



UNIVERSIDAD DEL BÍO-BÍO, CHILE

FACULTAD DE CIENCIAS EMPRESARIALES

Departamento de Sistemas de Información

BÚSQUEDA DINÁMICA DE PATRONES SOBRE TRAYECTORIAS

TESIS PRESENTADA POR LUIS EMILIO CABRERA CROT
PARA OBTENER EL GRADO DE MAGÍSTER EN CIENCIAS DE LA COMPUTACIÓN
DIRIGIDA POR DRA. MÓNICA CANIUPÁN Y DRA. ANDREA RODRÍGUEZ

2016

Agradecimientos

En primer lugar, quiero dar gracias a mi profesora guía en esta tesis, Dra. Mónica Caniupán, ya que con su apoyo, paciencia y guía pude llevarla a cabo. Gracias también a mi co-guía, Dra. Andrea Rodríguez y al profesor de la Universidad de Concepción Dr. Diego Seco, quienes siempre tuvieron la voluntad y disposición de aclarar las dudas que iban surgiendo durante su desarrollo.

También agradezco a mis compañeros de Magíster Juan José, Raúl, Cristian, Jorge y Joel, quienes hicieron más grato el ir a la oficina a trabajar.

A mi familia, en especial a mi madre, mi padre, mi hermana y sobrino, muchas gracias por el ánimo que me brindaron, por la paciencia que me tuvieron y por la confianza que depositan en mí cada día. A Yohana, gracias por ser mi compañera, por tu amor y compañía, y por sobre todo, gracias por cada vez que dices “Tú puedes”.

A mi otra familia, Acción Sin Límites, gracias por levantar mi ánimo cada vez que lo necesitaba, sin ustedes esto no hubiera sido lo mismo.

Y en especial, muchas gracias a todas las personas que de alguna u otra manera, aportaron en que esta tesis se haya vuelto realidad.

Abstract

Nowadays, there are several mobile data sources such as wireless networks, GPS, etc., that allow to collect geographical locations of objects steadily, generating big collections of data. This data can be used to generate object trajectories. A data trajectory is a sequence of line segments that represent the steadily movement of an object between two time instants.

Data mining applications allow to find similar behavior among object trajectories, which is known as object trajectories patterns. These patterns are useful to take urban planning decisions, to analyze traffic congestion, to understand animal migration, to study the behavior of natural phenomena like hurricanes, twisters, etc.

In this thesis we present a new concept of *aggregated trajectory* and also present algorithms to obtain aggregated trajectories dynamically. An *aggregated trajectory* is a new trajectory that is obtained from a set of trajectories with similar behavior. An aggregated trajectory shows, in addition to its behavior pattern, the density of that pattern. The process to obtain an aggregated trajectory consists of two phases: (i) clusters generation, and (ii) aggregation of similar clusters.

In this thesis we propose a process to obtain clusters that is able of capturing relationship between sub-segments of trajectories. Moreover, the algorithms obtain trajectory dynamically, unlike other algorithms in literature. In other words, the process of obtaining aggregated trajectories is adapted to receive new trajectories, in such a way, that it is not necessary to process the entire set of trajectories again. Finally, this thesis considers spatial and temporal proximity, unlike other proposals. The experimental assessment shows that the proposed algorithms are effective in the obtaining aggregated trajectories.

Keywords — Trajectory, patterns, trajectory aggregation, data mining

Resumen

Hoy en día existen diversas fuentes de datos móviles tales como redes inalámbricas, GPS, y otras, que nos permiten recolectar las ubicaciones geográficas de objetos de manera constante generando grandes colecciones de datos. Estos datos pueden ser usados para generar trayectorias de objetos, donde una trayectoria es una secuencia de segmentos de líneas representando el movimiento continuo de un objeto entre dos instantes de tiempo.

Existen aplicaciones de minería de datos que permiten obtener patrones de trayectorias de objetos, es decir, encontrar comportamientos similares que siguen las trayectorias. Estos patrones son útiles, por ejemplo, para decidir sobre planificación urbana, analizar congestión vehicular, comprender migración de animales, estudiar comportamiento de fenómenos naturales como huracanes, etc. En esta tesis se presenta un nuevo concepto de *trayectoria agregada* y se presentan algoritmos para obtenerlas de manera dinámica. Una *trayectoria agregada* corresponde a una nueva trayectoria que se obtiene a partir de un conjunto de trayectorias que siguen un patrón similar de comportamiento. Una trayectoria agregada no solo muestra el patrón de comportamiento, sino que además, nos indica la densidad de ese patrón. El procesamiento para obtener una trayectoria agregada se realiza en dos fases: (i) obtención de clústeres o grupos de segmentos de trayectorias diferentes y (ii) agregación de clústeres similares.

En esta tesis proponemos aplicar un procesamiento de obtención de clústeres que permita capturar relaciones entre sub-segmentos de segmentos de trayectorias que los algoritmos comunes de la literatura. Además, nuestros algoritmos calculan trayectorias de manera dinámica, a diferencia de otros algoritmos de búsqueda de patrones sobre trayectorias que realizan el procesamiento de manera estática. Es decir, el proceso de construcción de trayectorias agregadas se adapta a medida que se procesan nuevas trayectorias y no es necesario procesar todo el conjunto de trayectorias desde cero si nuevas trayectorias son consideradas en la búsqueda de la trayectoria agregada. Finalmente, a diferencia de otras propuestas no solo se considera la cercanía espacial de trayectorias, sino que además la cercanía temporal. La evaluación experimental realizada comparando resultados en el estado de arte muestra que los algoritmos propuestos son eficaces en el cómputo de trayectorias agregadas.

Palabras Clave — Trayectorias, agregación, patrones, data mining.

Índice general

1. Introducción	1
1.1. Contribuciones	3
1.2. Organización de la Tesis	3
2. Hipótesis y Objetivos	4
2.1. Hipótesis	4
2.2. Objetivos	4
2.2.1. General	4
2.2.2. Específicos	4
2.3. Alcance de la investigación	4
2.4. Metodología	5
3. Estado del Arte	6
3.1. Medidas de Similitud	6
3.2. Agrupamiento y Agregación de Trayectorias	6
4. Trayectorias Agregadas	10
4.1. Preliminares de Trayectorias	10
4.2. Relaciones entre Segmentos	12
4.3. Clúster de segmentos de trayectorias	13
4.4. Generación Dinámica de Clústeres de Segmentos	16
4.5. Concepto de Trayectoria Agregada	17
5. Algoritmos para obtener Trayectorias Agregadas	19
6. Resultados	25
6.1. Descripción de Datos	25
6.2. Trayectorias Agregadas	26
6.2.1. Comparación con Estado del Arte	28
7. Conclusión	31
Referencias	32

Índice de figuras

1.1. Trayectorias y patrones de trayectorias (Hung et al., 2015).	1
1.2. Ejemplo de trayectoria agregada	2
3.1. Componentes de la función de distancia presentada en (Lee et al., 2007).	7
4.1. Un ejemplo de trayectoria	11
4.2. Formas de Relaciones Parciales entre segmentos S_1 y S_2	13
4.3. Generación de clústeres para trayectorias de Ejemplo 4.2	14
4.4. Generación dinámica de clústeres de Ejemplo 4.3	16
4.5. Cómputo de trayectoria agregada del Ejemplo 4.4	17
6.1. Número de Clústeres para distintos valores de Δ_S	26
6.2. Clústeres obtenidos para $\Delta_S = 22$ y $\Delta_T = 0$	26
6.3. Trayectorias agregadas para $\Delta_J = 0\%$	27
6.4. Trayectorias agregadas para $\Delta_J = 15\%$	28
6.5. Trayectorias agregadas para $\Delta_J = 60\%$	29
6.6. Trayectorias agregadas para $\Delta_J = 100\%$	29
6.7. Patrones de trayectorias de huracanes (Lee et al., 2007; Li et al., 2010)	30
6.8. Trayectorias agregadas filtradas	30

Índice de tablas

5.1. División de clústeres	22
6.1. Resultados para distintos umbrales de compatibilidad (Δ_J)	27

Capítulo 1

Introducción

Hoy en día existen diversas fuentes de datos móviles tales como redes inalámbricas, dispositivos de comunicaciones, etc., de las cuales se puede recolectar gran cantidad de ubicaciones geográficas de objetos (Giannotti et al., 2007). El procesamiento de estas ubicaciones permite generar trayectorias espacio-temporales de objetos. Una trayectoria en el espacio 2D se define como una secuencia de n_i segmentos de línea, donde cada segmento representa el movimiento continuo de un objeto o entre localizaciones consecutivas definidas por 3D tuplas de la forma (x_i, y_i, t_i) , con (x_i, y_i) el punto en el espacio en el cual el objeto o estuvo en el instante de tiempo t_i (Orlando et al., 2007). La Figura 1.1 muestra un conjunto de trayectorias T_1, T_2, T_3, T_4, T_5 y T_6 de objetos.

Surge entonces la necesidad de contar con herramientas analíticas que permitan tomar estos datos sobre posiciones de objetos, procesarlos y convertirlos en conocimiento. Existen trabajos en el área de minería de datos donde el objetivo es la obtención de *patrones de trayectorias* (Andrienko y Andrienko, 2011; Furtado et al., 2012; Giannotti et al., 2006, 2007; Hung et al., 2015; Meratnia y de By, 2002; Orlando et al., 2007). De acuerdo a (Giannotti et al., 2007) un patrón de trayectoria representa a un conjunto de trayectorias individuales que visitan la misma secuencia de lugares con similares tiempos de viajes. Por ejemplo, las trayectorias de la Figura 1.1 pueden ser agrupadas en los grupos 1 y 2 (“Cluster 1” y “Cluster 2” en la figura), debido al comportamiento similar espacio-temporal, generándose luego dos patrones de trayectorias (“Trajectory Pattern 1” y “Trajectory Pattern 2” en la figura).

Obtener patrones de trayectorias es útil en diversas aplicaciones, tales como, planificación urbana, congestión vehicular, migración de animales, análisis comportamiento de fenómenos na-

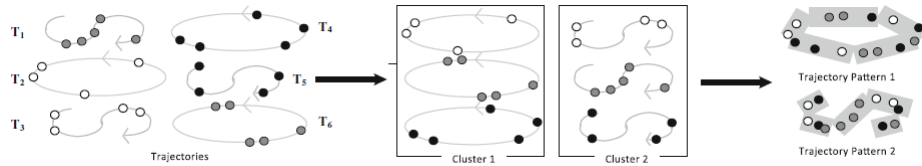


Figura 1.1: Trayectorias y patrones de trayectorias (Hung et al., 2015).

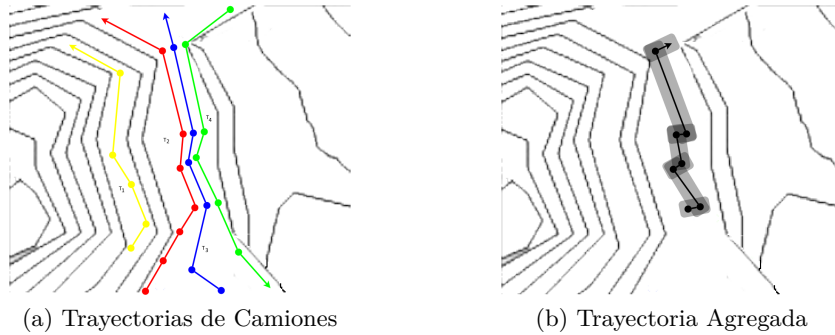


Figura 1.2: Ejemplo de trayectoria agregada

turales como huracanes, etc.

