficheros donde cada registro contiene información sobre un individuo (ciudadano o compañía)

## 2. Técnicas de clasificación automática

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.

Los métodos de análisis de clusters han sido estudiados desde hace muchos años [8, 11, 21]. 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.

## 2.1. Clustering 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

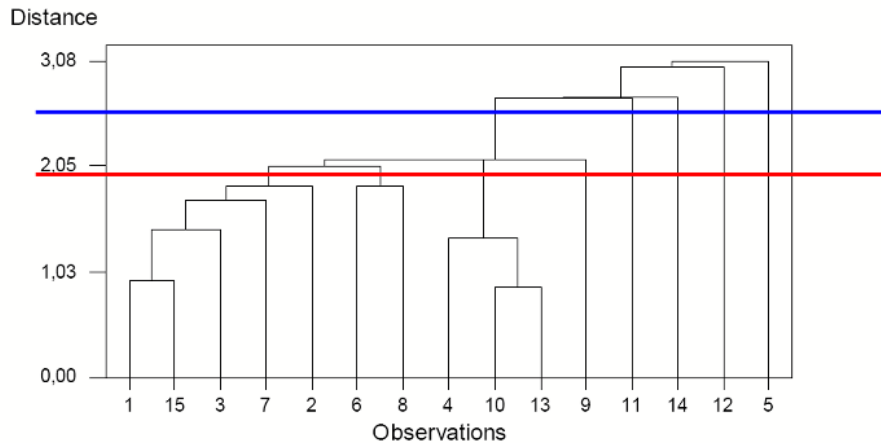


Figura 1: Ejemplo de un dendograma

visualmente la composición de los clusters en etapas intermedias. Observando la Figura 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).

## 2.2. Clustering 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 conocidos son el *k-Means*. Este método 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).

### 2.3. Otras técnicas de clustering

Al margen de las técnicas de jerárquicas y partitivas, existen otras técnicas destacables [21]:

- **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-Organizados de Características) y *ART* (Adaptative 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.

## 3. Técnicas de microagregación

El concepto de microagregación en SDC toma como base la técnica de clustering, pero con la 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.

Como ya se ha dicho en la introducción la **microagregación** es una técnica de control de revelación estadística para microdatos que sigue una tendencia perturbativa/sustitutiva. En [6] se afirma la adecuación de la microagregación para salvaguardar la  $k$ -Anonymity resultando más adecuada que otros métodos perturbativos de protección. Los algoritmos de microagregación constan de los siguientes pasos:

- 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.

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 ambos casos, es necesario definir una medida de similitud entre elementos, que se tomará como base para determinar la partición y para generar un prototipo similar a los miembros del cluster. En [9] se explican las distintas medidas de similitud para secuencias de valores numéricos y categóricos. En este trabajo se detectó que las medidas existentes para el caso de valores categóricos presentaban algunos inconvenientes, por lo cual se definió una nueva medida llamada *Ordered-based Sequence Similarity* (OSS).

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 (*SSE*: Sum of Squared Error) para cada objeto  $x_{ij}$  a los centroides sea minimizada (donde  $G_i$  es el grupo al cual  $x_{ij}$  pertenece).

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

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 [17], 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.

### 3.1. Tipos

En [15] 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.

#### 3.1.1. 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 [4, 15] 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 microaggregation** y consigue una agrupación más homogénea de los datos y, por tanto, una pérdida de información menor.

### 3.1.2. 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.

## 3.2. 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.

### 3.2.1. 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:

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

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

$$d(G_i, G_j) = \frac{n_i n_j}{n_i + n_j} (\bar{x}_i - \bar{x}_j)^2 \quad (3)$$

donde  $\bar{x}_i$  es la media de los elementos de  $G_i$  y  $\bar{x}_j$  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 [15, 4], 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 [13] 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).

### 3.2.2. Algoritmo MDAV

El algoritmo de MDAV (Maximum Distance to Average Vector)[4] 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 [18] 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 [6] 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.

En [16] 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.

**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

### 3.2.3. Otros algoritmos

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

- **MHM algorithm** (Multivariate version of the Hansen-Mukherjee algorithm) [3] (2006): intenta optimizar el cálculo de la microagregación minimizando la pérdida de información.
- **$\mu$ -Approx algorithm** [5] (2008): aproximación a la microagregación óptima basado en la descomposición de grafos.
- **k-Means variation** [19] (2004): una modificación del algoritmo de k-means para datos categóricos.

