

Estudio de métodos de detección de patrones de movimiento para sistemas de recomendación turísticos.

Sergio Torrijos López de la Manzanara



TABLA DE CONTENIDOS

1. Introducción

2. Diseño e
Implementación

3. Pruebas y
resultados

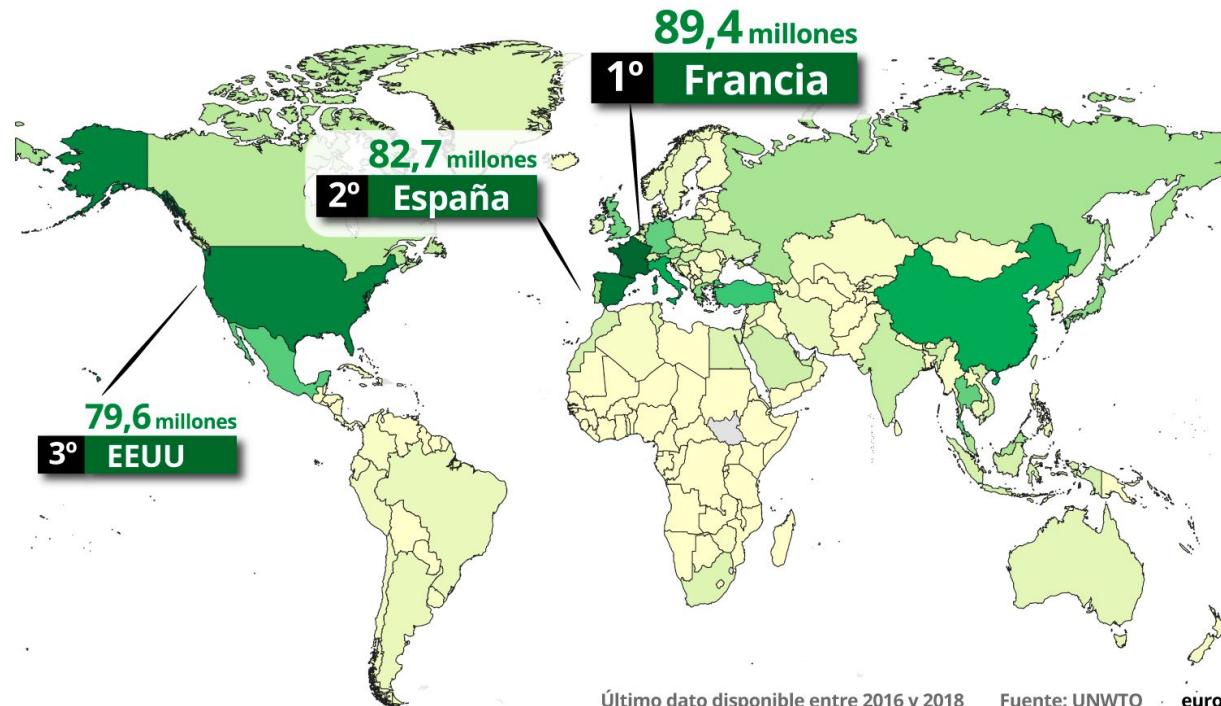
4. Conclusión y
trabajo futuro



1. INTRODUCCIÓN

1. INTRODUCCIÓN

Datos turismo mundial



1. INTRODUCCIÓN

Sistemas de recomendación turísticos

El personal muy atento y disponible para cualquier cosa

restaurants

Volver a los resultados

Me gusta Compartir

Ana SM 7 reseñas

★★★★★ Hace un mes
Volver a Roma para volver a este restaurante. Los camareros son geniales y la comida espectacular. Deseando volver. Muy recomendable!

Me gusta Compartir

Gabriel Diaz Guimaraens 7 reseñas

★★★★★ Hace un mes
Un restaurante mágico con un trato genial de los camareros. Pasta fresca con albóndigas de ensueño y un chupito de limoncello para finalizar. Pasé dos noches en Roma y las dos cené aquí, estoy deseando volver!

Me gusta Compartir

[Ver todas las reseñas](#)

Otras personas también buscan

La Fraschetta di C... Italiana 4,5 ★ (1.680)

I Sofá Bar Restaur... Italiana 4,5 ★ (245)

Pizzeria Tonnarello Italiana 4,3 ★ (1.060)

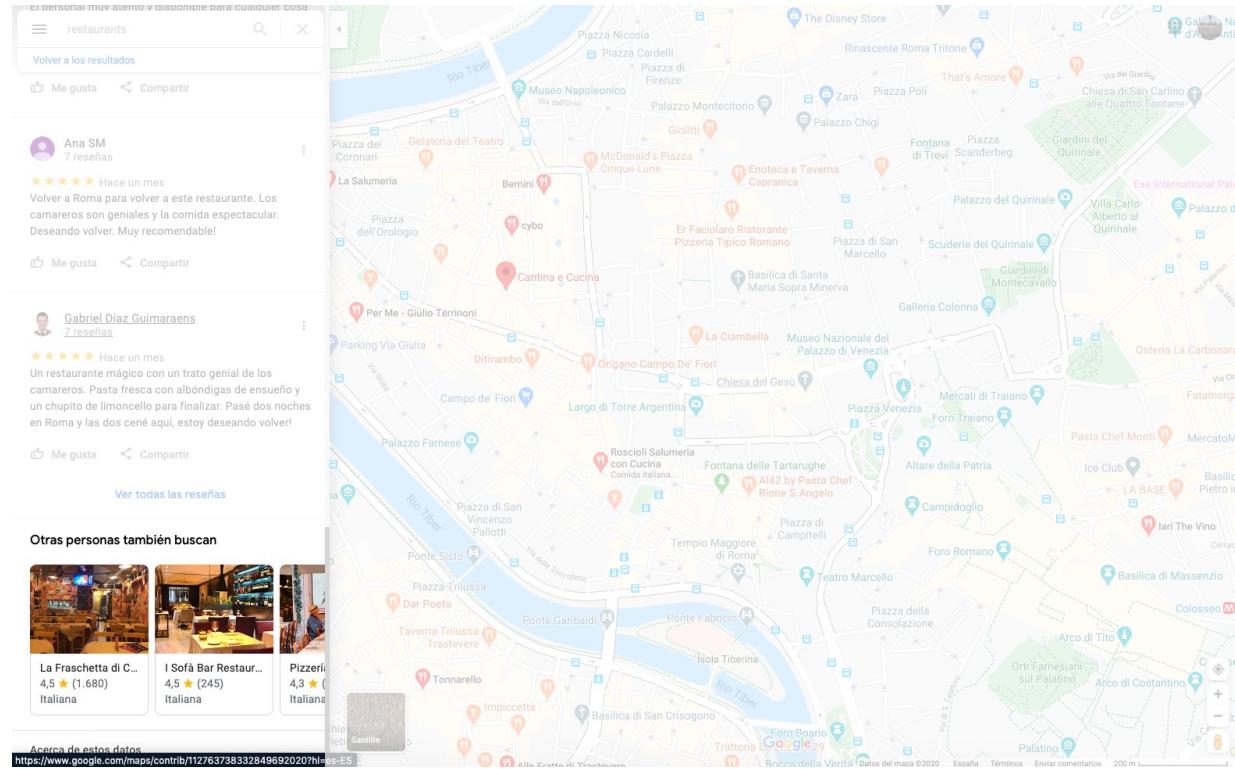
[Acerca de estos datos](#)

<https://www.google.com/maps/contrib/1127637383328496920?hl=es-ES>

This image shows a Google Maps interface for Rome, Italy. On the left, there is a sidebar displaying three restaurant reviews. The reviews are from users Ana SM and Gabriel Diaz Guimaraens, both of whom have given 5-star ratings. The sidebar also includes a link to 'Ver todas las reseñas' (View all reviews) and a section for 'Otras personas también buscan' (Other people also search for) with three more restaurant suggestions. On the right, the main map of Rome is shown, highlighting various landmarks and areas. The map includes labels for the Tiber River, the Colosseum, the Pantheon, the Roman Forum, and numerous piazzas and monuments. Numerous red location markers are scattered across the map, indicating the locations of the restaurants mentioned in the reviews. The map also shows green shaded areas representing parks and green spaces.

1. INTRODUCCIÓN

Sistemas de recomendación turísticos



El personal muy atento y disponible para cualquier cosa

restaurants

Volver a los resultados

Me gusta Compartir

Ana SM 7 reseñas

★★★★★ Hace un mes
Volver a Roma para volver a este restaurante. Los camareros son geniales y la comida espectacular. Deseando volver. Muy recomendable!

Me gusta Compartir

Gabriel Diaz Guimaraens 7 reseñas

★★★★★ Hace un mes
Un restaurante mágico con un trato genial de los camareros. Pasta fresca con albóndigas de ensueño y un chupito de limoncello para finalizar. Pasé dos noches en Roma y las dos cené aquí, estoy deseando volver!