En esta tesis proponemos el concepto de *trayectoria agregada* que, de forma intuitiva, corresponde a una nueva trayectoria construida en base a otras trayectorias que siguen un patrón de comportamiento común considerando un umbral espacial y temporal. El siguiente ejemplo motiva la generación de trayectorias agregadas.

Ejemplo 1.1 Considere una empresa de extracción de minerales cuya planificación de rutas (trayectorias) de extracción se realiza de manera dinámica. En esta empresa la información de camiones y otros vehículos de extracción se obtiene de manera continua lo que permite generar nuevas rutas de extracción. Por ejemplo, la Figura 1.2(a) muestra cuatro trayectorias pertenecientes a camiones en faena de extracción. Se puede entonces, generar una *trayectoria agregada* en base a las trayectorias de extracción que siguen el mismo patrón de comportamiento, lo que podría entregar información de algún área que se encuentra saturada por camiones.

La Figura 1.2(b) muestra la trayectoria agregada resultante. □

Para obtener una trayectoria agregada es necesario agrupar trayectorias o sub-trayectorias que tienen un mismo comportamiento, es decir, cumplen con la condición de ser cercanas desde un punto de vista espacial y temporal. Para determinar si dos trayectorias son cercanas es importante definir una medida de cercanía, por ejemplo la distancia euclidiana. Intuitivamente, para el caso del Ejemplo 1.1, las trayectorias que se pueden agrupar corresponden a sub-trayectorias de las trayectorias T_2 y T_3 en la Figura 1.2(a). Sin embargo, no es posible agrupar las trayectorias T_2 y T_3 completamente porque sus secciones iniciales y finales no son cercanas desde un punto de vista temporal ni tampoco espacial. La trayectoria T_1 no puede considerarse porque todos sus puntos están muy distantes a los puntos que definen a T_2 y T_3 . Tampoco puede ser considerada la trayectoria T_4 , ya que, a pesar de estar cercana espacialmente a T_2 y T_3 , no comparte una distancia temporal apropiada (va en otro sentido).

Una **Trayectoria Agregada (AT)** representa entonces, la agregación de trayectorias en base a patrones de comportamiento con una densidad relevante. Para obtener una trayectoria agregada es necesario cumplir con dos fases: (i) Obtención de clústeres de segmentos de trayectorias

diferentes. Es decir, primero se deben agrupar segmentos de trayectorias cercanas. (ii) Agregación de clústeres similares.

1.1. Contribuciones

Las contribuciones de esta tesis son las siguientes:

- (a) Generación de algoritmos que permitan capturar relaciones entre sub-segmentos de trayectorias y no solo relaciones entre segmentos completos de trayectorias (ver Capítulo 4).
- (b) Procesamiento de clústeres considerando cercanía espacial y temporal, a diferencia de las propuestas de la literatura que no consideran la parte temporal de las trayectorias (por ejemplo, el trabajo presentado en (Lee et al., 2007)).
- (c) Generación de algoritmos que obtienen clústeres de segmentos de trayectorias de manera dinámica, a diferencia de otros algoritmos de búsqueda de patrones sobre trayectorias que realizan el procesamiento de manera estática (Andrienko y Andrienko, 2011; Giannotti et al., 2006; Hung et al., 2015; Meratnia y de By, 2002). Es decir, una vez obtenidos los clústeres de segmentos de trayectorias, es posible actualizar la generación de estos clústeres si nuevas trayectorias son adheridas al procesamiento, y por lo tanto, no es necesario procesar todo el conjunto de trayectorias desde cero. Esto nos permite obtener trayectorias agregadas de forma eficiente (ver Capítulo 5). Para lograr esto seguimos la idea de *macro agregación* presentada en (Li et al., 2010) para clústeres de trayectorias considerando distancia espacial (Lee et al., 2007).

1.2. Organización de la Tesis

El resto del documento se organiza de la siguiente manera. En el Capítulo 2 se define de manera formal la hipótesis y objetivos de esta investigación. En el Capítulo 3 se realiza una descripción del estado del arte sobre cómputo de patrones entre trayectorias. En el Capítulo 4 se describen conceptos preliminares y se presenta el concepto de trayectoria agregada. En el Capítulo 5 se presentan los algoritmos implementados para obtener trayectorias agregadas. En el Capítulo 6 se describen experimentos que muestran la eficacia de nuestros algoritmos. Finalmente, en el Capítulo 7 se sintetizan las principales contribuciones de esta tesis y se describe el trabajo futuro.

Capítulo 2

Hipótesis y Objetivos

En este capítulo se presenta la hipótesis y los objetivos de la tesis.

2.1. Hipótesis

La hipótesis de esta tesis es que es posible obtener de manera dinámica trayectorias agregadas en base a una función de similitud que incluya la distancia espacial y temporal.

2.2. Objetivos

2.2.1. General

El objetivo general de esta tesis es formalizar un nuevo concepto de trayectoria agregada, e implementar algoritmos para generarlas de manera dinámica.

2.2.2. Específicos

Los objetivos específicos de la tesis son:

- (a) Formalizar el concepto de trayectoria y trayectoria agregada.
- (b) Implementar estructuras de datos que permitan almacenar trayectorias agregadas.
- (c) Implementar algoritmos que permitan obtener trayectorias agregadas de forma dinámica.
- (d) Implementar algoritmos estáticos definidos en la literatura para comparar nuestra solución.
- (e) Realizar experimentos que demuestren la efectividad de los algoritmos propuestos.

2.3. Alcance de la investigación

Para la generación de trayectorias, pruebas de algoritmos y la experimentación propuesta en esta tesis se utiliza un conjunto de datos de prueba que contiene información de trayectorias de huracanes. Este conjunto de datos ha sido utilizado en artículos que presentan algoritmos para obtener patrones de trayectorias (Lee et al., 2007; Li et al., 2010).

2.4. Metodología

Las siguientes son las actividades que se llevaron a cabo para completar los objetivos específicos planteados en la tesis.

- Análisis de literatura relevante sobre minería de trayectorias, agregación de trayectorias y búsqueda de patrones en trayectorias.
- Definición formal del concepto de trayectoria agregada.
- Análisis y procesamiento de conjuntos de datos de prueba disponibles en la red.
- Implementación de estructuras de datos para el almacenamiento de trayectorias agregadas.
- Diseño e implementación de algoritmos para construir trayectorias agregadas en base a conjuntos de trayectorias de entrada.
- Implementación de algunos algoritmos definidos en la literatura para comparar nuestra solución.
- Ejecución de experimentos para probar la eficacia de nuestros algoritmos.

Capítulo 3

Estado del Arte

Los algoritmos de agrupamiento de trayectorias están dirigidos a agrupar trayectorias similares (o parte de trayectorias), de acuerdo a una medida de similitud. Agrupar subconjuntos de trayectorias puede ser útil en diversos escenarios, tales como, descubrir las regiones congestionadas en el tráfico, analizar los movimientos de los huracanes, los movimientos de los animales, entre otros. El principal problema a la hora agregar trayectorias es la imprecisión de los datos espacio-temporales (Meratnia y de By, 2002). Por lo tanto, es importante definir una buena medida de similitud para comparar dos trayectorias diferentes.

3.1. Medidas de Similitud

Existen varias nociones acerca de similitud de trayectorias. Por ejemplo, dos trayectorias pueden considerarse similares si coinciden (aproximadamente) en sus puntos inicial y final. Por otra parte, dos trayectorias pueden considerarse similares si coinciden en algunas partes de las trayectorias (Lee et al., 2007; Sankararaman et al., 2013), y así sucesivamente.

El trabajo en (Wang et al., 2013) analiza diferentes enfoques para medidas de similitud clasificadas en: (i) distancia euclidiana, (ii) deformación dinámica de tiempo, (iii) modificación de distancia, y (iv) sub-secuencias comunes más largas. En este trabajo se analiza cómo estas medidas se ven afectadas por la presencia de ruido, diferentes frecuencias de muestreo, y los cambios de trayectorias, concluyendo que todos ellos tienen ventajas y desventajas. Para efectos de este trabajo, se utiliza la medida distancia euclidiana. Además, utilizamos este enfoque, ya que es sensible a la comparación visual, es computacionalmente asequible, y se complementa con estrategias para evitar la vulnerabilidad a la velocidad de muestreo y el ruido. De todas maneras, el tipo de distancia a utilizar en nuestros algoritmos puede ser reemplazado por otro sin mayores problemas.

3.2. Agrupamiento y Agregación de Trayectorias

En (Lee et al., 2007) los autores proponen el marco de *partición y agrupamiento* que permite el descubrimiento de sub-trayectorias comunes desde una base de datos de trayectorias. Se trabaja

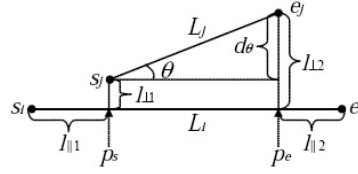


Figura 3.1: Componentes de la función de distancia presentada en (Lee et al., 2007).

en dos fases. En primer lugar, se particionan trayectorias en un conjunto de segmentos de línea y luego se agrupan segmentos de línea completos y similares. Las trayectorias se particionan de acuerdo a puntos característicos. Entonces, generan una trayectoria representativa para cada grupo. La principal diferencia con nuestra propuesta es que ellos agrupan los segmentos de línea completos y nosotros somos capaces de agrupar porciones de segmentos de línea. La ventaja de nuestra solución es que nos permite capturar relaciones entre sub-segmentos que de otra forma se perderían. De esta forma nuestros clústeres de segmentos son más precisos.

El algoritmo para obtener clústeres se basa en una adecuada *función de distancia* que se compone de la *distancia perpendicular* ($d_{\perp}$), la *distancia paralela* ($d_{\parallel}$), y la *distancia angular* (d_{θ}). Considere dos segmentos de línea d-dimensional $L_i = s_i e_i$ y $L_j = s_j e_j$, donde s_i, s_j, e_i, e_j son, respectivamente, los puntos iniciales y finales de los segmentos de línea L_i y L_j . Los tres componentes de la función de distancia están ilustrados en la en la Figura 3.1, donde p_s y p_e son los puntos de proyección de s_j y e_j en el segmento de línea L_i , $l_{\perp 1}$ es la distancia Euclidiana entre s_j y p_s , y $l_{\perp 2}$ es la distancia euclidiana entre e_j y p_e .