## 4. El método KSHC: *k-sized Hierarchical Clustering*

En la sección anterior se han presentado los algoritmos para microagregación *k*-Ward y MDAV. 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 de datos secuenciales. 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 varianza 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 varianza 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, 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 tradicionales presentados en la sección 2, 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 utilizados en minería de datos, y también porque su utilización para datos secuenciales ya ha sido propuesta por otros autores [7]. 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 nuevo algoritmo de microagregación 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 1.** *Se denomina Cluster Válido a aquel que tiene entre  $k$  y  $2k - 1$  elementos.*

**Definición 2.** *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:*

$$minVC = \left\lceil \frac{n}{(2k - 1)} \right\rceil \quad (4)$$

$$maxVC = \left\lfloor \frac{n}{k} \right\rfloor \quad (5)$$

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), 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  $N$ .

A continuación se muestra el pseudo-código del algoritmo que se ha diseñado 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

**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**

**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 Si**

**Fin Mientras**

**FAlgoritmo**

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

## 4.1. Ejemplos

En esta sección se mostrará el funcionamiento del método de clustering KSHC con un par de ejemplos. En ambos casos se ha usado la medida de similitud OSS (Ordered-based Sequence Similarity) para comparar pares de secuencias de eventos. Tal como se ha dicho en la sección 3 esta medida ha sido especialmente diseñada para manejar secuencias de datos categóricos, y ofrece mejores resultados que la medidas clásicas como la distancia de Hamming o la de Levenstein [9].

## 4.2. Ejemplo 1

El primer ejemplo utiliza los datos mostrados en la Tabla 1. El primer paso del algoritmo de clustering consiste en construir la matriz de disimilitudes entre pares de registros. Después de aplicar la medida OSS el resultado es el que se muestra en la Tabla 4.2). A continuación se procede a realizar la microagregación de elementos con tamaño variable  $k = 3$ , usando la técnica de agrupación Single Linkage.

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 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 2: Matriz de disimilaridad al inicio del clustering

Al finalizar el *Step 1* se obtienen 3 clusters, quedando el elemento 0 sin fusionar.

- Cluster 9: 1,4 (Inválido porque no tiene un mínimo de 3 elementos)
- Cluster 11: 3,5,7 (Válido)
- Cluster 13: 2,6,8 (Válido)

En el *Step 2*, se consideran los Clusters 11, 12 y 13 y el elemento 0, como elementos a clasificar. En este punto, la matriz de disimilitud obtenida tras aplicar el Single Linkage es la que se muestra en la Tabla 4.2. La matriz de disimilitud tiene el mínimo en 0.2, lo cuál indica que se debería fusionar el

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 y por tanto se realiza. En este punto, el método finaliza porque no quedan elementos sin agregar a los clusters, y ya se han obtenido 3 clusters válidos.

Id.Reg.	0	C9	C11	C13
0	0.0	0.46	1.0	0.75
C9	0.46	0.0	0.75	0.75
C11	1.0	0.75	0.0	0.20
C13	0.79	0.75	0.20	0.0

Cuadro 3: Matriz de disimilaridad al finalizar el Step1

### 4.3. Ejemplo2

Supongamos que tenemos 8 secuencias, que se pueden mapear en un espacio bi-dimensional de acuerdo con su similitud, tal y como se muestra en la Figura 2.

Aplicaremos el algoritmo de microagregación KSHC estableciendo un valor de  $k = 3$  en la creación de los clusters, por lo que exclusivamente se permitirá la creación de aquellos que contengan entre 3 y 5 elementos.

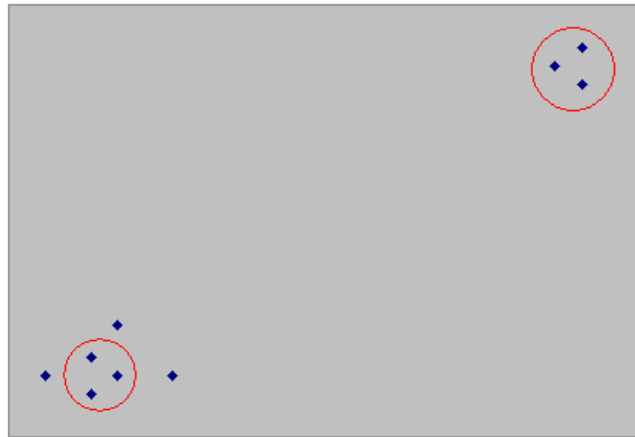


Figura 2: Centroides Válidos Step1

En la Figura 2, se muestra el estado del 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.