Me gusta Compartir

Ver todas las reseñas

Otras personas también buscan

La Fraschetta di C... 4,5 ★ (1.680) Italiana

I Sofá Bar Restaur... 4,5 ★ (245) Italiana

Pizzeria... 4,3 ★ (1.000) Italiana

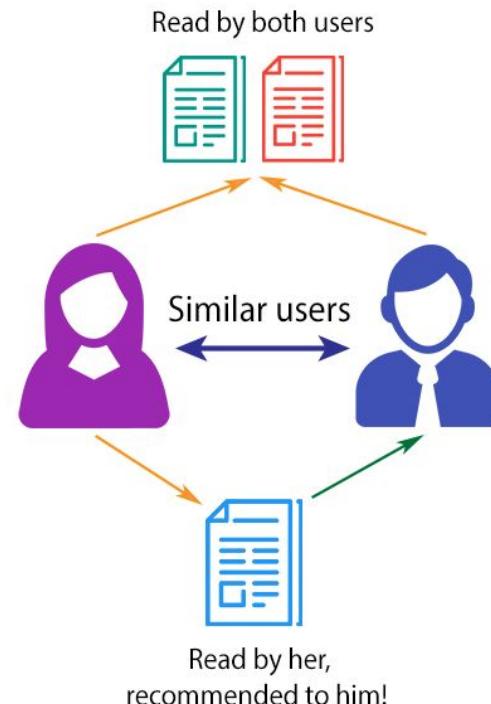
Acerca de estos datos

<https://www.google.com/maps/contrib/11276373833284969207hl=es-ES>

1. INTRODUCCIÓN

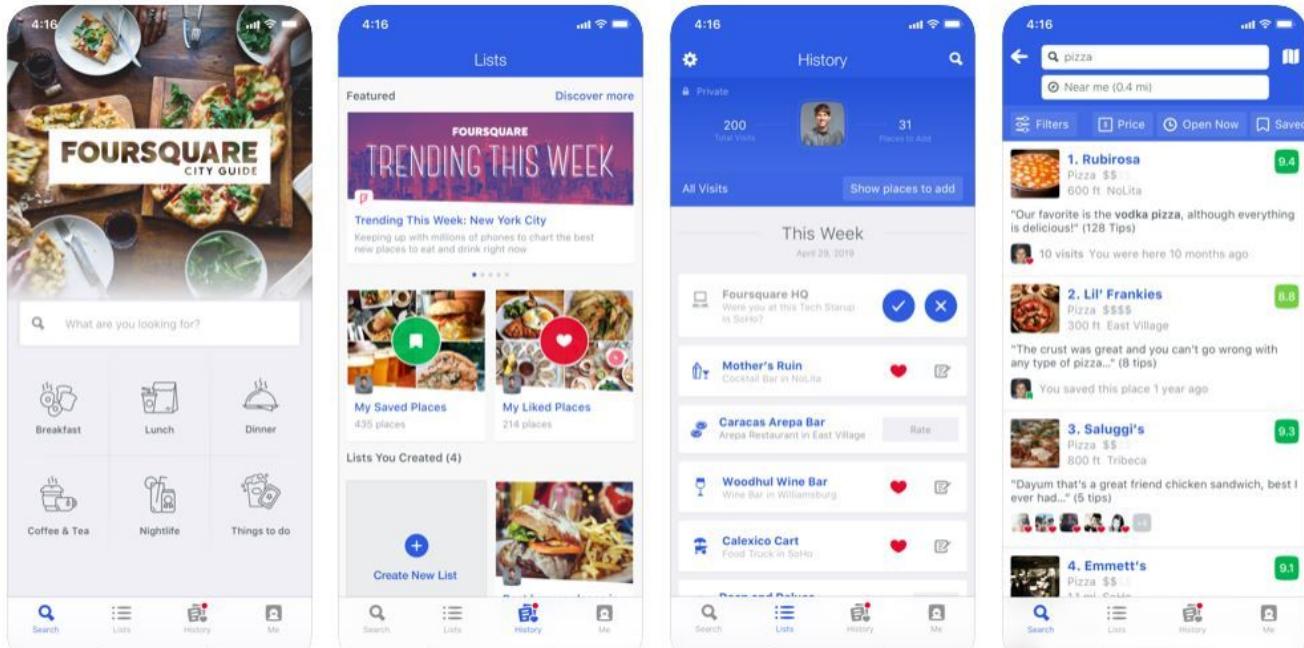
Sistemas de recomendación

COLLABORATIVE FILTERING



1. INTRODUCCIÓN

Sistemas de recomendación turísticos



1. INTRODUCCIÓN

¿Bandada de pájaros = Patrones de movimiento?



1. INTRODUCCIÓN

Tema de estudio

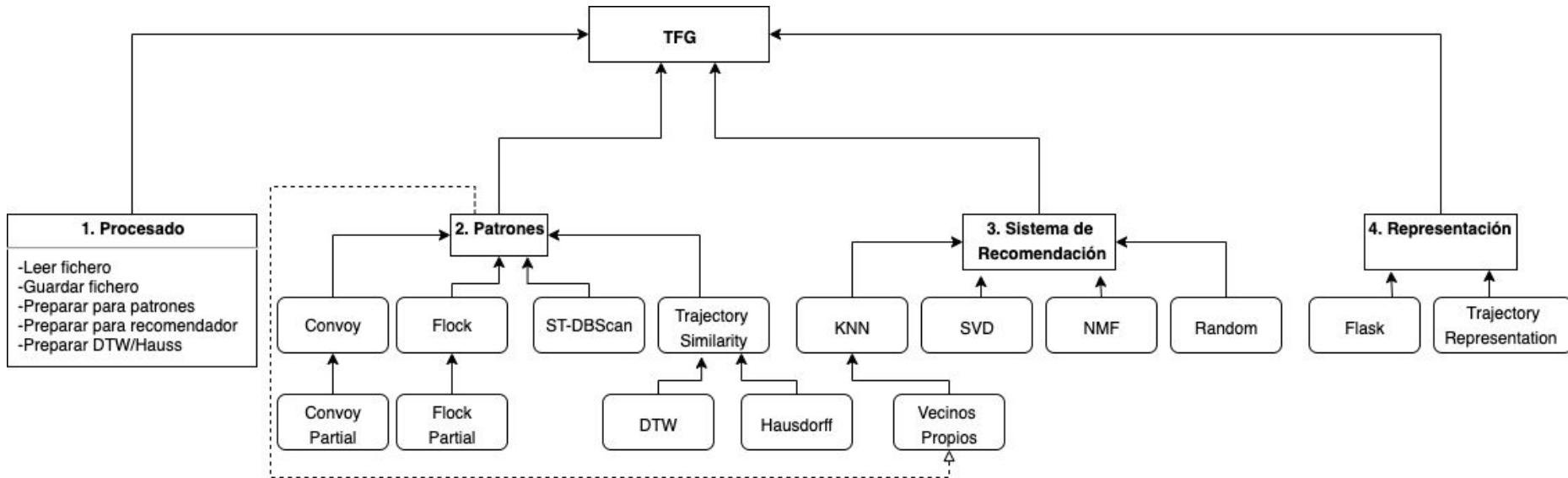
Estudiar, aplicar y adaptar los patrones de movimiento para los sistemas de recomendación.



2. DISEÑO E IMPLEMENTACIÓN

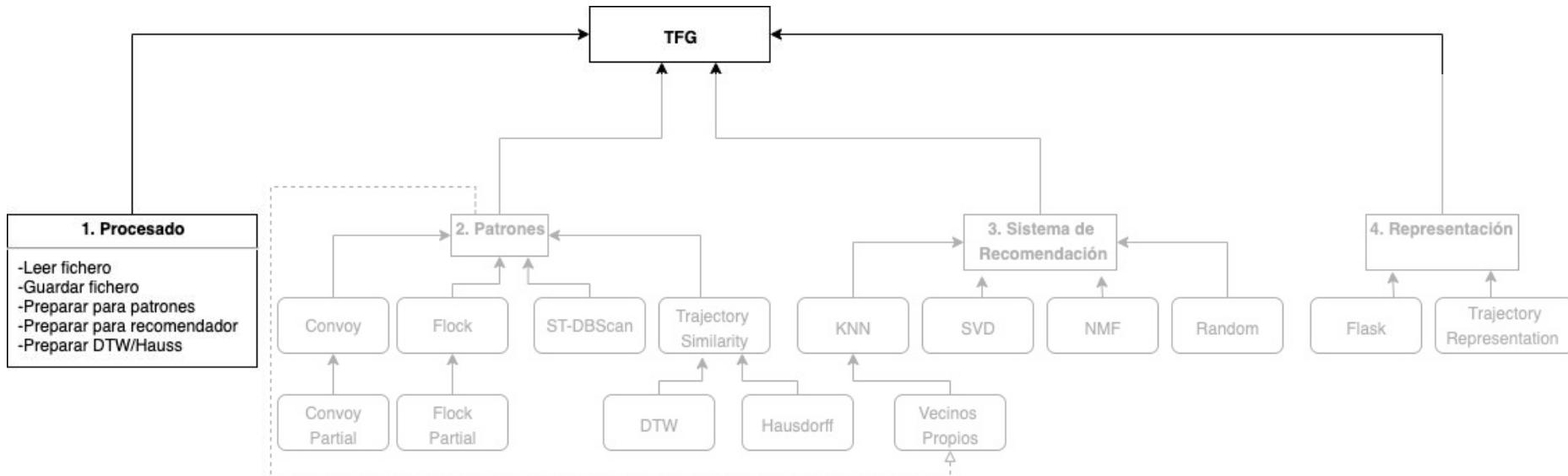
2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



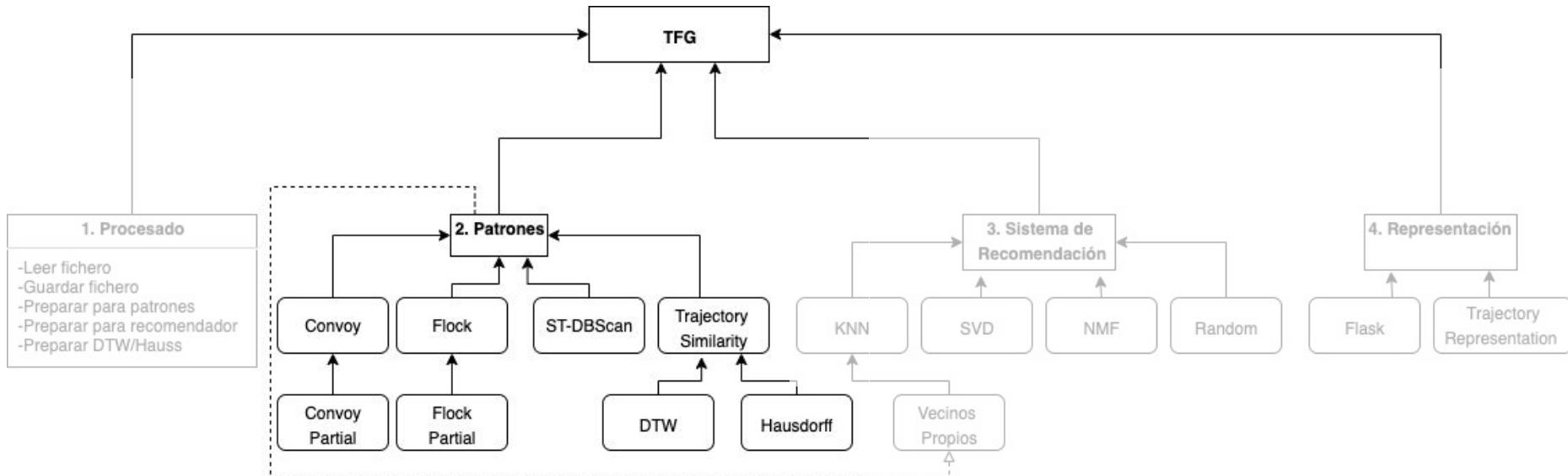
2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



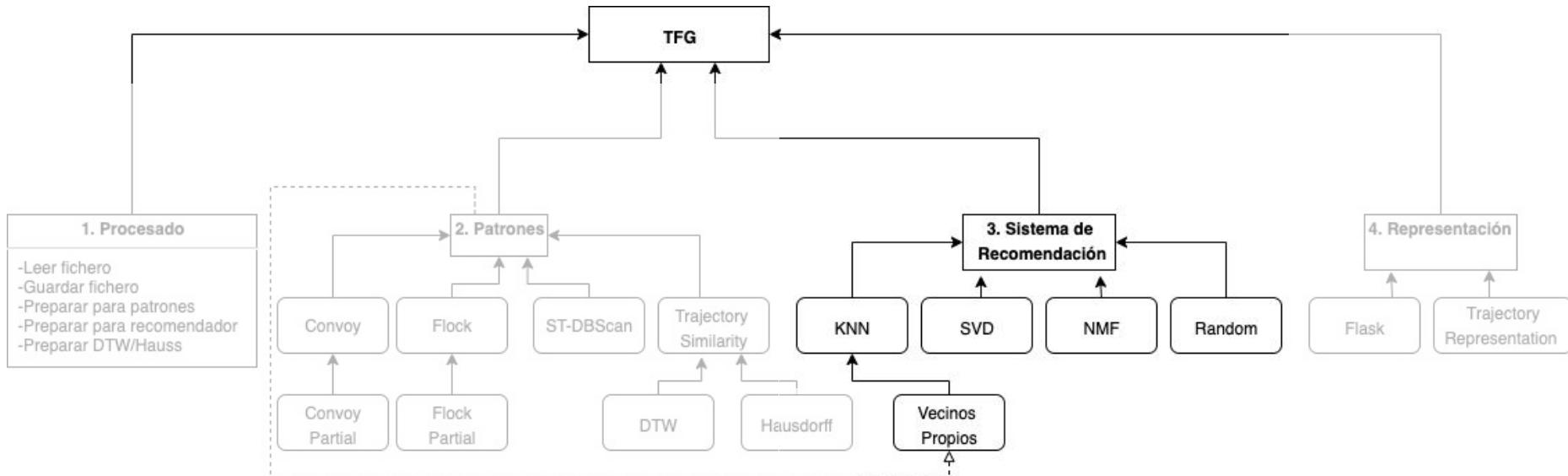
2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