La distancia perpendicular entre L_i y L_j se define por la fórmula:

$$d_{\perp}(L_i, L_j) = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}} \quad (3.1)$$

La distancia paralela entre L_i y L_j se define por la fórmula 3.2, donde $l_{\parallel 1}$ es el mínimo de las distancias euclidianas de p_s hasta s_i (el punto de partida de L_i) y e_i (el punto final de L_i). De manera similar, $l_{\parallel 2}$ es el mínimo de las distancias euclidianas de p_e hasta s_i y e_i .

$$d_{\parallel}(L_i, L_j) = \text{MIN}(l_{\parallel 1}, l_{\parallel 2}) \quad (3.2)$$

La distancia angular entre L_i y L_j se define por la fórmula 3.3, en donde, $\|L_j\|$ es la longitud de L_j y θ ($0^\circ \leq \theta \leq 180^\circ$) es el ángulo de intersección menor entre L_i y L_j .

$$d_{\theta}(L_i, L_j) = \begin{cases} \|L_j\| \times \sin(\theta), & \text{if } 0^\circ \leq \theta < 90^\circ \\ \|L_j\|, & \text{if } 90^\circ \leq \theta \leq 180^\circ \end{cases} \quad (3.3)$$

Entonces, la distancia entre dos segmentos de línea se define por la fórmula 3.4 (Lee et al., 2007), donde $w_{\perp}$, $w_{\parallel}$ y w_{θ} son pesos. Los autores en (Lee et al., 2007) utilizan 1 como valor prede-

terminado para todos los pesos. Esta definición de la distancia es simétrica, esto es, $dist(L_i, L_j)$ es igual a $dist(L_j, L_i)$.

$$dist(L_i, L_j) = w_{\perp} \cdot d_{\perp}(L_i, L_j) + w_{\parallel} \cdot d_{\parallel}(L_i, L_j) + w_{\theta} \cdot d_{\theta}(L_i, L_j) \quad (3.4)$$

El algoritmo propuesto en (Lee et al., 2007) funciona de la siguiente manera. En primer lugar, todos los segmentos de línea se consideran “no clasificados”, y se pueden clasificar como parte de un clúster o como ruido. A continuación, el algoritmo calcula el vecindario- ϵ de cada segmento de línea “no clasificado” L . El vecindario- ϵ para un segmento de línea L_i es un conjunto de segmentos de línea de tal manera que, para cada segmento de línea L_j dentro de este conjunto, se cumple con $dist(L_i, L_j) \leq \epsilon$. Si para un segmento de línea L se determina que es un segmento de línea base, esto es, si la cardinalidad de su vecindario- ϵ es igual o mayor que un valor específico (llamado MinLns), el algoritmo expande un clúster. Sólo los grupos con una cardinalidad superior a un umbral determinado se consideran válidos. Por último, el algoritmo calcula una trayectoria representativa para cada clúster.

La principal diferencia de este trabajo previo con respecto a nuestra propuesta, es que nosotros somos capaces de agrupar porciones de segmentos de línea, en vez de agrupar segmentos de línea completos. Además, nosotros consideramos el tiempo en la comparación de los segmentos y cuando se realiza el agrupamiento, no se excluyen segmentos de línea considerándolos ruido, sino que se conservan para luego utilizarlos en futuros cálculos de los clústeres. Solo al final del proceso de agrupamiento, son descartados los segmentos que no se hayan agrupado. De esta manera, nuestra aplicación logra capturar relaciones entre sub-segmentos de trayectorias y permite generar clústeres de segmentos mas precisos.

El trabajo presentado en (Li et al., 2010) amplía los algoritmos presentados en (Lee et al., 2007) al permitir un cálculo incremental de los clústeres. El nuevo enfoque se basa en dos pasos llamados “mantenimiento online de micro-clústeres” y “creación offline de macro-clústeres”. En la primera fase, cuando llegan nuevas trayectorias, se simplifican en conjuntos de segmentos y, de ser posible, se agregan en clústeres apropiados existentes. De lo contrario, se crea un nuevo grupo para cada segmento de la nueva trayectoria. Luego, a petición del usuario, se pueden computar macro-clústeres sobre los micro-clústeres. A este tipo de procesamiento, nosotros lo catalogamos como procesamiento dinámico de clústeres de segmentos de trayectorias, el cual también es explorado en esta tesis. Sin embargo, a diferencia del trabajo presentado en (Li et al., 2010), nosotros consideramos computación de clústeres considerando cercanía espacial y temporal de segmentos de trayectorias, y no solo cercanía espacial como la propuesta en (Li et al., 2010).

El trabajo presentado en (Hung et al., 2015) propone un marco de minería de patrones de trayectorias llamado “Similitud de Trayectorias basados en Pistas” o CATS por su sigla en inglés (Clue-Aware Trajectory Similarity), para determinar un clúster o grupo de trayectorias. Este trabajo se basa en el concepto de *hot regions* (regiones calientes, en español). Una hot region es una región espacial donde varios objeto aparecen en un intervalo de tiempo específico. Más formalmente, corresponde a una estructura espacio-temporal r_i que agrupa trayectorias similares y satisface las siguientes restricciones: (i) el intervalo de tiempo de r_i , denotado por $[r_i.S, r_i.E] \geq \tau$, siendo τ un umbral determinado. (ii) La proyección de r_i en el plano XY es un rectángulo de tal manera que cualquier punto x dentro de este rectángulo tiene una distancia

con respecto a la línea representativa $\overline{L_i}$ menor que un umbral determinado ϵ , $\overline{L_i}$ es la regresión lineal de los puntos. Luego de identificar regiones hot se obtienen *patrones de trayectoria*, que son secuencias ordenadas de regiones hot de la forma $TP = r_1 r_2 \dots r_k$.

CATS utiliza la *Función de Decaimiento Espacial* para medir la similitud de dos trayectorias. Dados dos puntos de datos $p_1 = (x_1, y_1, t_1)$ y $p_2 = (x_1, y_1, t_1)$ a partir de dos trayectorias diferentes, la *Función de Decaimiento Espacial* se define por:

$$f_\epsilon(p_1, p_2) = \begin{cases} 0 & \text{if } dist(p_1, p_2) > \epsilon \\ 1 - \frac{dist(p_1, p_2)}{\epsilon} & \text{otherwise} \end{cases}, \quad (3.5)$$

donde $dist(\cdot)$ es la distancia Euclidiana, y ϵ es un parámetro de tolerancia. El valor de esta función varía entre 0 a 1. Si los puntos son cercanos entre sí, el valor de las funciones es más grande. Si los puntos son los mismos, el valor es 1.

El algoritmo de agrupamiento presentado en (Hung et al., 2015) trabaja en tres pasos. En el primero, se utiliza un grafo para representar los valores CATS entre pares de trayectorias. En la segunda etapa, las trayectorias con valores de CATS a cada una de las demás suficientemente grandes, se agrupan como núcleos de conjuntos. Finalmente, los núcleos de conjuntos se combinan en grupos. Sólo los grupos con un soporte de trayectorias suficiente se identifican como comportamientos movimientos frecuentes. Para cada grupo, las regiones hot se derivan para obtener patrones de trayectoria. La diferencia con nuestro trabajo es que ellos obtienen clúster de puntos, pero no clúster de segmentos. Entonces, un patrón de trayectoria es una secuencia de regiones hot, y nuestros patrones de trayectoria son agregaciones de clústeres de segmentos.

Por otro lado, el trabajo en (Andrienko y Andrienko, 2011) ataca el problema de mejorar los métodos de extracción de datos para visualizar datos de objetos en movimiento. Una agregación de trayectorias corresponde a la aplicación de un operador que reúne una serie de trayectorias en una sola trayectoria. El método funciona de la siguiente manera. Se extraen puntos específicos de trayectorias, se agrupan estos puntos de acuerdo a su proximidad espacial, y se utilizan los centroides de los grupos para generar una teselación de Voronoi. A continuación, la partición genera celdas para agregar trayectorias. Puntos interesantes de una trayectoria son: inicio, fin, giros significativos, paradas significativas. La agrupación de puntos de interés se realiza mediante el uso de un algoritmo ad-hoc que produce agrupaciones espaciales convexas con extensión espacial deseada y luego un algoritmo que optimiza los grupos mediante la re-agrupación de puntos alrededor del centro de regiones densas. Las trayectorias son representadas como una secuencia de visitas a celdas, es decir, secuencia de celdas con su respectivo intervalo de tiempo que refleja el momento de visita del objeto a las celdas y una secuencia de movimientos, es decir, entradas hacia y salidas desde las celdas con el intervalo de tiempo del movimiento.

Similar al trabajo en (Hung et al., 2015), en (Giannotti et al., 2007) los autores presentan algoritmos de minería de datos para obtener patrones espacio-temporales, definidos como una secuencia de regiones hot o zonas de interés. El patrón de una trayectoria representa trayectorias que visitan la misma secuencia de lugares con los tiempos de viaje de transición similares. En este trabajo se analiza el tiempo de viaje de los objetos pero no se consideran los instantes de tiempo en que suceden los eventos.

Capítulo 4

Trayectorias Agregadas

En este capítulo se presenta el nuevo concepto de **Trayectoria Agregada**. Para comprender este concepto es necesario introducir algunos conceptos relativos a trayectorias de objetos, segmentos de trayectorias, relaciones entre segmentos y clústeres de segmentos de trayectoria, las cuales son presentadas en la Sección 4.1.

4.1. Preliminares de Trayectorias

En esta sección se presentan conceptos básicos aplicados a las trayectorias de objetos que serán utilizados para la definición, obtención y manipulación de trayectorias agregadas (ver Sección 4.5).

Definición 4.1 (Intervalo de Tiempo) Un *intervalo de tiempo* es definido por un Intervalo $I = [t_s, t_e]$, donde $t_s, t_e \in \mathbb{R}$ y $t_e \geq t_s$. $\square$

Definición 4.2 (SPoint vs TPoint) Un SPoint es una tupla (x, y) en el espacio Euclidiano. Un TPoint es una tupla $(SPoint, t)$ donde t es un instante de tiempo. $\square$

Por simplicidad, usamos la siguiente notación:

- Dado un TPoint $P = ((x_i, y_i), t_i)$, $P.space = (x_i, y_i)$.
- Dado un SPoint $P' = (x_i, y_i)$, $P'.x = x_i$ y $P'.y = y_i$.

Abusando de la notación, podemos decir que:

- Dado un TPoint $P = ((x_i, y_i), t_i)$, $P.x = x_i$, $P.y = y_i$ y $P.t = t_i$.
- Dado un Intervalo $IT = [t_s, t_e]$, $IT.start = t_s$ y $IT.end = t_e$.

Definición 4.3 (Segmento) Un Segmento S de una trayectoria de objetos es un par de TPoints $\langle P_s, P_e \rangle$, donde $P_s = ((x_s, y_s), t_s)$, $P_e = ((x_e, y_e), t_e)$, y $t_e > t_s$. $\square$

La siguiente es la definición formal de trayectoria que se considerará en este trabajo, la cual está basada en la definición descrita en (Orlando et al., 2007).