El resultado del *Step2* se muestra en la Figura 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.

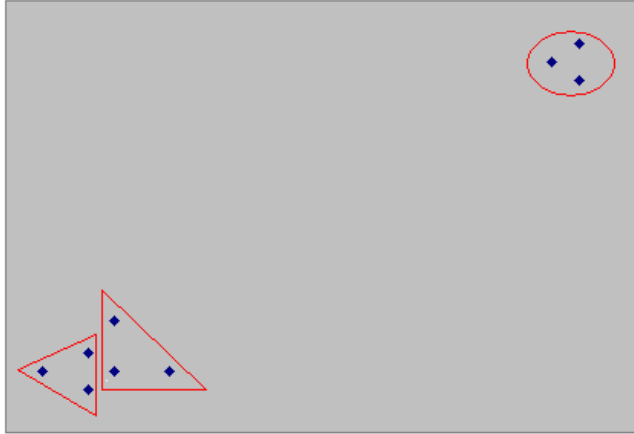


Figura 3: Centroides Válidos Step2. Resultado final

## 5. Conclusiones y líneas futuras

Tal y como se ha demostrado en este documento, los datos secuenciales presentan una mayor complejidad que los datos estáticos tradicionales y su protección se encuentra poco estudiada, lo que supone una línea de investigación muy interesante.

En este documento se ha presentado el estado del arte en técnicas de clustering de minería de datos y en algoritmos de microagregación para la protección de la privacidad. Una vez estudiada la literatura, se ha detectado la necesidad de desarrollar un nuevo enfoque adecuado para el caso de datos secuenciales categóricos. En el trabajo se describe un nuevo algoritmo de microagregación de tamaño variable entre  $k$  y  $2k-1$ , denominado K-sized Hierarchical Clustering (KSHC). El método se ha probado con datos sintéticos.

Las líneas de investigación que se derivan de este trabajo se pueden resumir en:

- Realizar una evaluación más exhaustiva del funcionamiento del método propuesto.
- Estudiar la inclusión de información temporal en las secuencias de datos. Hasta ahora 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.
- Definir de un método de cálculo de prototipos (o centroides) para secuencias de eventos. Este método permitirá finalizar el proceso de protección de una base de datos, puesto que la última fase consiste en sustituir los elementos de cada cluster por su prototipo.

## 6. Agradecimientos

Este trabajo está financiado por el Ministerio Español con los proyectos ARES (CONSOLIDER INGENIO 2010 CSD2007-00004) y eAEGIS (TSI2007-65406-C03-02) y por la Universidad Rovira i Virgili con el proyecto 2008TURISME-02.



## Referencias

- [1] O. Abul, M. Atzori, F. Bonchi, and F. Giannotti. Hiding sequences. In *ICDE Workshops*, pages 147–156. IEEE Computer Society, 2007.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] J. Domingo-Ferrer, F. Sebé, and A. Solanas. A polynomial-time approximation to optimal multivariate microaggregation. *Comput. Math. Appl.*, 55(4):714–732, 2008.
- [6] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Min. Knowl. Discov.*, 11(2):195–212, 2005.
- [7] G. Dong and J. Pei. *Sequence Data Mining*, volume 33 of *Advances in Database Systems*. Springer US, 2007.
- [8] B. Everitt. *Cluster Analysis*. Social Science Research Council by Heinemann Educational Books, 1974.
- [9] 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.
- [10] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2nd edition, 2006.
- [11] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall Advanced Series. Prentice Hall, 1988.
- [12] A. K. Jain, M.Ñ. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [13] 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.
- [14] T. W. Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874, November 2005.
- [15] 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.
- [16] J.Ñin 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.

- [17] 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.
- [18] 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.
- [19] 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.
- [20] A. Valls, C. Gómez-Alonso, and V. Torra. Generation of prototypes for masking sequences of events. In *4th. Int. Conference on Availability, Reliability and Security*, 2009 (in press).
- [21] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.