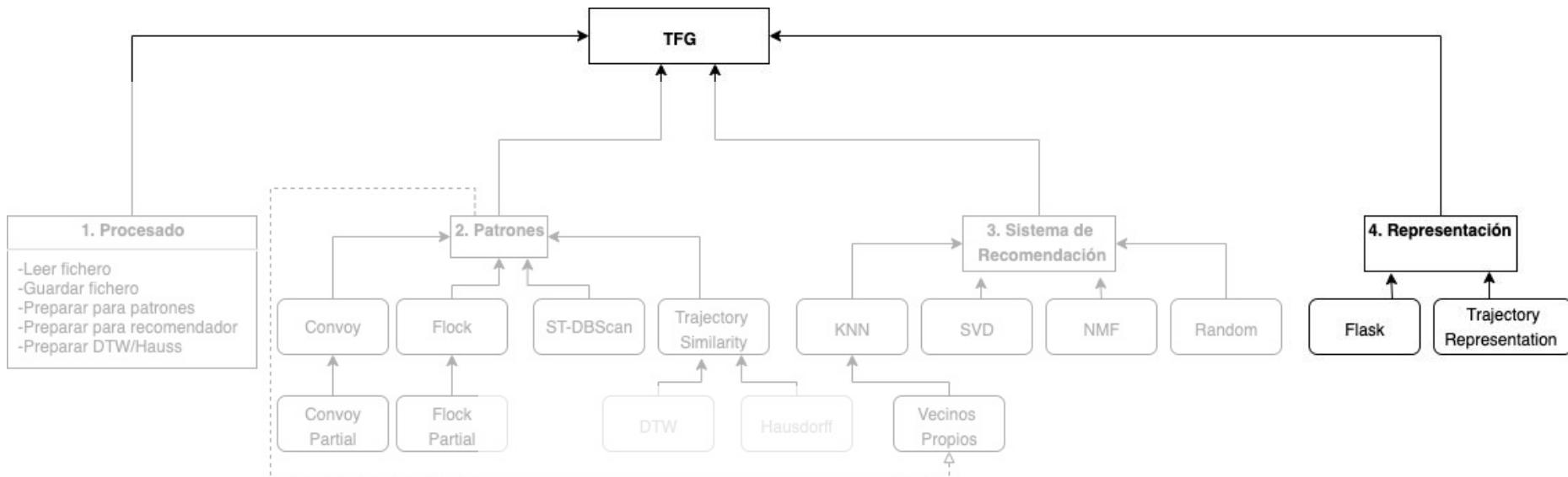
2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



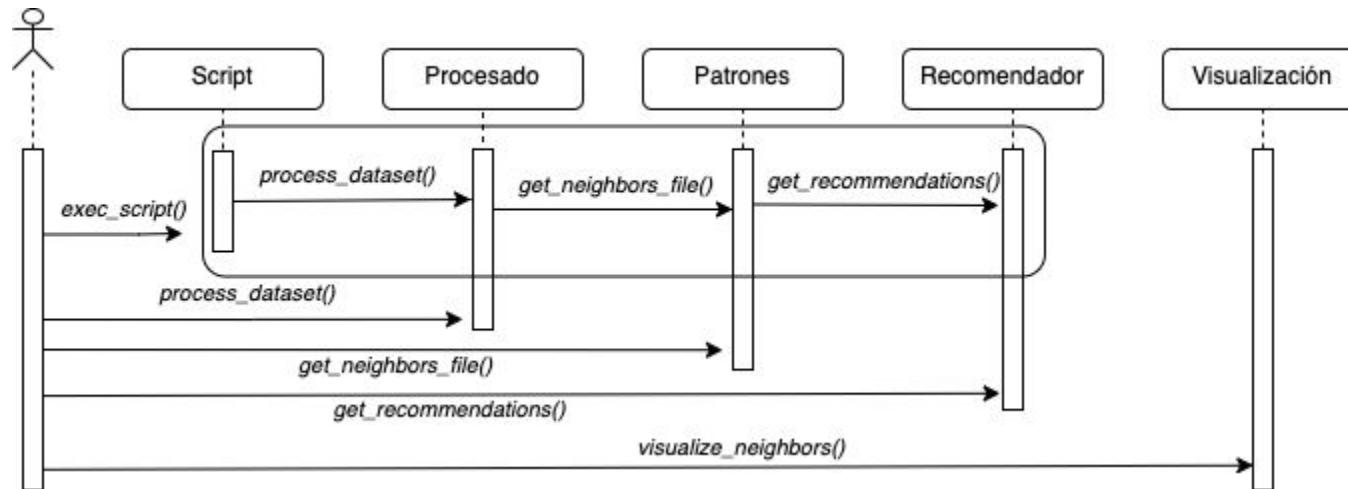
2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



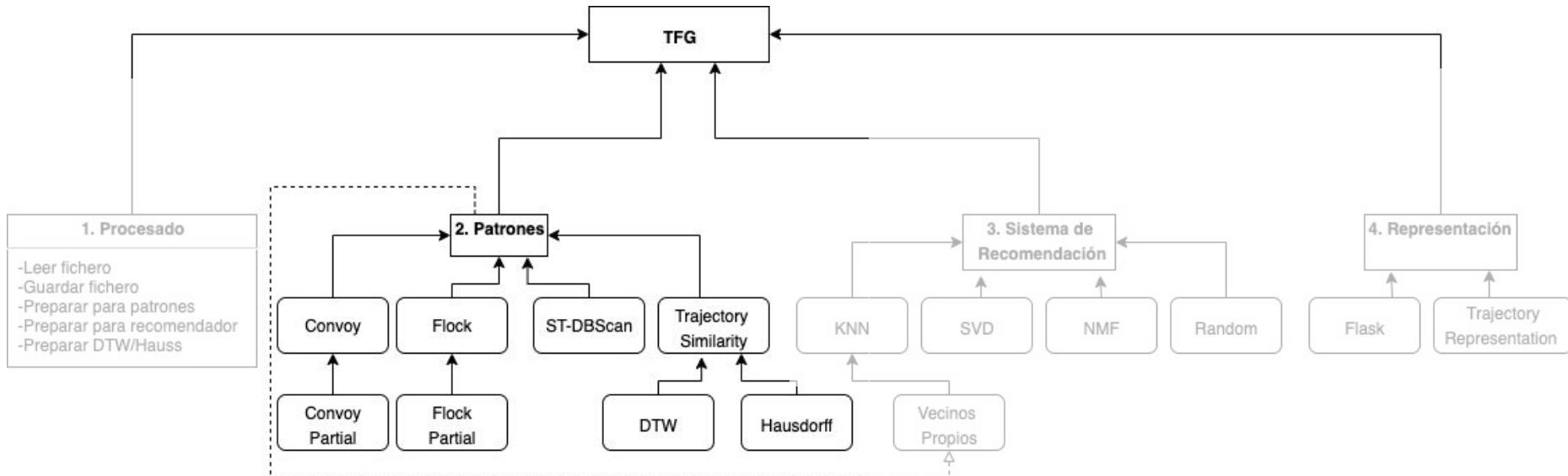
2. DISEÑO E IMPLEMENTACIÓN

Diseño - Diagrama de secuencia



2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



2. DISEÑO E IMPLEMENTACIÓN

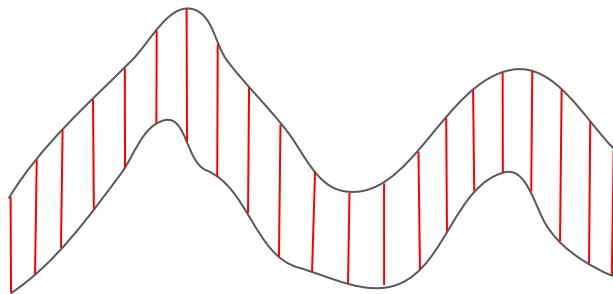
Patrones de movimiento



2. DISEÑO E IMPLEMENTACIÓN

Patrones de movimiento - Dynamic Time Warping

(a) Distancia euclídea



(b) Dynamic Time Warping

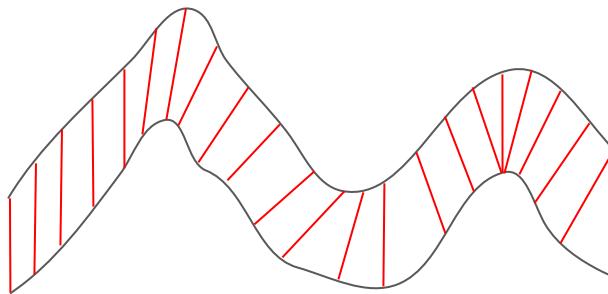
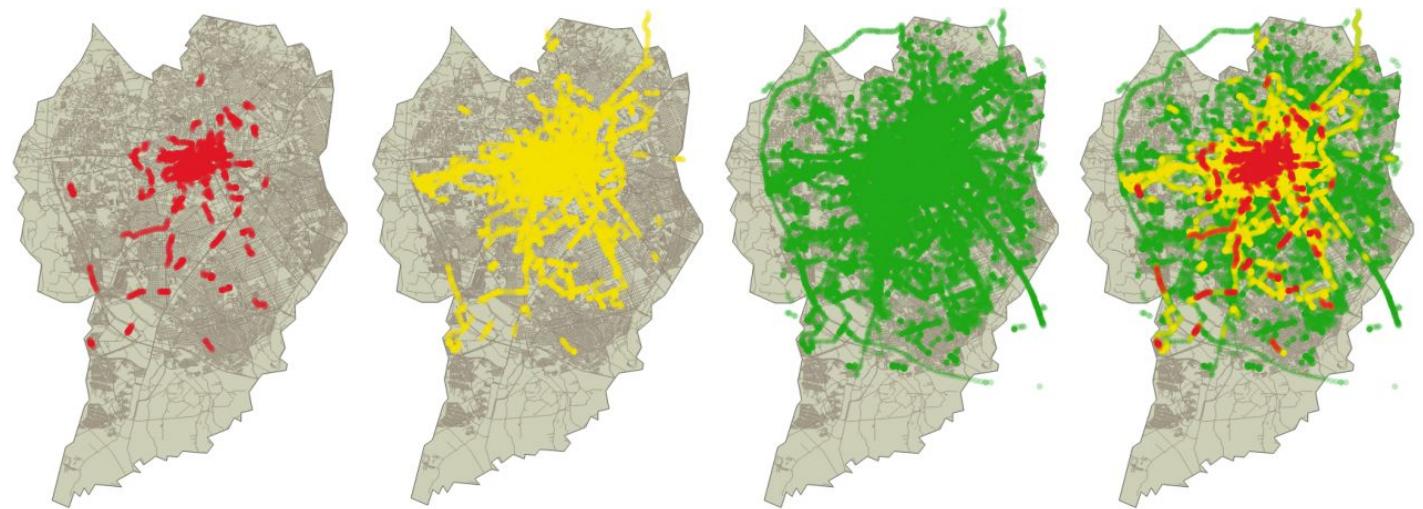


Figura 2.5: Comparación entre dos secuencias: (a) mientras la distancia euclídea es rígida en el tiempo, (b) Dynamic Time Warping (DTW) es flexible en el tiempo para tratar la posible distorsión de tiempo entre las secuencias [10].