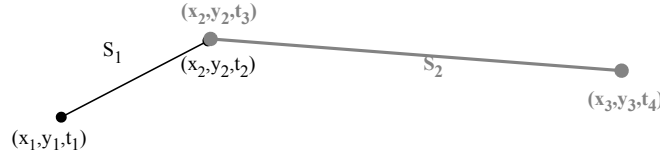


Figura 4.1: Un ejemplo de trayectoria

Definición 4.4 (Trayectoria) Una Trayectoria T de un objeto es una secuencia de segmentos consecutivos $\langle S_1, S_2, \dots, S_m \rangle$ que describe el paso de un objeto y donde para cada par de segmentos consecutivos $S_i = [P_s, P_e]$ y $S_j = [P'_s, P'_e]$ en T , $P_e.x = P'_s.x$, $P_e.y = P'_s.y$ y $P'_s.t = P_e.t$.

Una **sub-trayectoria** T_s es el subconjunto de segmentos consecutivos de una trayectoria de objeto T . Es importante recalcar que una sub-trayectoria es, a su vez, una trayectoria. Abusando de la notación, podemos decir que $T_s \subseteq T$ si T_s es una sub-trayectoria de T . $\square$

Ejemplo 4.1 Considere la trayectoria representada en la Figura 4.1, la cual está compuesta de dos segmentos consecutivos S_1 y S_2 . S_1 es temporalmente anterior a S_2 , porque el punto final del segmento S_1 coincide con el punto inicial del segmento S_2 y los instantes de tiempo de estos dos puntos son iguales, es decir, $t_3 = t_2$. $\square$

Se utilizarán los siguiente operadores sobre trayectorias.

Definición 4.5 (Operadores) Dada una Trayectoria $T = \langle S_1, S_2, \dots, S_n \rangle$ con segmentos $S_1 = [P_s, P_e]$ y $S_n = [P'_s, P'_e]$, y dada una secuencia de SPoints (PS). Los siguientes operadores son válidos:

- $SD(P, P')$ denota la distancia espacial Euclidiana entre los SPoints P y P' . Si los puntos estuvieran conectados, ésta debería ser la distancia en la grilla de representación. Sobrecargando este operador, $SD(S, P)$ es la distancia Euclidiana más corta entre el SPoint P y el segmento S (por ejemplo, el largo de la perpendicular a S que pasa por P).
- $TD(P, P') = |P.t - P'.t|$ es la distancia temporal entre los TPoints P y P' .
- $START_s(T)$ retorna el segmento inicial de la trayectoria T .
- $END_s(T)$ retorna el segmento final de la trayectoria T .
- $START_p(S)$ retorna el punto inicial del segmento S .
- $START_p(T)$ retorna el punto inicial de la trayectoria T , es decir: $START_p(T) = START_p(START_s(T))$.
- $END_p(S)$ retorna el punto final del segmento S .
- $END_p(T)$ retorna el punto final de la trayectoria T , es decir, $END_p(T) = END_p(END_s(T))$.
- $LENGTH(T)$ retorna el número de segmento de T .
- Dado un segmento S y un SPoint P , $NEAR(S, P)$ retorna verdadero si $SD(S, P) \leq \Delta_n$, donde Δ_n es un margen predefinido.
- $ANGLE(S_1, S_2)$ retorna el ángulo formado entre los segmentos S_1 y S_2 . $\square$

4.2. Relaciones entre Segmentos

En esta tesis se propone agrupar trayectorias que son similares, y para ello es necesario definir una medida de similitud entre trayectorias. Nuestra propuesta, analiza segmentos de trayectorias y no trayectorias completas. Para el análisis y comparación entre segmentos de trayectoria, se han definido dos tipos de relaciones entre segmentos: **relación total** y **relación parcial**.

Intuitivamente, dos segmentos S_1 y S_2 se **relacionan totalmente** cuando tanto sus puntos iniciales como sus puntos finales son cercanos, espacial y temporalmente, y la dirección de ambos segmentos es semejante, es decir, el ángulo formado entre ellos es menor a 90° . La siguiente definición formaliza este tipo de relación.

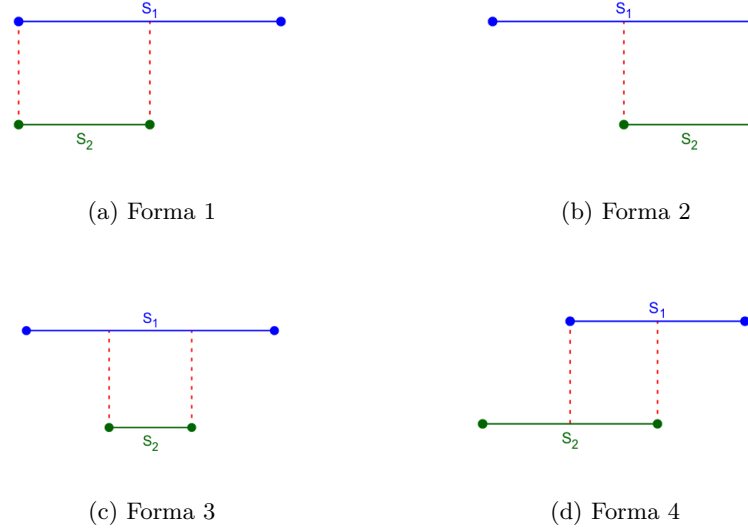
Definición 4.6 (Segmentos relacionados totalmente) Dos segmentos S_1 y S_2 se **relacionan totalmente** si y sólo si:

- $SD(START_p(S_1), START_p(S_2)) \leq \Delta_S$
- $TD(START_p(S_1), START_p(S_2)) \leq \Delta_T$
- $SD(END_p(S_1), END_p(S_2)) \leq \Delta_S$
- $TD(END_p(S_1), END_p(S_2)) \leq \Delta_T$
- $ANGLE(S_1, S_2) < 90^\circ$. □

Por otra parte, dos segmentos se **relacionan parcialmente** cuando, sin estar relacionados totalmente y teniendo la misma dirección, un extremo de un segmento es cercano, espacial y temporalmente, a algún punto (no necesariamente extremo) del otro. En este caso, pueden surgir diferentes formas de relación parcial entre segmentos. En esta tesis hemos considerado cuatro formas de relación parcial entre segmentos las que se describen en la siguiente definición.

Definición 4.7 (Segmentos relacionados parcialmente) Dos segmentos S_1 y S_2 se **relacionan parcialmente** si y sólo si S_1 y S_2 no se relacionan de manera total y cumplen con alguna de las siguientes cuatro formas, las cuales están representadas en la Figura 4.2:

- Forma 1 (Figura 4.2a). En este caso:
 - Los puntos iniciales de ambos segmentos S_1 y S_2 son espacio-temporalmente cercanos entre sí (satisfacen los umbrales Δ_S y Δ_T).
 - El punto final del segmento S_1 no es cercano a ningún punto del segmento S_2 .
 - El punto final del segmento S_2 es cercano a algún punto (no extremo) del segmento S_1 .
- Forma 2 (Figura 4.2b). En este caso:
 - Los puntos finales de ambos segmentos S_1 y S_2 son espacio temporalmente cercanos entre sí.
 - El punto inicial del segmento S_1 no es cercano a ningún punto del segmento S_2 .
 - El punto inicial del segmento S_2 es cercano a algún punto (no extremo) del segmento S_1 .
- Forma 3 (Figura 4.2c). En este caso:
 - Tanto el punto inicial como el punto final del segmento S_1 no son cercanos a ningún punto del segmento S_2 .

Figura 4.2: Formas de Relaciones Parciales entre segmentos S_1 y S_2

- Tanto el punto inicial como el punto final del segmento S_2 son cercanos a algún punto (no extremo) del segmento S_1 .
- Forma 4 (Figura 4.2d). En este caso:
 - El punto inicial del segmento S_1 es cercano a algún punto (no extremo) del segmento S_2 .
 - El punto final del segmento S_1 no es cercano a ningún punto del segmento S_2 .
 - El punto inicial del segmento S_2 no es cercano a ningún punto del segmento S_1 .
 - El punto final del segmento S_2 es cercano a algún punto (no extremo) del segmento S_1 .

□

4.3. Clúster de segmentos de trayectorias

En el área de las bases de datos espacio-temporales, regularmente se utiliza el término **clustering** como el proceso de agrupar conjuntos de objetos espaciales (y/o temporales) que tienen características comunes y/o comportamientos similares, los cuales se conocen como **clústeres** (clusters).

En esta investigación, se utiliza el concepto de **clúster de segmentos de trayectorias** para referirse a un conjunto de segmentos $SS = S_1, S_2, \dots, S_n$ de diferentes trayectorias (no necesariamente consecutivos) que se relacionan de manera total. En (Li et al., 2010) utilizan el concepto de **micro-clústeres** para referirse a un clúster de segmentos de trayectorias.

Como se ilustró anteriormente, dos segmentos de trayectorias pueden relacionarse de manera

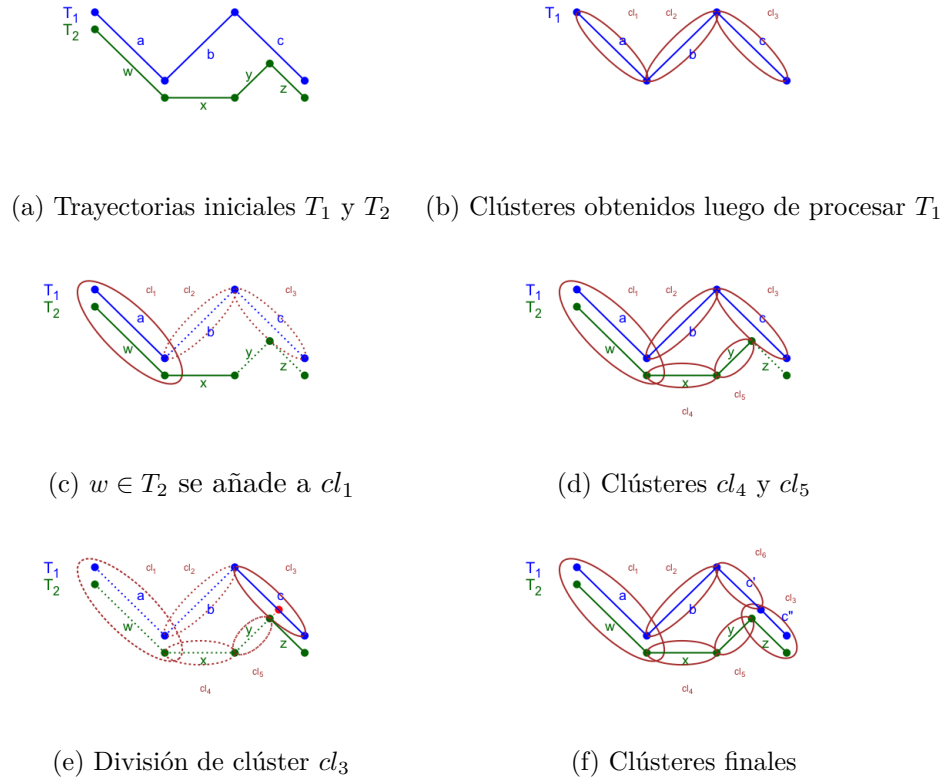


Figura 4.3: Generación de clústeres para trayectorias de Ejemplo 4.2

parcial, y no total. Si esto sucede, los segmentos no formarían parte de un clúster si un algoritmo estricto se aplica. En esta propuesta, creemos que si dos segmentos se relacionan de manera parcial, entonces, los sub-segmentos que se relacionan de manera total, deben ser considerados en algún clúster. De esta forma, podemos obtener clústeres más precisos y no perder información relevante de las trayectorias.

Para lograr lo anterior, se procede de la siguiente manera considerando dos segmentos S_1 y S_2 : (i) Se determina cuál forma de relación parcial existe entre S_1 y S_2 (de acuerdo a las formas definidas en la Definición 4.7. (ii) Se genera un nuevo segmento S_s que corresponde a la proyección del segmento de longitud menor sobre el segmento de longitud mayor. (iii) Se genera un nuevo clúster con el segmento de longitud menor y el nuevo segmento S_s .

El siguiente ejemplo muestra el proceso de obtención de clústeres de segmentos de trayectorias.

Ejemplo 4.2 Considere el conjunto de trayectorias de la Figura 4.3(a) compuesto de trayectorias $\{T_1, T_2\}$, y asuma que la primera trayectoria a procesar es T_1 , se procede como sigue:

- (a) Cada uno de los segmentos de T_1 da origen a un clúster, generándose, los clústeres cl_1 , cl_2 y cl_3 en Figura 4.3(b).

- (b) Luego se considera la trayectoria T_2 . Para cada uno de los segmentos de T_2 se analiza la relación de estos con segmentos de los clústeres ya computados, es decir, se determina si son *admisibles de ser añadidos* a un clúster previamente computado.
Consideremos el segmento $w \in T_2$. Se compara w con el clúster cl_1 , es decir, se determina el tipo de relación (total o parcial) entre w y el único segmento a de cl_1 (4.3(b)). Como a y w se relacionan totalmente, se procede a añadir el segmento w al clúster cl_1 tal como lo muestra la Figura 4.3(c).
- (c) Luego, se considera como segmento entrante al segmento $x \in T_2$. En primer lugar, se compara con cl_1 4.3(c), pero como cl_1 ya contiene un segmento perteneciente a la trayectoria T_2 , se continua con el siguiente clúster.
Dado que x no se relaciona de manera total ni parcial con los segmentos en los clústeres cl_2 y cl_3 , por lo tanto, se genera un nuevo clúster cl_4 conformado por el segmento x , el cual se ilustra en la Figura 4.3(d).
- (d) Luego, se considera el segmento $y \in T_2$ y se compara con el clúster cl_1 . Dado que cl_1 ya contiene un segmento de la trayectoria T_2 no es posible añadirlo a él (lo mismo sucede cuando se compara con el clúster cl_4). Este segmento no se añade a ninguno de los restantes clústeres dado que no se relaciona ni total ni parcialmente con los segmentos en ellos.
Como y no puede ser añadido a ningún clúster, se crea un nuevo clúster cl_5 conformado por el segmento y , el cual se ilustra en la Figura 4.3(d).
- (e) Finalmente se procesa el segmento $z \in T_2$, el cual no puede ser añadido al clúster cl_1 puesto que éste ya contiene un segmento de la trayectoria T_2 . Este segmento tampoco puede ser añadido al clúster cl_2 ya que z no se relaciona ni total ni parcial con el segmento b del clúster cl_2 . Sin embargo, z se relaciona parcialmente con el segmento c del clúster cl_3 . La forma de relación corresponde a la forma 2 de la Definición 4.7. Por lo tanto, se divide el clúster cl_3 en dos. Se obtiene el clúster cl_6 que contiene el nuevo segmento c' (sub-segmento de c obtenido desde $START_p(c)$ hasta la proyección de $START_p(z)$ en c) y se actualiza el clúster cl_3 con el sub-segmento c'' que se obtiene computando la proyección de $START_p(z)$ hasta $END_p(c)$, esto se ilustra en la Figura 4.3(e). El segmento z se añade al nuevo clúster cl_3 como se muestra en la Figura 4.3(f). $\square$

Para la definición del concepto de trayectoria agregada, necesitamos además el concepto de **clúster agregado de segmentos de trayectorias**, al cual llamaremos simplemente *clúster agregado*, el cual corresponde a un conjunto de *clústeres de segmentos de trayectorias* que comparten las siguientes características:

- Cada clúster de segmentos dentro de un clúster agregado debe contener por lo menos dos segmentos.
- El área asociada a cada clúster de segmentos incluido en un clúster agregado intersecta al menos un punto el área (o áreas) de otro clúster (otros clústeres) de segmento(s) en el mismo clúster agregado.
- Los clústeres de segmentos dentro del clúster agregado cumplen con un nivel de compatibilidad mayor a un umbral Δ_J definido. La compatibilidad corresponde a un porcentaje de trayectorias en común entre los clústeres de segmentos dentro de un clúster agregado.

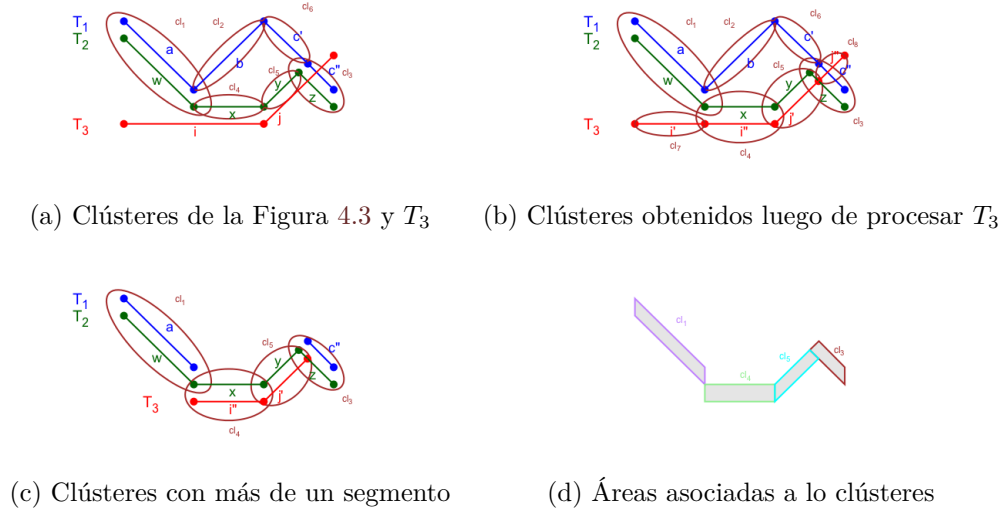


Figura 4.4: Generación dinámica de clústeres de Ejemplo 4.3

Cabe destacar que un clúster agregado, puede ser tratado como un clúster de segmento dentro de un proceso iterativo para el cómputo de trayectoria agregada. Este concepto de clúster agregado corresponde al concepto de **macro-clústeres** en (Li et al., 2010).

4.4. Generación Dinámica de Clústeres de Segmentos

El dinamismo en la generación de clústeres de segmentos de trayectorias se entiende como la capacidad de permitir que, dado un conjunto de clústeres de segmentos CS ya computados, se es capaz de actualizar los clústeres de segmentos ante la llegada de nuevas trayectorias, sin partir con todo el procesamiento desde cero. Es importante mencionar que solo se considera la inserción de nuevas trayectorias pero no la eliminación de trayectorias. Nuestros algoritmos permiten el procesamiento dinámico de trayectorias ya que si una nueva trayectoria T ingresa luego de haber computado un conjunto de clústeres CS , los segmentos de esta nueva trayectoria pueden ser incorporados a algún clúster en CS o bien se pueden generar nuevos clústeres que los contenga.

Ejemplo 4.3 Considere los clústeres de segmentos de la Figura 4.3(f) y la inserción de una nueva trayectoria T_3 a los clústeres existentes (Figura 4.4(a)). El segmento i de T_3 es particionado en dos sub-segmentos, i' queda en el cluster cl_7 , i'' se incorpora al cluster cl_4 . Lo mismo sucede con el segmento j que es dividido en dos sub-segmentos que se ubican en los clústeres cl_5 y cl_8 , respectivamente. La Figura 4.4(b) muestra los ocho clústeres de segmentos resultantes al ingresar la nueva trayectoria.

Para computar los nuevos clústeres agregados, se descartan los clústeres de segmentos cl_2 , cl_6 , cl_7 y cl_8 ya que contienen un único segmento. La Figura 4.4(c) muestra los clústeres que pueden ser agregados, cl_1 , cl_3 , cl_4 y cl_5 . La Figura 4.4(d) muestra el área asociada a cada cluster (cierres convexo de los puntos de cada uno de los segmentos en el clúster). Si el nivel de compatibilidad establecido (Δ_J) entre clústeres se cumple, entonces estos cuatro clústeres podrían ser agregados, dado que sus áreas se intersectan y tienen más de un segmento. $\square$

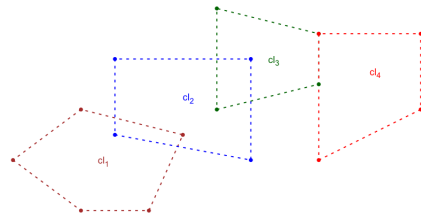
4.5. Concepto de Trayectoria Agregada

Intuitivamente, una **Trayectoria Agregada** (AT) es una estructura que representa la agregación de trayectorias en base a patrones de comportamiento con una densidad relevante.

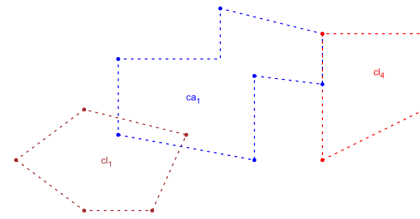
La siguiente definición presenta el concepto formal de Trayectoria Agregada.

Definición 4.8 (Trayectoria Agregada) Una **Trayectoria Agregada** es una tupla de la forma $AT = [CS, N]$, donde:

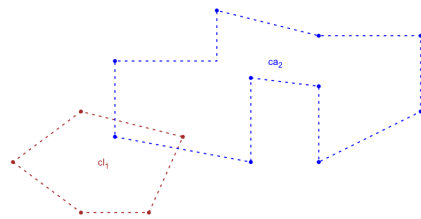
- CS es un conjunto ordenado de clústeres agregados.
- N es el número de trayectorias individuales que participan en la trayectoria agregada. $\square$



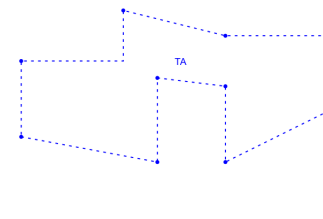
(a) Clústeres de segmentos



(b) Clúster agregado ca_1



(c) Clúster agregado ca_2



(d) Trayectoria agregada

Figura 4.5: Cómputo de trayectoria agregada del Ejemplo 4.4

Ejemplo 4.4 Considere los clústeres de segmentos de la Figura 4.5(a), un umbral de compatibilidad de un 75 % y las siguientes compatibilidades entre clústeres:

	cl_2	cl_3	cl_4
cl_1	60	80	90
cl_2	-	85	90
cl_3	85	-	95

Los clústeres cl_2 y cl_3 intersectan y tienen una compatibilidad igual a 85 %, mayor al umbral exigido (75 %), por lo tanto pueden ser agregados, formando el cluster $ca_1 = \{cl_2, cl_3\}$ que se muestra en la Figura 4.5(b). El cluster ca_1 será agregado a los clústeres de segmentos si satisfacen el umbral de compatibilidad, dado que si se intersectan. Asumamos que la compatibilidad entre ca_1 y cl_4 es mayor a 75 %, pero la compatibilidad entre ca_1 y cl_1 es menor al 75 %. La Figura 4.5(c) muestra el clúster agregado ca_2 que corresponde a la agregación del clúster agregado ca_1 con el clúster cl_4 . La Figura 4.5(d) muestra la trayectoria agregada final y que corresponde a la estructura $AT = [\{cl_2, cl_3, cl_4\}, N]$, donde N corresponde al número de trayectorias individuales involucradas en la trayectoria agregada. $\square$

Capítulo 5

Algoritmos para obtener Trayectorias Agregadas

En términos algorítmicos, un conjunto de trayectorias agregadas AT es una estructura que representa la agregación de un conjunto de clústeres de segmentos CS . El procedimiento para obtener trayectorias agregadas se puede dividir en dos fases. En la primera, se procede a obtener todos los clústeres de segmentos CS asociados a las trayectorias de entrada en TS . En la segunda parte, se procede a computar los clústeres agregados desde el conjunto de clústeres de segmentos CS de tal forma de obtener trayectorias agregadas. El algoritmo 1 describe este procedimiento.