2. DISEÑO E IMPLEMENTACIÓN

Implementación - ST-DBSCAN



(a) $Eps1 = 100m$

(b) $Eps1 = 166m$

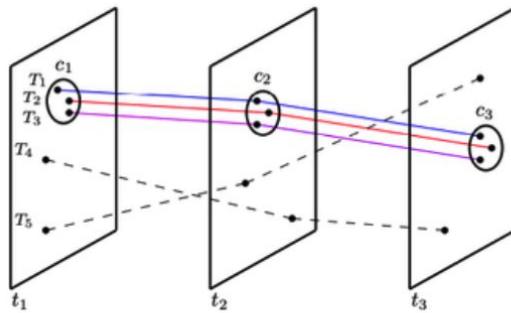
(c) $Eps1 = 330m$

(d) $Eps1 = 330, 166, 100m$

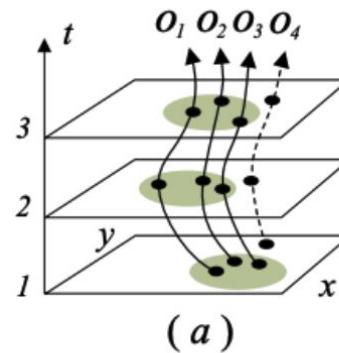
Birant, D. and Kut, A. (2007). St-dbscan: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60(1):208–221. Intelligent Data Mining.

2. DISEÑO E IMPLEMENTACIÓN

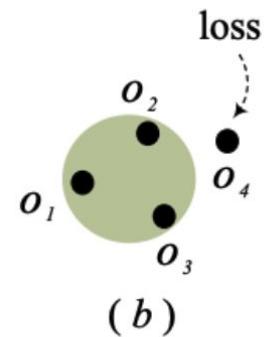
Patrones de movimiento - Flock



(a) Trayectoria Flock [7] donde se muestran cómo se escogen los puntos para la trayectoria en función de si cumplen el criterio del radio del disco.



(a)



(b) Demostración de la pérdida de elementos ocasionada por Flock [13], donde se aprecia que el punto O_4 no entra en las restricciones de Flock pese a seguir una trayectoria similar.

2. DISEÑO E IMPLEMENTACIÓN

Patrones de movimiento - Convoy

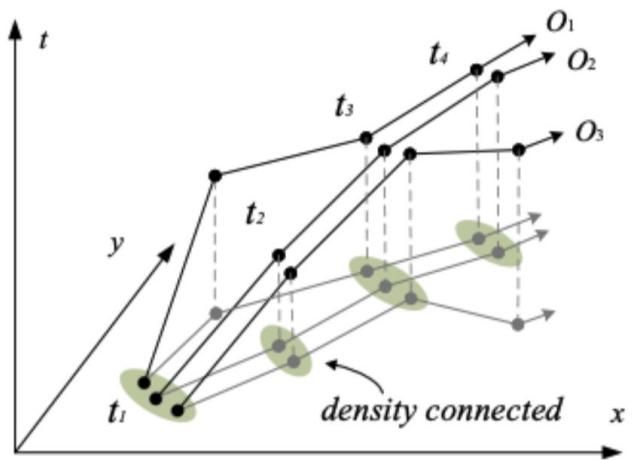
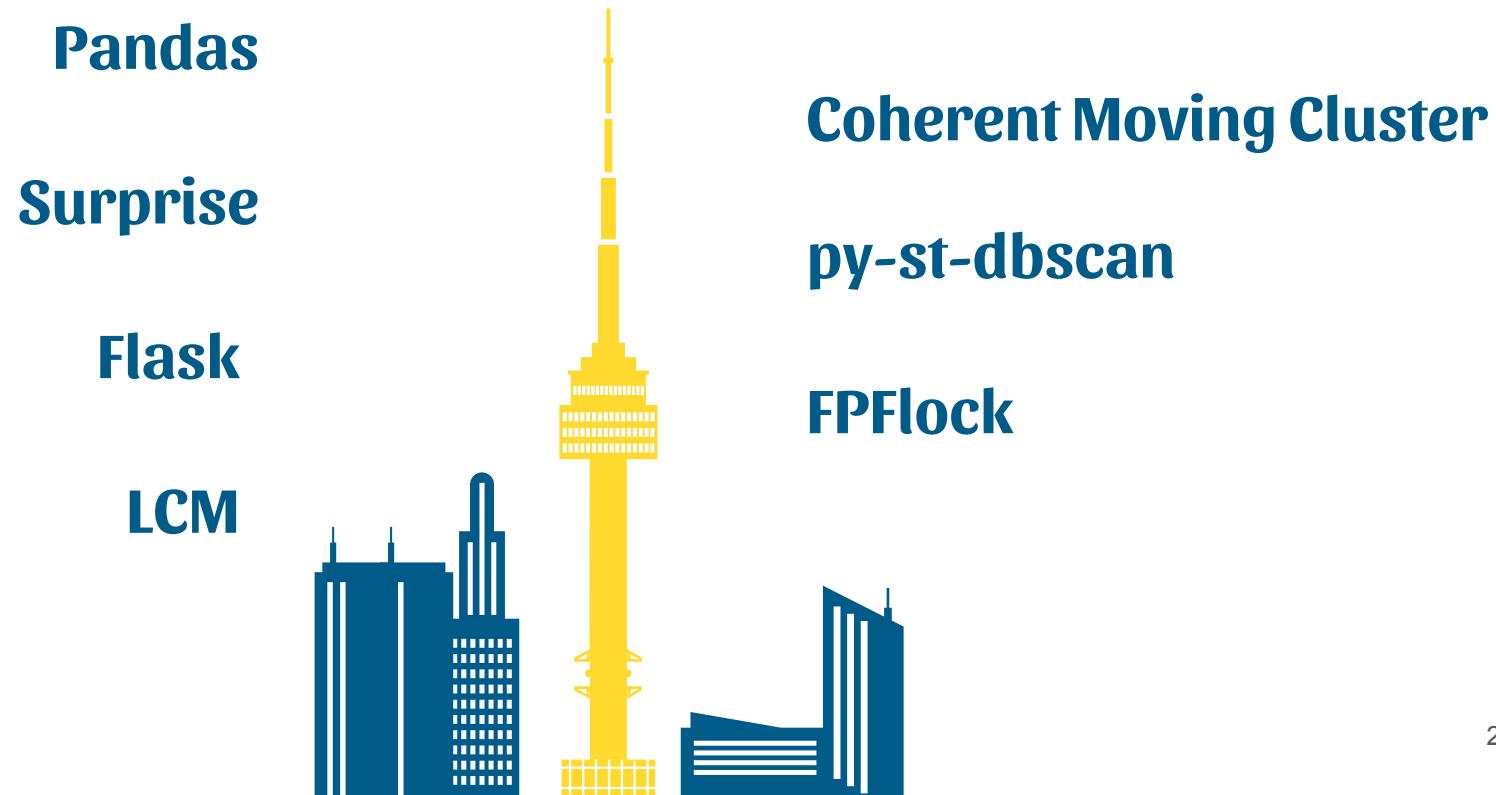


Figura 2.2: Ejemplo de Trayectoria Convoy [13], marcando en verde los discos que agrupan los puntos y las líneas marcando la trayectoria.

2. DISEÑO E IMPLEMENTACIÓN

Patrones de movimiento - Librerías utilizadas



2. DISEÑO E IMPLEMENTACIÓN

Patrones de movimiento - Librerías utilizadas

Librerías implementadas y adaptadas a la caracterización de recomendación turística.

2. DISEÑO E IMPLEMENTACIÓN

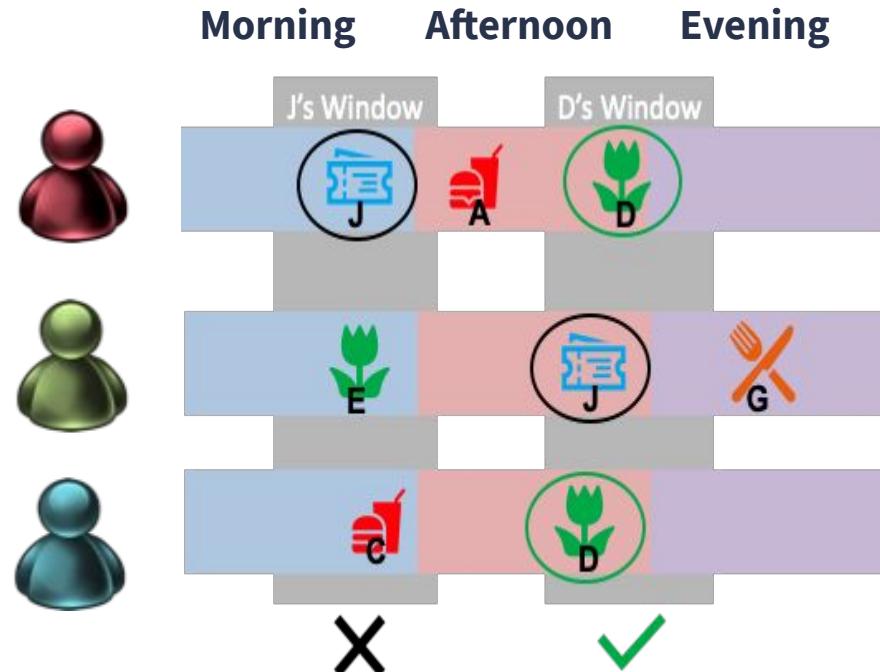
Implementación optimizada de Flock: Partials

Ejecución en 2 fases:

- 1.- Calcular la similitud de todos los usuarios mediante un método rápido (DTW o ST-DBSCAN).
- 2.- Sobre los resultados obtenidos, coger los k usuarios más similares a nuestro usuario, y ejecutar Flock iterativamente la lista reducida.