Algoritmo 1: AGREGACIÓN DE TRAYECTORIAS - “**TrajectoryAggregation**”

Data: Un conjunto de Trayectorias TS , un umbral espacial Δ_S , un umbral temporal Δ_T , un umbral de compatibilidad Δ_J

Result: Un conjunto de trayectorias agregadas AT

```

1  $CS \leftarrow \text{TrajectoryClustering}(TS, \Delta_S, \Delta_T);$ 
2  $AT \leftarrow \text{ClusterAggregation}(CS, \Delta_J);$ 
3 return  $AT$ 

```

El agrupamiento de segmentos de trayectorias es realizado a partir del Algoritmo 2 el cual hace uso del Algoritmo 3 para comparar segmentos. El Algoritmo 2 toma cada segmento de trayectoria y trata de asignarlo a un clúster existente en CS , si es que existe, y genera un conjunto de segmentos de residuo R , si es necesario. Los segmentos en el conjunto residuo R son segmentos que aún no se han asignado a un clúster debido a que se generan por partición de segmentos en caso de encontrar admisibilidad parcial y que requieren ser analizados en conjunto con los otros clústeres.

Inicialmente, dado que no existen clústeres, se genera un cluster por cada uno de los segmentos de la primera trayectoria, esto como resultado de la llamada al Algoritmo 3. Luego a medida que se continúa con segmentos de otras trayectorias, por cada segmento se busca en la llamada al Algoritmo 3 un clúster con el cual agrupar y si se producen segmentos residuos, trata de asignarlos a los clústeres que existen (líneas 5 a 10) antes de continuar con otros segmentos de

Algoritmo 2: AGRUPAMIENTO DE SEGMENTOS - “TrajectoryClustering”

Data: Un conjunto TS de trayectorias (conjunto de segmentos), un umbral espacial Δ_S ,
un umbral temporal Δ_T

Result: El conjunto de clústeres de segmentos CS

```

1  $CS \leftarrow \text{New ClusterSet}()$  /*Crea un conjunto de clústeres inicialmente vacío */
2 foreach  $T \in TS$  do
3   foreach  $s \in T$  do
4      $R \leftarrow \text{SegmentClustering}(s, \Delta_S, \Delta_T, \text{var } CS);$ 
5     while  $R \neq \emptyset$  do
6        $s' \leftarrow \text{Next}(R)$  /*Toma el próximo elemento de  $R$  */
7       /*Agrega el segmento  $s'$  a un clúster y, si es necesario, crea un
        clúster residuo  $R'$  */
8        $R' \leftarrow \text{SegmentClustering}(s', \Delta_S, \Delta_T, \text{var } CS);$ 
9       /*Inserta segmentos del clúster residuo  $R'$  en conjunto  $R$  */
10       $R \leftarrow \text{Merge}(R, R');$ 
11       $\text{Delete}(R');$ 
12      /*Elimina  $s'$  de  $R$  */
13       $R \leftarrow \text{Remove}(R, s');$ 

```

trayectoria.

El proceso de asignación de un segmento s a un clúster en CS se realiza con el Algoritmo 3. Este algoritmo revisa tres casos posibles: (1) Encuentra un clúster C en CS con el cual s tiene una admisibilidad total, si es que existe; es decir, existe al menos un segmento en C con el cual existe una cercanía en base a Δ_S y Δ_T y tienen la misma dirección. Si esto se cumple con todos los elementos de C , entonces la admisibilidad es *total y completa* (línea 7 y 8), o por lo contrario, es *total e incompleta*, en cuyo caso aplica una división de C en 2 clústeres usando un algoritmo k-means (Jain et al., 1999), con $k = 2$ (líneas 10 y 11). (2) Encuentra un clúster C en CS con el cual s tiene una admisibilidad parcial; es decir, el segmento cumple la relación de cercanía espacial y temporal con parte de algún o algunos segmentos de C . En este caso se aplica un proceso de clustering parcial según los casos planteados en Tabla 5.1. (3) No encuentra un clúster con quién asociar el segmento en cuyo caso crea un nuevo clúster con ese segmento y lo agrega al conjunto de clústeres CS .

Para determinar la admisibilidad parcial entre un segmento s y un clúster de segmentos C , se compara s respecto a los segmentos en C . Por simplicidad en la presentación, la Tabla 5.1 indica y explica las posible relaciones parciales entre s y un clúster, con su consecuente acción que realiza la función *PartialClustering* en línea 16 del Algoritmo 3.

El Algoritmo 4 genera trayectorias agregadas desde un conjunto de clústeres de segmentos CS y en base a la medida de compatibilidad definida en el parámetro Δ_J . Los clústeres son primero ordenados en base a su posición espacial respecto a su primera coordenada del eje horizontal, de tal manera de filtrar parcialmente la cercanía espacial (línea 2). Para proceder a agregar los

Algoritmo 3: AGRUPAMIENTO DE SEGMENTOS - “SegmentClustering”

Data: Una conjunto de clústeres CS , un segmento s , un umbral espacial Δ_S , un umbral temporal Δ_T

Result: El conjunto de clústeres CS actualizado y un conjunto de segmentos R

*/*Busca el conjunto de clústeres que cumplen cercanía espacial y temporal al segmento */*

```

1   $CS' \leftarrow \text{SpatialSelection}(s, CS, \Delta_S, \Delta_T)$ 
2   $C \leftarrow \emptyset$ 
3   $R \leftarrow \text{New SegmentSet}()$ 
4  if  $CS' \neq \emptyset$  then
5      foreach  $C \in CS'$  do
6          /*Verifica si existe compatibilidad Total Completa o Incompleta */
7           $B \leftarrow \text{TotalAdmissible}(s, C, \Delta_S, \Delta_T)$ 
8          if  $B = \text{'Completa'}$  then
9               $c \leftarrow \text{Add}(s, C)$  /*Agrega el segmento s al clúster c */
10             break
11         else if  $B = \text{'Incompleta'}$  then
12              $\text{TwoMeans}(s, C, \Delta_S, \Delta_T, \text{var } CS)$ 
13             break
14         /*Si no ha encontrado admisibilidad total  $C = \emptyset$  y busca admisibilidad parcial */
15         if  $C \neq \emptyset$  then
16             foreach  $C' \in CS'$  do
17                 if  $\text{PartialAdmissible}(s, C')$  then
18                      $R \leftarrow \text{PartialClustering}(s, C', \Delta_S, \Delta_T, \text{var } CS)$ 
19                     Break
20         if  $C = \emptyset$  then
21              $C' \leftarrow \text{NewSegmentSet}()$  /*Crea nuevo clúster */
22              $C' \leftarrow \text{Add}(s, C')$ 
23              $CS \leftarrow \text{Add}(C', CS)$ 
24 return  $R$ 

```

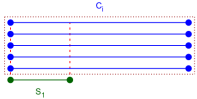
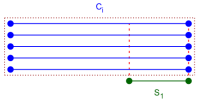
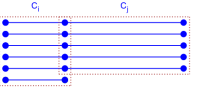
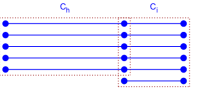
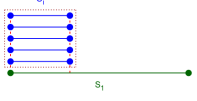
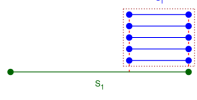
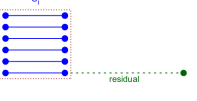
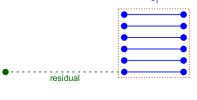
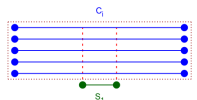
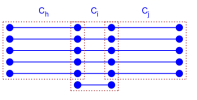
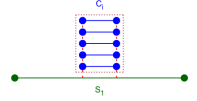
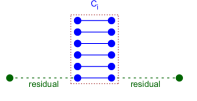
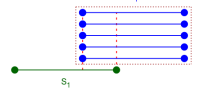
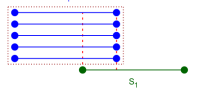
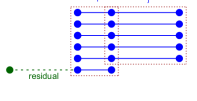
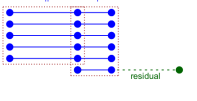
Caso	Resultado	Explicación
 	 	El clúster C_i se divide en dos clústeres C_i (que contiene a S_1) y C_j
 	 	El segmento S_1 se divide en dos sub-segmentos, uno de ellos se incorpora a C_i y el otro queda de residuo
		El clúster C_i se divide en tres clústeres y S_1 se incorpora a C_i
		El segmento S_1 se divide en tres sub-segmentos, dos de los cuales son residuos
 	 	El clúster C_i se divide en dos, uno incluye a una parte de S_1 y el resto de S_1 queda de residuo

Tabla 5.1: División de clústeres

clústeres dentro de CS' se analizan los clústeres de a pares. Dados dos clústeres $C \in CS'$ y $C' \in CS'$, si estos son compatibles y sus áreas se sobreponen (línea 7), entonces se genera un nuevo clúster de la agregación de C y C' (línea 8), quien reemplaza a C y C' de CS' . Dos clústeres son compatibles si el porcentaje de trayectorias en común es mayor al umbral de compatibilidad Δ_J . Si no son compatibles, se prueba con un nuevo par de clústeres.

Análisis de los algoritmos propuestos. Cada segmento en el Algoritmo 3 es comparado con todos

Algoritmo 4: AGREGACIÓN DE CLÚSTERES - “*Cluster Aggregation*”

Data: Conjunto de clústeres CS, umbral de compatibilidad Δ_J
Result: Conjunto de trayectorias Agregadas AT

```

1 AT ← New PatternSet() /*Crea un conjunto vacío de trayectorias agregadas */
2 SA ← New SpatialSortedArray(CS) /*Crea un arreglo ordenado por orden
   creciente de la coordenada en el eje X de cada clúster */
3 foreach  $i = 1 \rightarrow |SA| - 1$  do
   /*SA[i] = null cuando el clúster asociado ha sido agregado */
4   if SA[i] ≠ null then
5     foreach  $j = i + 1 \rightarrow |SA|$  do
6       if SA[j] ≠ null then
7         if Compatible(SA[i], SA[j]) ≥  $\Delta_J \wedge$ 
8           Intersects(Area(SA[i]), Area(SA[j])) then
9           AC ← Agregar(SA[i], SA[j]);
10          SA[j] ← null /*eliminación lógica */
11          SA[i] ← AC;
12           $j \leftarrow n_1 + 1$ ;
13           $i \leftarrow i - 1$  /*Vuelve a considerar el clúster agregado en la
              iteración */
14 foreach  $i = 1 \rightarrow n_1 - 1$  do
15   if SA[i] ≠ null then
16     AT ← AT ∪ SA[i];
17 return AT

```

los segmentos de los clústeres que estén espacialmente cercanos (línea 1 de Algoritmo), los que en el peor caso pueden ser todos los clústeres. El costo de evaluar la admisibilidad total (*TotalAdmissible* en línea 6 de Algoritmo 3) o admisibilidad parcial (*PartialAdmissible* en línea 15 de Algoritmo 3) es lineal con respecto al número de segmentos en un clúster. El uso de k-means (con $k=2$) tiene un costo también lineal respecto al número de segmentos en el clúster a dividir en dos, ya que el algoritmo converge en una segunda iteración. De esta forma el costo total del Algoritmo 3) *SegmentClustering* es acotado superiormente por $O(n)$, con n el número de segmento en CS.

El costo total del agrupamiento de segmentos en todas las trayectoria del Algoritmo 2 depende del costo de la función *SegmentClustering* en Algoritmo 3, la cual es llamada por cada segmento de trayectoria (línea 4) o por segmentos en el conjunto residuo (línea 7). El análisis teórico (asintótico) del algoritmo de agrupamiento de trayectorias indica que un segmento es comparado a lo más una vez con algún otro segmento, lo que da un número de comparaciones polinomial respecto al número de segmentos. El problema es que dada una entrada inicial de n segmentos (lo que considera los segmentos de todas las trayectorias), segmentos residuos pueden ser creados,

llevando a un crecimiento teórico exponencial en el número de segmentos. De hecho, un segmento puede teóricamente volver a dividirse en 1, 2 o 3 segmentos, y así sucesivamente, hasta que las características geométricas de los segmentos o la representación discreta del espacio no permita mayores subdivisiones. Afortunadamente, como se verá en la experimentación, este costo teórico no impide la ejecución satisfactoria del algoritmo.

El Algoritmo 4 toma clústeres en $\mathcal{CS}$ y los compara de pares de manera que un clúster es comparado a lo más con todos los otros clústeres. Notar que cada vez que se agregan dos clústeres, este nuevo clúster reemplaza al clúster en la posición de ordenamiento al clúster original con coordenada en el eje horizontal menor; es decir, el clúster en posición i del ciclo exterior del algoritmo. Aunque se tiene un nuevo clúster, se eliminan los clústeres agregados. En consecuencia, la agregación no altera el máximo número de comparaciones posteriores posibles entre clústeres, dando un costo superior de $O(m^2)$, con m el número de clústeres en $\mathcal{CS}$, y donde el costo de evaluar la compatibilidad entre clústeres se considera constante.

Capítulo 6

Resultados

En este capítulo se muestra la eficacia de nuestros algoritmos en términos de la calidad de los patrones encontrados.

Para la implementación y pruebas de los algoritmos propuestos, se utilizó un conjunto de datos sobre trayectorias de huracanes. Este conjunto de datos ha sido utilizado en artículos que presentan algoritmos para obtener patrones de trayectorias (Lee et al., 2007; Li et al., 2010) con los cuales comparamos nuestros resultados.

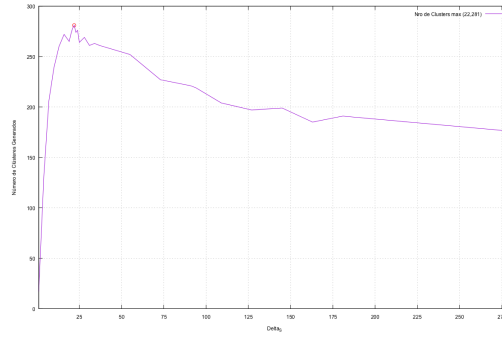
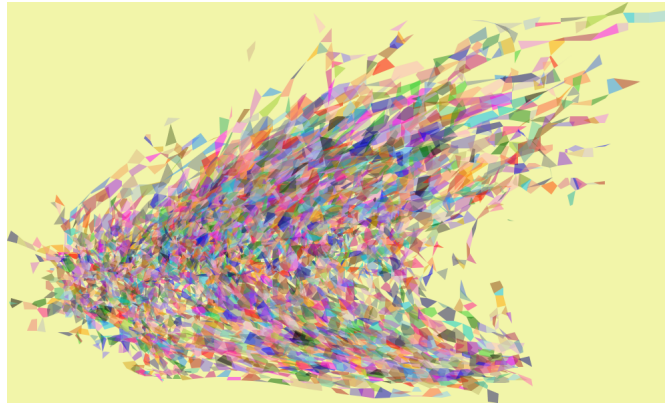
Los algoritmos se ejecutaron en un servidor con dos procesadores Intel Xeon E5620, cada uno de ellos cuenta con cuatro núcleos, 2,4 GHz de frecuencia y 12 MB de caché, además el servidor contaba con 61 GB de memoria RAM.

6.1. Descripción de Datos

Este conjunto de datos contiene 608 trayectorias de huracanes ocurridos en Estados Unidos entre 1970 y 2006. En total se almacenan 18.343 segmentos, con un promedio de 31 segmentos por trayectoria. El umbral espacial Δ_S utilizado para el cómputo de clústeres de segmentos es 22 unidades de medida, no se consideró umbral temporal, porque los datos de huracanes no consideran tiempo, se prueban diversos umbrales de compatibilidad.

Para determinar el umbral espacial en 22 unidades se llevo a cabo un experimento donde se probaron distintos umbrales espaciales, y se decidió por aquella medida que generaba una mayor cantidad de clústeres con más de un segmento. Un umbral menor al escogido, reduce la cantidad de relaciones entre segmentos, produciendo menos clústeres y aumentando el ruido al dejar segmentos sin poder formar clústeres. Un umbral mayor al escogido, relaciona segmentos más distantes, provocando que los clústeres no se ajusten a las características reales de las trayectorias, ya que tendrían más área de lo que realmente tiene las trayectorias.

La Figura 6.1 muestra la gráfica para distintos valores del umbral espacial Δ_S . Como se aprecia en la gráfica, cuando el umbral toma el valor 22 se obtiene la mayor cantidad de clústeres para el conjunto de trayectorias de huracanes. La Figura 6.2 muestra los 4,532 clústeres de segmentos de trayectoria obtenidos al utilizar $\Delta_S = 22$ sobre el conjunto de trayectorias de huracanes.


 Figura 6.1: Número de Clústeres para distintos valores de Δ_S

 Figura 6.2: Clústeres obtenidos para $\Delta_S = 22$ y $\Delta_T = 0$

6.2. Trayectorias Agregadas

En esta sección presentamos las trayectorias agregadas que se obtienen considerando distintos umbrales de compatibilidad. La compatibilidad corresponde a un porcentaje de trayectorias en común entre los clústeres de segmentos dentro de un clúster agregado.

La Tabla 6.1 muestra el detalle de los umbrales de compatibilidad que fueron experimentados para el conjunto CS de 4.532 clústeres formado por las 608 trayectorias de entrada de huracanes. En la tabla se especifica además la cantidad de trayectorias agregadas (TA) obtenidas para cada umbral, el número de clústeres sin agregar (NAC) y la proporción de trayectorias agregadas obtenidas, es decir, número total de trayectorias agregadas dividido por la suma de trayectorias agregadas y clústeres sin agregar ($\frac{TA}{TA+NAC}$).

Trayectorias Agregadas con Compatibilidad 0 % En esta prueba el umbral de compatibilidad Δ_J es igual a cero, y por lo tanto, todos los clústeres que cumplan con la condición de

Δ_j	# TA	# NAC	$\frac{TA}{TA+NAC}$
0 %	1	2	33,3 %
5 %	192	270	41,6 %
10 %	369	275	57,3 %
15 %	685	321	68,1 %
20 %	880	425	67,4 %
25 %	1034	730	58,6 %
30 %	1042	1492	41,6 %
35 %	728	2539	22,3 %
40 %	710	2657	21,1 %
45 %	567	3144	15,3 %
50 %	557	3180	14,9 %
55 %	354	3727	8,7 %
60 %	333	3774	8,1 %
...	...	...	... %
...	...	...	... %
...	...	...	... %
80 %	123	4262	2,8 %
90 %	112	4285	2,5 %
100 %	112	4285	2,5 %

Tabla 6.1: Resultados para distintos umbrales de compatibilidad (Δ_J)

traslape de áreas asociadas, quedan en una misma trayectoria agregada. La Figura 6.3 muestra el resultado de este experimento donde se genera 1 trayectoria agregada (con identificador 11977) y quedan 2 clústeres sin agregar (con identificadores 5228 y 1621).

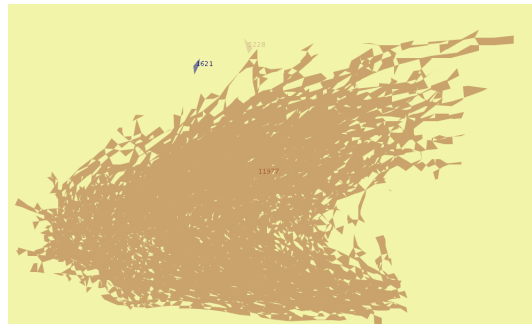


Figura 6.3: Trayectorias agregadas para $\Delta_J = 0$ %

Trayectorias Agregadas con Compatibilidad 15 % El valor $\Delta_J = 15\%$ genera una mayor proporción de trayectorias agregadas (68 %), generándose 685 trayectorias agregadas y quedando 321 clústeres de segmentos sin agregar. La Figura 6.4 muestra este resultado.