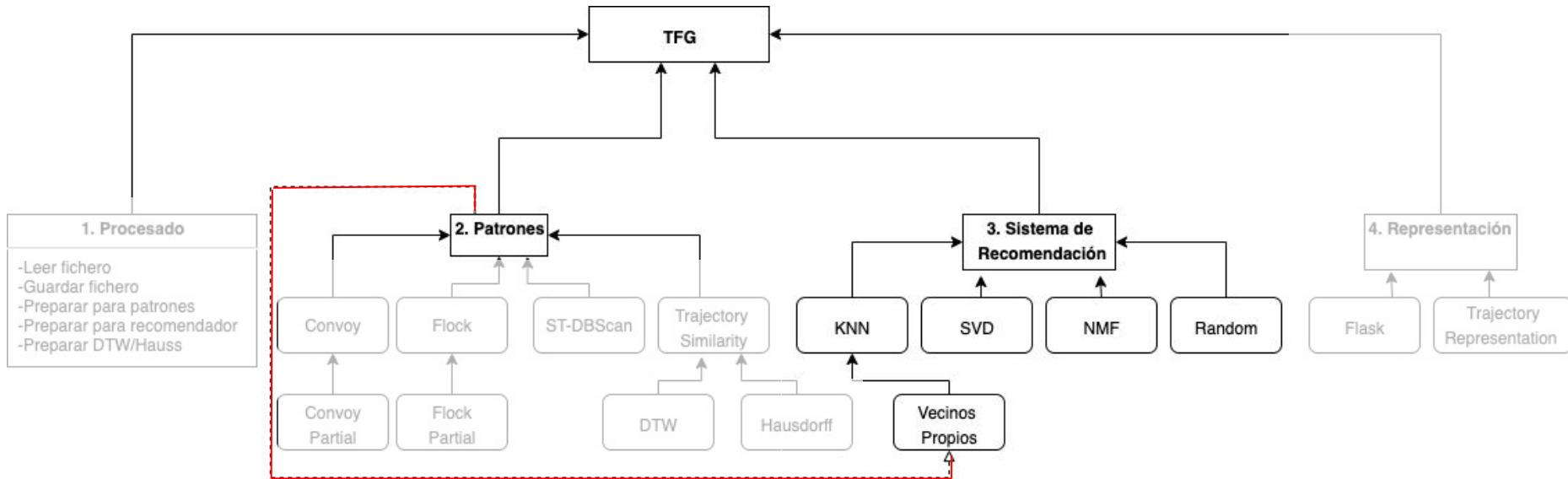
2. DISEÑO E IMPLEMENTACIÓN

Nuevo patrón de movimiento: Ad-Hoc



2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



2. DISEÑO E IMPLEMENTACIÓN

Implementación - Entrada al Sistema de Recomendación

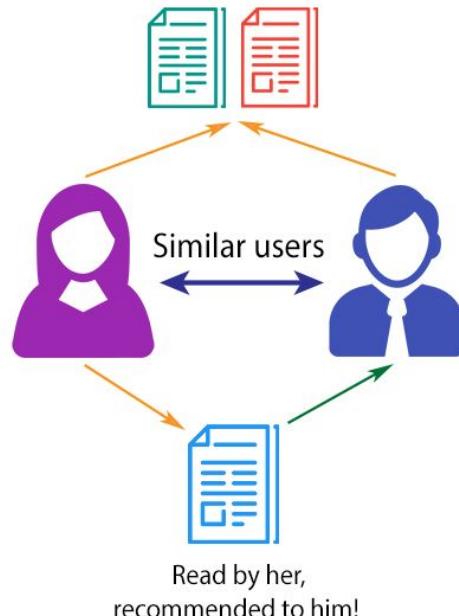
```
1          user1_id user2_id occurrences
2              2           21        1.0
3              21          39        2.0
```

2. DISEÑO E IMPLEMENTACIÓN

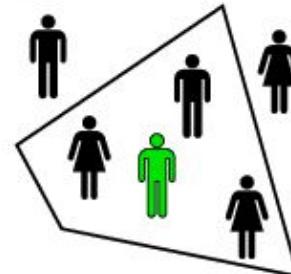
Implementación - Sistema de Recomendación

COLLABORATIVE FILTERING

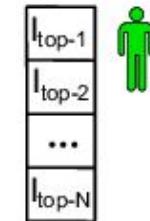
Read by both users



Target user neighborhood formation

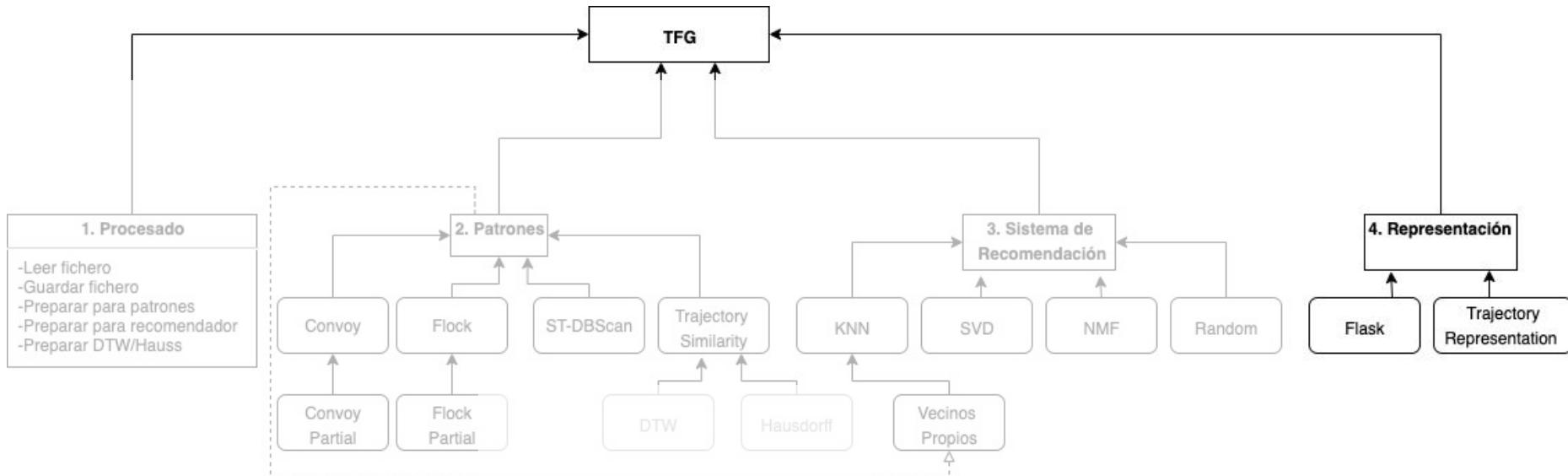


Top-N items for target user



2. DISEÑO E IMPLEMENTACIÓN

Diseño - Estructura general



2. DISEÑO E IMPLEMENTACIÓN

Implementación - Visualización

The image shows a web application interface for "Trajectory Mining" running on a local server at 127.0.0.1:5000. The application includes a file upload feature, a map of New York City with a highlighted trajectory, and a table of user and neighbor data.

File Upload: A "Select files for upload..." input field with a "Browse..." button and a "File TAG" button.

Map: A map of New York City showing a trajectory highlighted in red and blue. The map includes labels for various neighborhoods and landmarks.

File Browser: A "WEB" browser window showing the file system. It lists several text files:

- romeTempTrain_p ost.txt
- romeTempTrain.tx t_adhoc_3600.txt
- romeTempTrain.tx t_conv_artials.txt

Tags: A legend for color-coded tags:

- Red
- Purple
- Blue
- Important
- Work
- Home
- Gray
- All Tags...

Users: A table showing user data:

User ID
296
537
565
652

Neighbors: A table showing neighbor data:

KN	Neighbors	▼
Neighbor ID	Similarity	
2228	1	
2073	1	
2066	1	
1991	1	



3. PRUEBAS Y RESULTADOS

3. PRUEBAS Y RESULTADOS

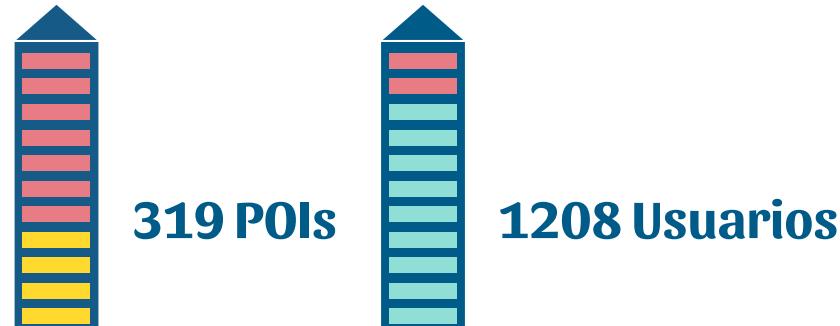
Esquema de pruebas realizadas



3. PRUEBAS Y RESULTADOS

Dataset: Roma

1	user_id	item_id	latitude	longitude	timestamp
2	15	14627	40.757564	-73.989238	1354881399



3. PRUEBAS Y RESULTADOS

1. Tiempos de ejecución

Algorithm	1K	10K	20K	60K
Ad-Hoc	0m0.477s	0m1.952s	0m5.253s	0m35.758s
Sim. tray. Hausdorff	0m10.568s	22m32.164s	88m17.315s	1157m31.251s
Sim. tray. DTW	0m3.386s	6m16.519s	26m9.038s	232m25.592s
ST-DBSCAN	0m0.817s	0m3.173s	0m9.487s	1m29.065s
Convoy	0m5.786s	0m6.086s	0m6.266s	0m6.098s
Convoy partials	0m2.786s	0m3.086s	0m3.266s	0m4.098s
Flock	0m5.199s	24m23.087s	49m34.065s	†
Flock partials	0m5.500s	0m30.545s	0m14.880s	3m11.811s

Dataset	1K	10K	20K	60K
Número POIs	163	319	346	394
Número usuarios	129	1208	2618	7954

3. PRUEBAS Y RESULTADOS

2. Número promedio de vecinos

Algoritmo	Parámetros	Número vecinos promedio
Ad-Hoc	$\delta=3600$	1.3
	$\delta=10800$	1.8
Sim. tray.	Hausdorff	1206
	DTW	1206
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	1.7
	$\varepsilon=5000, th=6000, minEps=5$	3.1
	$\varepsilon=4000, th=2000, minEps=1$	2.2
	$\varepsilon=1000, th=2000, minEps=1$	1.5
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	22.8
Convoy	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	2.1
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	1.23
	$min\ points=5, lifetime=2, dist\ max=10^{-2}$	5.92
	$min\ points=5, lifetime=2, dist\ max=10^{-15}$	6
	$min\ points=3, lifetime=5, dist\ max=10^2$	0
	$min\ points=2, lifetime=2, dist\ max=10^{-4}$	1.4
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	1.7
	$\varepsilon=10, \mu=2, \delta=2$	4
Flock	$\varepsilon=0.1, \mu=2, \delta=10$	2
	$\varepsilon=10, \mu=2, \delta=10^2$	0
	$\varepsilon=0.1, \mu=2, \delta=2$	4
	$\varepsilon=10, \mu=2, \delta=2$	5.2
Flock partials	$\varepsilon=0.1, \mu=2, \delta=10$	5.09
	$\varepsilon=0.1, \mu=2, \delta=2$	8.04

3. PRUEBAS Y RESULTADOS

2. Número promedio de vecinos

Algoritmo	Parámetros	Número vecinos promedio
Ad-Hoc	$\delta=3600$	1.3
	$\delta=10800$	1.8
Sim. tray.	Hausdorff	1206
	DTW	1206
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	1.7
	$\varepsilon=5000, th=6000, minEps=5$	3.1
	$\varepsilon=4000, th=2000, minEps=1$	2.2
	$\varepsilon=1000, th=2000, minEps=1$	1.5
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	22.8
Convoy	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	2.1
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	1.23
	$min\ points=5, lifetime=2, dist\ max=10^{-2}$	5.92
	$min\ points=5, lifetime=2, dist\ max=10^{-15}$	6
	$min\ points=3, lifetime=5, dist\ max=10^2$	0
	$min\ points=2, lifetime=2, dist\ max=10^{-4}$	1.4
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	1.7
	$\varepsilon=10, \mu=2, \delta=2$	4
Flock	$\varepsilon=0.1, \mu=2, \delta=10$	2
	$\varepsilon=10, \mu=2, \delta=10^2$	0
	$\varepsilon=0.1, \mu=2, \delta=2$	4
	$\varepsilon=10, \mu=2, \delta=2$	5.2
Flock partials	$\varepsilon=0.1, \mu=2, \delta=10$	5.09
	$\varepsilon=0.1, \mu=2, \delta=2$	8.04