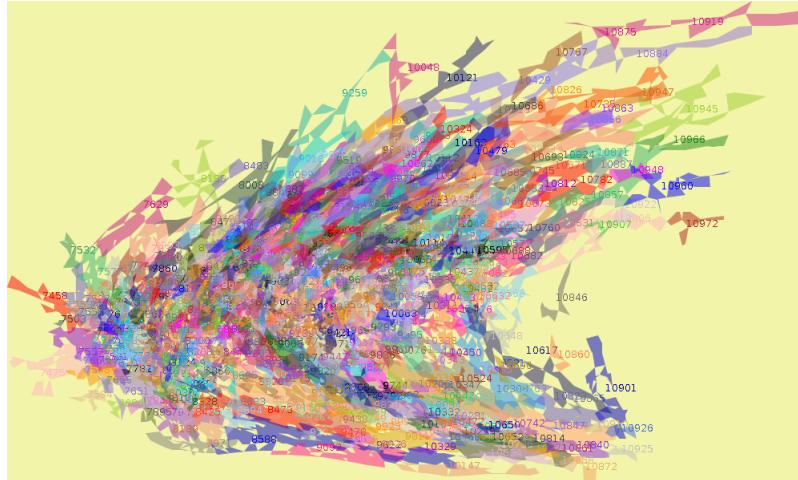


Figura 6.4: Trayectorias agregadas para $\Delta_J = 15\%$

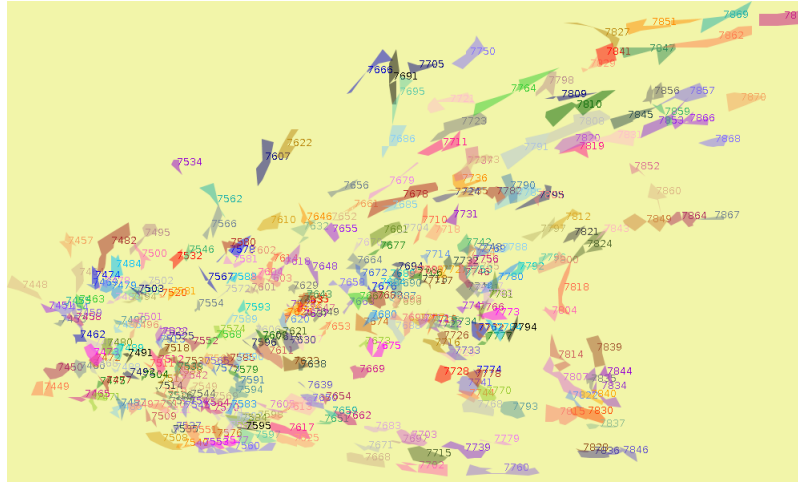
Trayectorias Agregadas con Compatibilidad 60 % De acuerdo a la Tabla 6.1, un valor de Δ_J superior a 35 % nos entrega una menor proporción de trayectorias agregadas, respecto de la suma de trayectorias agregadas y clústeres no agregados. La Figura 6.5 muestra las 333 trayectorias agregadas que se obtienen con $\Delta_J = 60\%$ las que corresponden a un 8,1 % del total de trayectorias agregadas obtenidas más los clústeres sin agregar.

Trayectorias Agregadas con Compatibilidad 100 % En esta prueba, el objetivo es maximizar el efecto del umbral de compatibilidad Δ_J , de manera de que para que dos clústeres puedan ser agregados, estos deben cumplir con la condición de traslape de áreas asociadas y además deben estar formados en base a las mismas trayectorias. La Figura 6.6 muestra las 112 trayectorias agregadas obtenidas para este umbral de compatibilidad.

6.2.1. Comparación con Estado del Arte

La Figura 6.7 presenta los resultados obtenidos en (Lee et al., 2007; Li et al., 2010), donde se muestran patrones de comportamiento de las diferentes 608 trayectorias de huracanes donde se obtienen 8 patrones de trayectoria.

Es posible obtener un resultado similar al reportado en (Lee et al., 2007; Li et al., 2010) con nuestros algoritmos aplicando un filtro a las trayectorias agregadas obtenidas. Consideremos las trayectorias agregadas obtenidas con el umbral de compatibilidad del 15 %, el cual produce una mayor proporción de trayectorias agregadas (ver Figura 6.4). Podemos filtrar o eliminar

Figura 6.5: Trayectorias agregadas para $\Delta_J = 60\%$ Figura 6.6: Trayectorias agregadas para $\Delta_J = 100\%$

aquellas trayectorias agregadas que contengan menos de un número determinado de trayectorias individuales que la conforman, valor dado por N en cada trayectoria agregada (ver Definición 4.8).

La Figura 6.8 muestra el resultado de filtrar aquellas trayectorias agregadas con un $N \geq 20$ de las trayectorias agregadas obtenidas con un umbral de compatibilidad del 15%. Se observan los 8 patrones obtenidos en (Lee et al., 2007; Li et al., 2010) para el mismo conjunto de trayectorias de entrada.

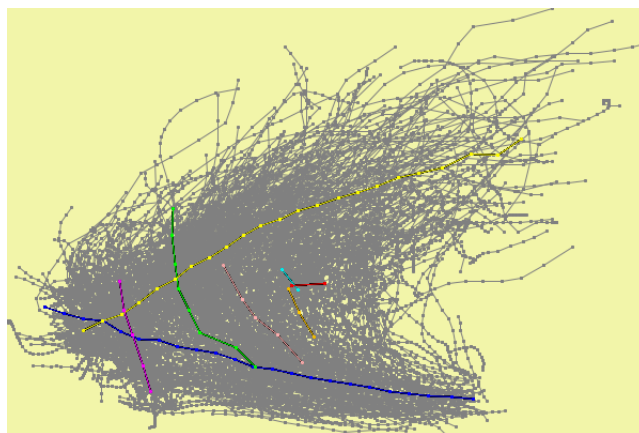


Figura 6.7: Patrones de trayectorias de huracanes (Lee et al., 2007; Li et al., 2010)

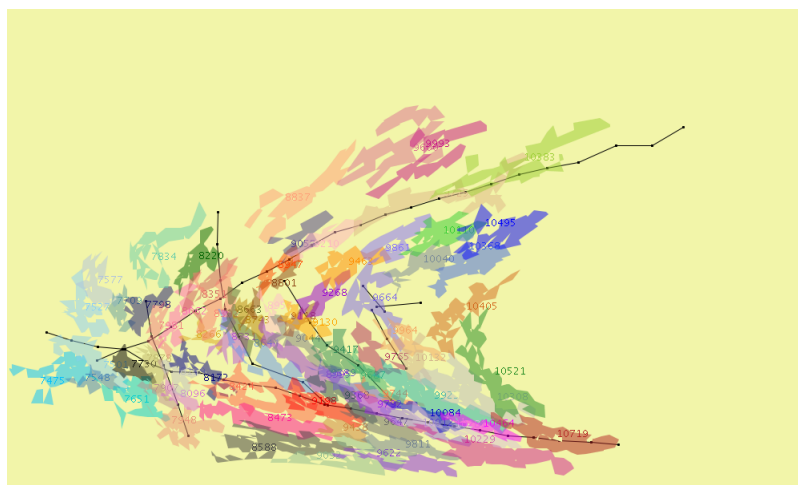


Figura 6.8: Trayectorias agregadas filtradas

Capítulo 7

Conclusión

En esta tesis presentamos un nuevo concepto de trayectoria agregada, el cual corresponde a la agregación de clústeres de segmentos de trayectorias, con una densidad relevante.

Los algoritmos que obtienen trayectorias agregadas trabajan en dos partes. Primero se obtienen clústeres de segmentos de trayectorias diferentes, para los cuales es necesario definir un umbral espacial y temporal. Luego de computar clústeres de segmentos de trayectorias se procede a obtener clústeres agregados considerando un umbral de compatibilidad, el cual puede variar en diferentes escenarios.

Los algoritmos propuestos en esta tesis permiten la generación dinámica de clústeres de segmentos de trayectorias. Es decir, es posible incorporar una nueva trayectoria al procesamiento de clústeres de segmentos aunque el proceso ya haya terminado, sin comenzar desde cero.

En términos de visualización, las trayectorias agregadas se representan a través de la generación de un área de los clústeres de segmentos mediante el cómputo del cierre convexo (convexhull). Esto puede luego ser llevado a representaciones mediante el skeleton de cada región, extrayendo trayectorias representativas de una agregación de trayectorias.

En esta tesis hemos mostrado que nuestros algoritmos entregan clústeres similares a los obtenidos por algoritmos del estado del arte. En particular, para el caso de los huracanes, las trayectorias agregadas con umbral de compatibilidad 15 % obtenidas por nuestros algoritmos, son similares al resultado mostrado en (Lee et al., 2007), donde se muestran agrupamientos de trayectorias. Cabe destacar que el trabajo presentado en (Lee et al., 2007) no considera la dimensión temporal como lo consideran nuestros algoritmos.

La posibilidad de lograr un algoritmo dinámico y que pueda considerar sub segmentos de trayectorias viene con un alto costo computacional. Cada segmento debe en el peor caso ser comparado con cada uno de los otros segmentos, segmentos de trayectorias originales o segmentos que hayan sido creados en procesos de partición durante el agrupamiento. Sin embargo, el peor caso teórico no ocurre en la práctica.

Como trabajo futuro, se propone la experimentación con otros conjuntos de datos espacio-temporales. Es necesario lograr una optimización del proceso de agrupamiento que permita una mayor escalabilidad del sistema cuando el número de trayectorias crece. Así mismo, analizar nuevas medidas del grado de agregación de trayectorias para poder discriminar la calidad de la agregación obtenida.

Referencias

- Natalia V. Andrienko y Gennady L. Andrienko. Spatial generalization and aggregation of massive movement data. *IEEE Trans. Vis. Comput. Graph.*, 17(2):205–219, 2011.
- Andre Salvaro Furtado, Renato Fileto, y Chiara Renso. M-attract: Assessing the attractiveness of places by using moving objects trajectories data. En *XIII Brazilian Symposium on Geoinformatics November 25-27 2012, Campos do Jordão, São Paulo, Brazil*, págs. 84–95. 2012.
- Fosca Giannotti, Mirco Nanni, y Dino Pedreschi. Efficient mining of temporally annotated sequences. En *Proceedings of the Sixth SIAM International Conference on Data Mining*, págs. 348–359. 2006.
- Fosca Giannotti, Mirco Nanni, Fabio Pinelli, y Dino Pedreschi. Trajectory pattern mining. En *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, págs. 330–339. 2007.
- Chih-Chieh Hung, Wen-Chih Peng, y Wang-Chien Lee. Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *The VLDB Journal*, 24(2):169–192, 2015.
- Anil K. Jain, M. Narasimha Murty, y Patrick J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, 1999.
- Jae-Gil Lee, Jiawei Han, y Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. En *SIGMOD Conference*, págs. 593–604. 2007.
- Zhenhui Li, Jae-Gil Lee, Xiaolei Li, y Jiawei Han. Incremental clustering for trajectories. En *Proceedings of the 15th International Conference on Database Systems for Advanced Applications*, págs. 32–46. Springer-Verlag, 2010.
- Nirvana Meratnia y Rolf A. de By. Aggregation and comparison of trajectories. En *Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, págs. 49–54. 2002.
- Salvatore Orlando, Renzo Orsini, Alessandra Raffaetà, Alessandro Roncato, y Claudio Silvestri. Trajectory data warehouses: Design and implementation issues. *Journal of Computing Science and Engineering*, 1(2):211–232, 2007.

-
- Swaminathan Sankararaman, Pankaj K. Agarwal, Thomas Mølhave, Jiangwei Pan, y Arnold P. Boedihardjo. Model-driven matching and segmentation of trajectories. En *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, págs. 234–243. 2013.
- Haozhou Wang, Han Su, Kai Zheng, Shazia Wasim Sadiq, y Xiaofang Zhou. An Effectiveness study on trajectory similarity measures. En *Twenty-Fourth Australasian Database Conference*, págs. 13–22. 2013.