3. PRUEBAS Y RESULTADOS

2. Número promedio de vecinos

Algoritmo	Parámetros	Número vecinos promedio
Ad-Hoc	$\delta=3600$	1.3
	$\delta=10800$	1.8
Sim. tray.	Hausdorff	1206
	DTW	1206
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	1.7
	$\varepsilon=5000, th=6000, minEps=5$	3.1
	$\varepsilon=4000, th=2000, minEps=1$	2.2
	$\varepsilon=1000, th=2000, minEps=1$	1.5
Convoy	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	22.8
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	2.1
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	1.23
	$min\ points=5, lifetime=2, dist\ max=10^{-2}$	5.92
	$min\ points=5, lifetime=2, dist\ max=10^{-15}$	6
	$min\ points=3, lifetime=5, dist\ max=10^2$	0
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-4}$	1.4
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	1.7
Flock	$\varepsilon=10, \mu=2, \delta=2$	4
	$\varepsilon=0.1, \mu=2, \delta=10$	2
	$\varepsilon=10, \mu=2, \delta=10^2$	0
	$\varepsilon=0.1, \mu=2, \delta=2$	4
Flock partials	$\varepsilon=10, \mu=2, \delta=2$	5.2
	$\varepsilon=0.1, \mu=2, \delta=10$	5.09
	$\varepsilon=0.1, \mu=2, \delta=2$	8.04

3. PRUEBAS Y RESULTADOS

2. Número promedio de vecinos

Algoritmo	Parámetros	Número vecinos promedio
Ad-Hoc	$\delta=3600$	1.3
	$\delta=10800$	1.8
Sim. tray.	Hausdorff	1206
	DTW	1206
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	1.7
	$\varepsilon=5000, th=6000, minEps=5$	3.1
	$\varepsilon=4000, th=2000, minEps=1$	2.2
	$\varepsilon=1000, th=2000, minEps=1$	1.5
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	22.8
Convoy	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	2.1
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	1.23
	$min\ points=5, lifetime=2, dist\ max=10^{-2}$	5.92
	$min\ points=5, lifetime=2, dist\ max=10^{-15}$	6
	$min\ points=3, lifetime=5, dist\ max=10^2$	0
	$min\ points=2, lifetime=2, dist\ max=10^{-4}$	1.4
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	1.7
	$\varepsilon=10, \mu=2, \delta=2$	4
Flock	$\varepsilon=0.1, \mu=2, \delta=10$	2
	$\varepsilon=10, \mu=2, \delta=10^2$	0
	$\varepsilon=0.1, \mu=2, \delta=2$	4
	$\varepsilon=10, \mu=2, \delta=2$	5.2
Flock partials	$\varepsilon=0.1, \mu=2, \delta=10$	5.09
	$\varepsilon=0.1, \mu=2, \delta=2$	8.04

3. PRUEBAS Y RESULTADOS

2. Número promedio de vecinos

Algoritmo	Parámetros	Número vecinos promedio
Ad-Hoc	$\delta=3600$	1.3
	$\delta=10800$	1.8
Sim. tray.	Hausdorff	1206
	DTW	1206
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	1.7
	$\varepsilon=5000, th=6000, minEps=5$	3.1
	$\varepsilon=4000, th=2000, minEps=1$	2.2
	$\varepsilon=1000, th=2000, minEps=1$	1.5
Convoy	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	22.8
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	2.1
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	1.23
	$min\ points=5, lifetime=2, dist\ max=10^{-2}$	5.92
	$min\ points=5, lifetime=2, dist\ max=10^{-15}$	6
	$min\ points=3, lifetime=5, dist\ max=10^2$	0
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-4}$	1.4
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	1.7
Flock	$\varepsilon=10, \mu=2, \delta=2$	4
	$\varepsilon=0.1, \mu=2, \delta=10$	2
	$\varepsilon=10, \mu=2, \delta=10^2$	0
	$\varepsilon=0.1, \mu=2, \delta=2$	4
Flock partials	$\varepsilon=10, \mu=2, \delta=2$	5.2
	$\varepsilon=0.1, \mu=2, \delta=10$	5.09
	$\varepsilon=0.1, \mu=2, \delta=2$	8.04

3. PRUEBAS Y RESULTADOS

2. Número promedio de vecinos

Algoritmo	Parámetros	Número vecinos promedio
Ad-Hoc	$\delta=3600$	1.3
	$\delta=10800$	1.8
Sim. tray.	Hausdorff	1206
	DTW	1206
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	1.7
	$\varepsilon=5000, th=6000, minEps=5$	3.1
	$\varepsilon=4000, th=2000, minEps=1$	2.2
	$\varepsilon=1000, th=2000, minEps=1$	1.5
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	22.8
Convoy	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	2.1
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	1.23
	$min\ points=5, lifetime=2, dist\ max=10^{-2}$	5.92
	$min\ points=5, lifetime=2, dist\ max=10^{-15}$	6
	$min\ points=3, lifetime=5, dist\ max=10^2$	0
	$min\ points=2, lifetime=2, dist\ max=10^{-4}$	1.4
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	1.7
Flock	$\varepsilon=10, \mu=2, \delta=2$	4
	$\varepsilon=0.1, \mu=2, \delta=10$	2
	$\varepsilon=10, \mu=2, \delta=10^2$	0
	$\varepsilon=0.1, \mu=2, \delta=2$	4
Flock partials	$\varepsilon=10, \mu=2, \delta=2$	5.2
	$\varepsilon=0.1, \mu=2, \delta=10$	5.09
	$\varepsilon=0.1, \mu=2, \delta=2$	8.04

3. PRUEBAS Y RESULTADOS

2. Número promedio de vecinos

Algoritmo	Parámetros	Número vecinos promedio
Ad-Hoc	$\delta=3600$	1.3
	$\delta=10800$	1.8
Sim. tray.	Hausdorff	1206
	DTW	1206
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	1.7
	$\varepsilon=5000, th=6000, minEps=5$	3.1
	$\varepsilon=4000, th=2000, minEps=1$	2.2
	$\varepsilon=1000, th=2000, minEps=1$	1.5
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	22.8
Convoy	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	2.1
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	1.23
	$min\ points=5, lifetime=2, dist\ max=10^{-2}$	5.92
	$min\ points=5, lifetime=2, dist\ max=10^{-15}$	6
	$min\ points=3, lifetime=5, dist\ max=10^2$	0
	$min\ points=2, lifetime=2, dist\ max=10^{-4}$	1.4
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	1.7
	$\varepsilon=10, \mu=2, \delta=2$	4
Flock	$\varepsilon=0.1, \mu=2, \delta=10$	2
	$\varepsilon=10, \mu=2, \delta=10^2$	0
	$\varepsilon=0.1, \mu=2, \delta=2$	4
	$\varepsilon=10, \mu=2, \delta=2$	5.2
Flock partials	$\varepsilon=0.1, \mu=2, \delta=10$	5.09
	$\varepsilon=0.1, \mu=2, \delta=2$	8.04

3. PRUEBAS Y RESULTADOS

3. Efecto en la recomendación

Algoritmo	Parámetros	Precisión	MAE
Ad-Hoc	$\delta=3600$	0.0361754	0.3487827
	$\delta=10800$	0.0349337	0.3637682
Sim. tray.	Hausdorff	0.0175497	0.5784518
	DTW	0.0206126	1.2903128
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	0.0369205	0.4251326
	$\varepsilon=5000, th=6000, minEps=10$	0.0316225	0.3530357
	$\varepsilon=5000, th=6000, minEps=10$	0.0369205	0.4000756
	$\varepsilon=1000, th=2000, minEps=1$	0.0336921	0.3530357
	$\varepsilon=4000, th=2000, minEps=1$	0.0277318	0.3525041
Convoy	$min\ points=2, lifetime=10, dist\ max=10^{-2}$	0.0362583	0.3566276
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	0.0332781	1.0634231
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0243377	0.5973882
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	0.0352649	0.4018430
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0360927	0.3925327
	$min\ points=2, lifetime=2, dist\ max=10^{-1}$	0.0368377	0.3525041
	$min\ points=3, lifetime=5, dist\ max=10^{-1}$	0.0276490	0.7646296
Flock	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0369205	0.3704636
	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
Flock partials	$\varepsilon=10, \mu 2, \delta=2$	0.0368377	0.3525041
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0355132	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=2$	0.0368377	0.3670011

3. PRUEBAS Y RESULTADOS

3. Efecto en la recomendación

Algoritmo	Parámetros	Precisión	MAE
Ad-Hoc	$\delta=3600$	0.0361754	0.3487827
	$\delta=10800$	0.0349337	0.3637682
Sim. tray.	Hausdorff	0.0175497	0.5784518
	DTW	0.0206126	1.2903128
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	0.0369205	0.4251326
	$\varepsilon=5000, th=6000, minEps=10$	0.0316225	0.3530357
	$\varepsilon=5000, th=6000, minEps=10$	0.0369205	0.4000756
	$\varepsilon=1000, th=2000, minEps=1$	0.0336921	0.3530357
	$\varepsilon=4000, th=2000, minEps=1$	0.0277318	0.3525041
Convoy	$min\ points=2, lifetime=10, dist\ max=10^{-2}$	0.0362583	0.3566276
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	0.0332781	1.0634231
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0243377	0.5973882
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	0.0352649	0.4018430
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0360927	0.3925327
	$min\ points=2, lifetime=2, dist\ max=10^{-1}$	0.0368377	0.3525041
	$min\ points=3, lifetime=5, dist\ max=10^{-1}$	0.0276490	0.7646296
Flock	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0369205	0.3704636
	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
Flock partials	$\varepsilon=10, \mu 2, \delta=2$	0.0368377	0.3525041
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0355132	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=2$	0.0368377	0.3670011

3. PRUEBAS Y RESULTADOS

3. Efecto en la recomendación

Algoritmo	Parámetros	Precisión	MAE
Ad-Hoc	$\delta=3600$	0.0361754	0.3487827
	$\delta=10800$	0.0349337	0.3637682
Sim. tray.	Hausdorff	0.0175497	0.5784518
	DTW	0.0206126	1.2903128
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	0.0369205	0.4251326
	$\varepsilon=5000, th=6000, minEps=10$	0.0316225	0.3530357
	$\varepsilon=5000, th=6000, minEps=10$	0.0369205	0.4000756
	$\varepsilon=1000, th=2000, minEps=1$	0.0336921	0.3530357
	$\varepsilon=4000, th=2000, minEps=1$	0.0277318	0.3525041
Convoy	$min\ points=2, lifetime=10, dist\ max=10^{-2}$	0.0362583	0.3566276
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	0.0332781	1.0634231
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0243377	0.5973882
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	0.0352649	0.4018430
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0360927	0.3925327
	$min\ points=2, lifetime=2, dist\ max=10^{-1}$	0.0368377	0.3525041
	$min\ points=3, lifetime=5, dist\ max=10^{-1}$	0.0276490	0.7646296
Flock	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0369205	0.3704636
	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
Flock partials	$\varepsilon=10, \mu 2, \delta=2$	0.0368377	0.3525041
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0355132	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=2$	0.0368377	0.3670011

3. PRUEBAS Y RESULTADOS

3. Efecto en la recomendación

Algoritmo	Parámetros	Precisión	MAE
Ad-Hoc	$\delta=3600$	0.0361754	0.3487827
	$\delta=10800$	0.0349337	0.3637682
Sim. tray.	Hausdorff	0.0175497	0.5784518
	DTW	0.0206126	1.2903128
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	0.0369205	0.4251326
	$\varepsilon=5000, th=6000, minEps=10$	0.0316225	0.3530357
	$\varepsilon=5000, th=6000, minEps=10$	0.0369205	0.4000756
	$\varepsilon=1000, th=2000, minEps=1$	0.0336921	0.3530357
	$\varepsilon=4000, th=2000, minEps=1$	0.0277318	0.3525041
Convoy	$min\ points=2, lifetime=10, dist\ max=10^{-2}$	0.0362583	0.3566276
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	0.0332781	1.0634231
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0243377	0.5973882
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	0.0352649	0.4018430
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0360927	0.3925327
	$min\ points=2, lifetime=2, dist\ max=10^{-1}$	0.0368377	0.3525041
	$min\ points=3, lifetime=5, dist\ max=10^{-1}$	0.0276490	0.7646296
Flock	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0369205	0.3704636
	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
Flock partials	$\varepsilon=10, \mu 2, \delta=2$	0.0368377	0.3525041
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0355132	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=2$	0.0368377	0.3670011

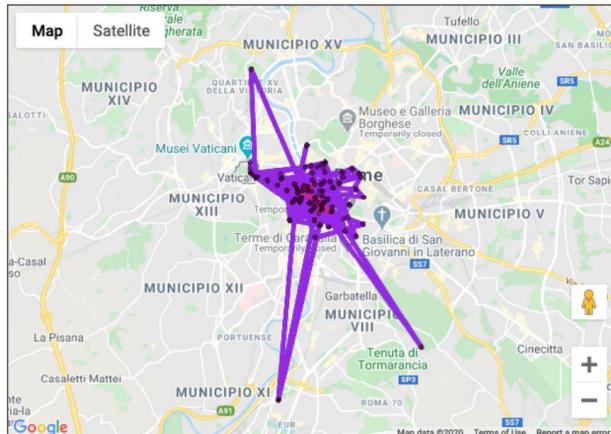
3. PRUEBAS Y RESULTADOS

3. Efecto en la recomendación

Algoritmo	Parámetros	Precisión	MAE
Ad-Hoc	$\delta=3600$	0.0361754	0.3487827
	$\delta=10800$	0.0349337	0.3637682
Sim. tray.	Hausdorff	0.0175497	0.5784518
	DTW	0.0206126	1.2903128
ST-DBSCAN	$\varepsilon=5000, th=6000, minEps=1$	0.0369205	0.4251326
	$\varepsilon=5000, th=6000, minEps=10$	0.0316225	0.3530357
	$\varepsilon=5000, th=6000, minEps=10$	0.0369205	0.4000756
	$\varepsilon=1000, th=2000, minEps=1$	0.0336921	0.3530357
	$\varepsilon=4000, th=2000, minEps=1$	0.0277318	0.3525041
Convoy	$min\ points=2, lifetime=10, dist\ max=10^{-2}$	0.0362583	0.3566276
	$min\ points=20, lifetime=2, dist\ max=10^{-2}$	0.0332781	1.0634231
	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0243377	0.5973882
	$min\ points=2, lifetime=5, dist\ max=10^{-2}$	0.0352649	0.4018430
Convoy partials	$min\ points=2, lifetime=2, dist\ max=10^{-3}$	0.0360927	0.3925327
	$min\ points=2, lifetime=2, dist\ max=10^{-1}$	0.0368377	0.3525041
	$min\ points=3, lifetime=5, dist\ max=10^{-1}$	0.0276490	0.7646296
Flock	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0369205	0.3704636
	$\varepsilon=10, \mu 2, \delta=2$	0.0369205	0.3530357
Flock partials	$\varepsilon=10, \mu 2, \delta=2$	0.0368377	0.3525041
	$\varepsilon=0.1, \mu 2, \delta=10$	0.0355132	0.3530357
	$\varepsilon=0.1, \mu 2, \delta=2$	0.0368377	0.3670011

3. PRUEBAS Y RESULTADOS

4. Visualización y validación de trayectorias



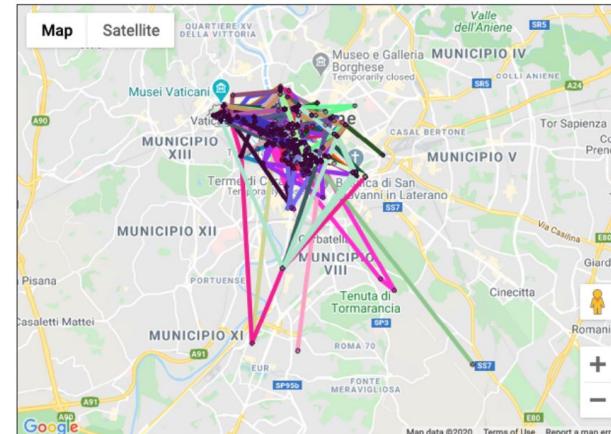
Users

KN

Neighbors

206	Neighbor ID	Similarity
2066	926	1
	257	1

(c) Trayectoria Ad-Hoc $\delta=3600$.



Users

KN

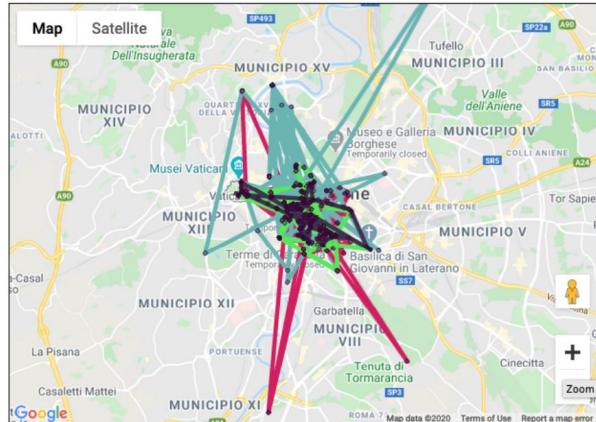
Neighbors

2066	Neighbor ID	Similarity
2066	1795	0.9808990905
	2223	0.9808990905

(d) Trayectoria DTW.

3. PRUEBAS Y RESULTADOS

4. Visualización y validación de trayectorias - Convoy



Users

KN

Neighbors

296

Neighbor ID

Similarity

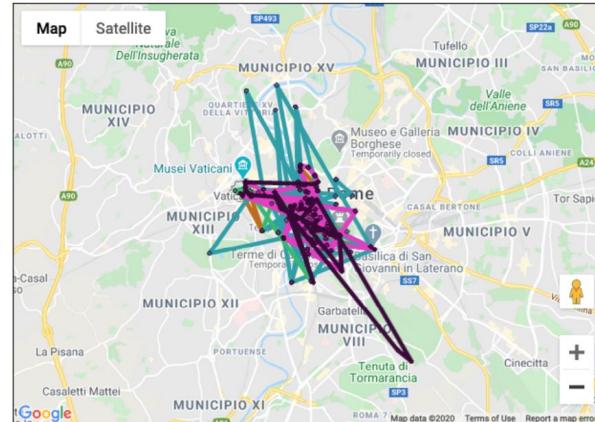
296

1589

1

539

1



Users

KN

Neighbors

296

Neighbor ID

Similarity

296

1685

1

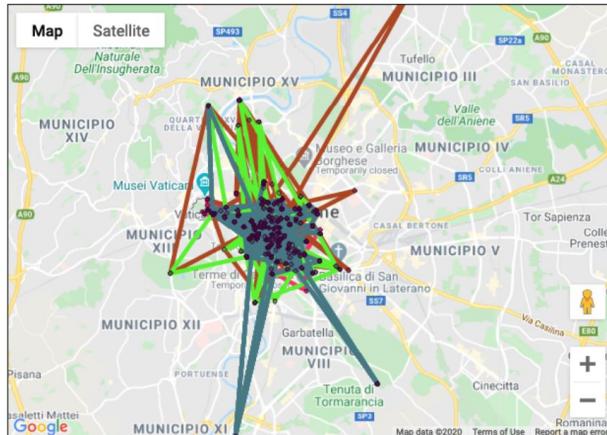
1310

(a) Trayectoria Convoy $min\ points=2$, $lifetime=2$, $dist\ max=0.01$.

(b) Trayectoria Convoy $min\ points=5$, $lifetime=2$, $dist\ max=0.01$.

3. PRUEBAS Y RESULTADOS

4. Visualización y validación de trayectorias - Flock



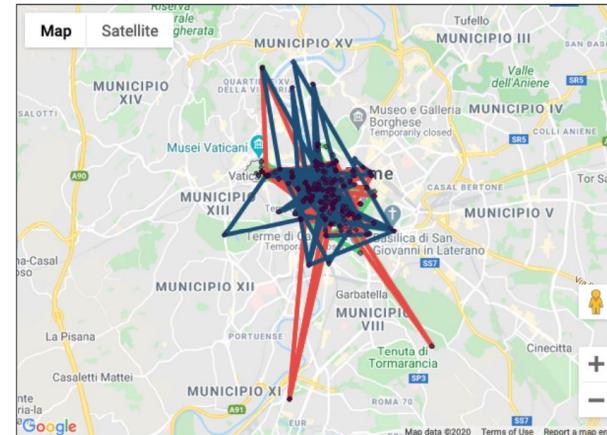
Users

KN

Neighbors

2066	Neighbor ID	Similarity
2066	0	1
	789	1

(a) Trayectoria Flock $\varepsilon=0.1$, $\mu=2$, $\delta=2$.



Users

KN

Neighbors

2066	Neighbor ID	Similarity
2066	789	1
	296	1

(b) Trayectoria Flock $\varepsilon=0.1$, $\mu=2$, $\delta=10$.



4. CONCLUSIONES Y TRABAJO FUTURO

4. CONCLUSIONES Y TRABAJO FUTURO

Conclusiones



1. Ventajas e inconvenientes

2. Utilidad estudios de trayectorias

3. Restricciones algoritmos

4. Publicación artículos

4. CONCLUSIONES Y TRABAJO FUTURO

Conclusiones - Publicaciones

Discovering Related Users in Location-Based Social Networks

Sergio Torrijos
Universidad Autónoma de Madrid
Madrid, Spain
sergio.torrijos@estudiante.uam.es

Alejandro Bellogín
Universidad Autónoma de Madrid
Madrid, Spain
alejandro.bellogin@uam.es

Pablo Sánchez
Universidad Autónoma de Madrid
Madrid, Spain
pablo.sanchezp@uam.es

ABSTRACT

Users from Location-Based Social Networks can be characterized by how and where they move. However, most of the works that exploit this type of information neglect either its sequential nature or its geographical properties. In this article, we focus on a specific family of recommender systems, those based on nearest neighbours; we define related users based on common check-ins and similar trajectories and analyse their effects on the recommendations. For this purpose, we use a real-world dataset and compare the performance on different dimensions against several state-of-the-art algorithms. The results show that better neighbours could be discovered with these approaches if we want to promote novel and diverse recommendations.

they visit, establish connections with other users, and check venue properties, such as their opening times, opinions, and pictures.

Because of the increasing number of users registered in LBSNs and similar systems, POI recommendation approaches have become particularly useful and several specific models have been proposed in recent years. In particular, such approaches tend to incorporate inherent properties of these systems, such as social, geographical, or temporal information [20, 21]. However, nearest neighbour techniques have been, in general, neglected in most of these studies, in favour of matrix factorisation or neural networks models [21, 24]. Nonetheless, we believe that algorithms based on similarities have a huge potential, since they may provide efficient computation, easy implementation, and explainable recommendations [23], but also because it has been demonstrated recently that these techniques

S. Torrijos, A. Bellogín, and P. Sánchez, “Discovering related users in location-based social networks,” in *User Modeling, Adaptation, and Personalization - 28th International Conference, UMAP 2020, Genoa, Italy, July 12-18, 2020. Proceedings, Lecture Notes in Computer Science, Springer, 2020.*

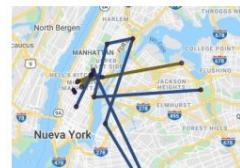
Analysis of co-movement pattern mining methods for recommendation

Extended Abstract

Sergio Torrijos
Universidad Autónoma de Madrid
Madrid, Spain
sergio.torrijos@estudiante.uam.es

ABSTRACT

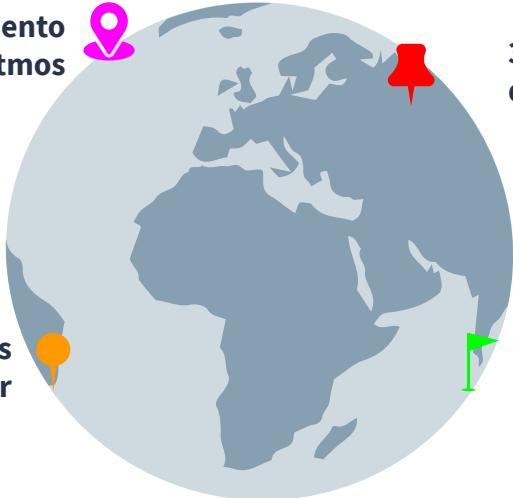
Location-Based Social Networks allow users to share the Points-of-Interest they visit, hence creating trajectories throughout their usual lives – even though they are also used by tourists to explore a city. There exist several algorithms in the trajectory pattern mining area able to discover and exploit interesting patterns from trajectory data, such as which objects tend to move together (co-movement), however, to the best of our knowledge, they have not been used with data coming from that type of systems. In this work, we analyse the extent to which these techniques can be applied to that type of data and under which circumstances they might be useful.



S. Torrijos and A. Bellogín, “Analysis of trajectory pattern mining methods for recommendation: Extended abstract,” in *Proceedings of the Joint Conference of the Information Retrieval Communities in Europe, CIRCLE 2020, Samatan, France, July 6-9, 2020, 2020.*

4. CONCLUSIONES Y TRABAJO FUTURO

Trabajo futuro

- 
1. Mejorar Rendimiento Algoritmos 📍
 2. Validación de las similitudes mediante Twitter 📍
 3. Solventar el problema de la dispersión en recomendación 📍
 4. Analizar resultados en función del tipo de usuario (local o turista) 📍

Estudio de métodos de detección de patrones de movimiento para sistemas de recomendación turísticos.

Sergio Torrijos López de la Manzanara

Muchas gracias por su atención

2. DISEÑO E IMPLEMENTACIÓN

Implementación - Dynamic Time Warping y Hausdorff

Fórmula similitud:

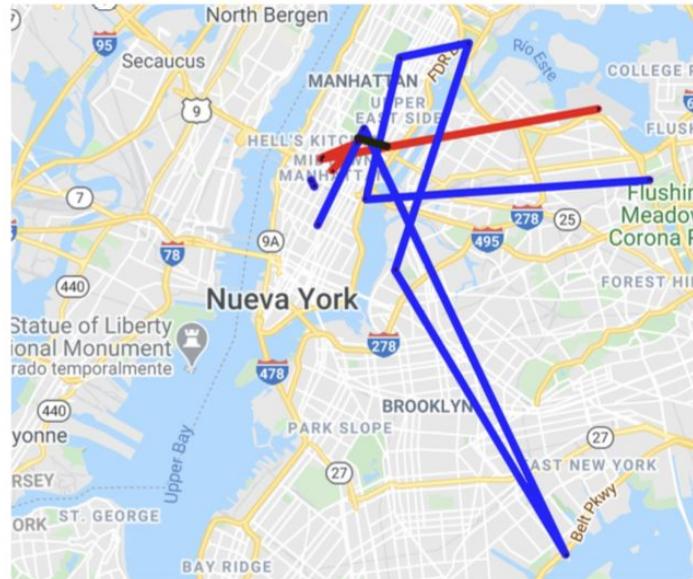
$$sim(u, v) = \frac{1}{n \cdot m} \sum_{j=1}^n \sum_{k=1}^m tsim(x_j^u, x_k^v)$$

Pseudocódigo:

```
1  D = [elementos_traj, elementos_traj]
2  for user in trayectorias_usuarios
3      calcular similitud DTW/Hausdorff
4  Almacenar los k mas similares
```

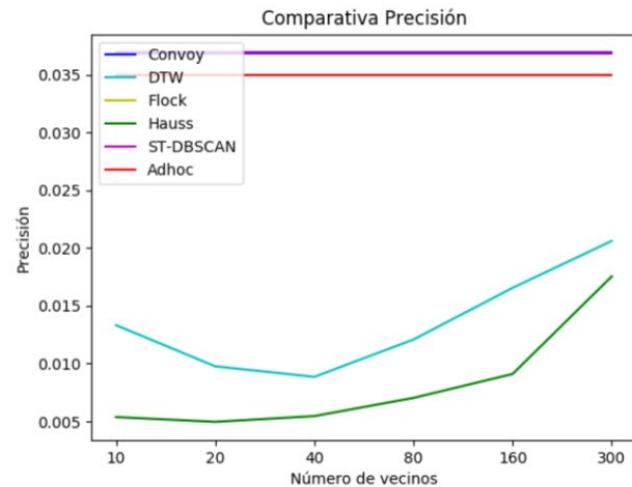
2. DISEÑO E IMPLEMENTACIÓN

Implementación - Dynamic Time Warping y Hausdorff

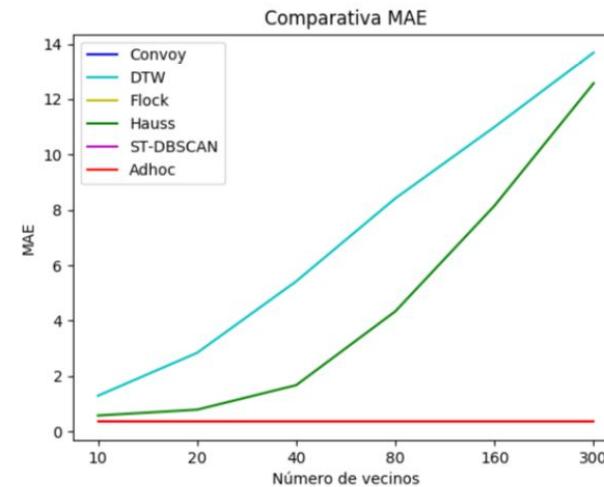


3. PRUEBAS Y RESULTADOS

Validación del Recomendador



(a) Evolución precisión recomendador.



(b) Evolución MAE recomendador.

3. PRUEBAS Y RESULTADOS

Comparativa entre recomendadores con similitud de trayectorias vs estándar

Recomendador	KNN Ad-Hoc	KNN Convoy	KNN DTW	KNN Flock	KNN Hausdorff	KNN ST-DBSCAN	SVD	Random	KNN Baseline	NMF
MAE	0.363768	0.352504	1.290313	0.352504	0.578452	0.353036	0.333831	0.475569	0.343762	0.272056



2. DISEÑO E IMPLEMENTACIÓN

Implementación - Recomendación

Formato fichero salida:

	user_id	item_id	rating
1	8	7	1.0964
2	8	64	1.2289
3	110	7	1.2235
4			

3. PRUEBAS Y RESULTADOS

Obtención de Trayectorias - Cluster de K-Cores

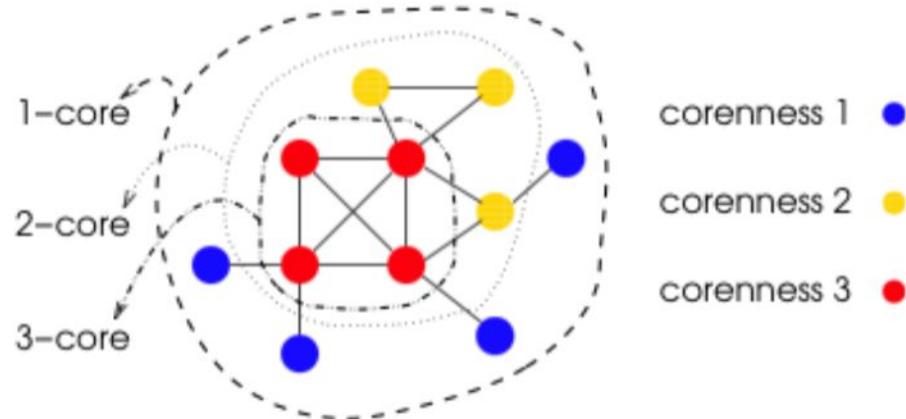


Figura 2.7: Procesado de K-Cores sobre un grafo [22].

3. PRUEBAS Y RESULTADOS

Relación entre tamaño del dataset y número de POIs/usuarios

Dataset	1K	10K	20K	60K
Número POIs	163	319	346	394
Número usuarios	129	1208	2618	7954

2. DISEÑO E IMPLEMENTACIÓN

Implementación - Ad-Hoc

Fórmula similitud:

$$sim_{\delta}(u, v) = \|i \in I : |t(u, i) - t(v, i)| < \delta\|$$

2. DISEÑO E IMPLEMENTACIÓN

Implementación - Visualización

Ficheros necesarios:

- El fichero del dataset procesado con las columnas (user_id, item_id, lat, long, timestamp).
- El fichero de similitudes con las columnas (user1_id, user2_id, similitud).
- Opcional: El fichero de trayectorias con el formato ('user_id': ['traj_counter': [...]]).

3. PRUEBAS Y RESULTADOS

Entorno de pruebas

Recursos	Características
Versión Python	2.7
S.O.	macOS Mojave 10.14.6 (18G103)
CPU	2,7 GHz Intel Core i7 (I7-8559U)
GPU	Intel Iris Plus Graphics 655 1536 MB graphics
RAM	16 GB 2133 MHz LPDDR3

2. DISEÑO E IMPLEMENTACIÓN

Implementación - Salida común ejecución patrones de movimiento

```
1          user1_id user2_id occurrences
2              21        21        1.0
3              21        39        2.0
```

2. DISEÑO E IMPLEMENTACIÓN

Patrones de movimiento - Flock

Parámetros:

ϵ

Distancia entre los elementos móviles (radio).

μ

Número mínimo de objetos móviles

δ

Intervalo de tiempo definido mínimo entre elementos.

2. DISEÑO E IMPLEMENTACIÓN

Implementación - Ejecución Flock

Ejecución en 2 fases:

1. Determinar los objetos móviles que se encuentran cerca según ε .
2. Combinación y agrupación de patrones que ocurran durante el parámetro δ .

```
1 KeyFlock: 30 Begin: 238 End 239 [0, 1025, 2437, 3720]
2 KeyFlock: 32 Begin: 231 End 239 [0, 1025, 2437]
3 KeyFlock: 34 Begin: 238 End 240 [0, 1025, 2437, 3720]
```

	<code>id</code>	<code>item_id</code>	<code>latitude</code>	<code>longitude</code>	<code>timestamp</code>
1	796	140514	40.758328	-73.985457	0
2	1024	788744	40.730084	-73.989256	0
3	1024	788734	40.730085	-73.989257	1
4	1024	788784	40.730086	-73.989258	2

2. DISEÑO E IMPLEMENTACIÓN

Implementación - Preparación del dataset

1. Selección atributos

1	user_id	item_id	latitude	longitude	timestamp
2	15	14627	40.757564	-73.989238	1354881399

1. 1. Selección atributos Hausdorff / DTW

```
1  {"user_id": [{"0": [[lat_elements], [long_elements]], "1": ...}]}  
2  
3  {"15": [{"0": [40.77383804321289, 40.757564544677734, 40.75837326049805,  
4  40.76288986206055], [-73.87122344970703, -73.9892349243164, -73.98849487304688,  
5  -73.97402954101562]], "1": [[40.753875732421875], [-73.98442840576172]],  
6  "2": [[40.76355743408203], [-73.97286987304688]]}}}
```

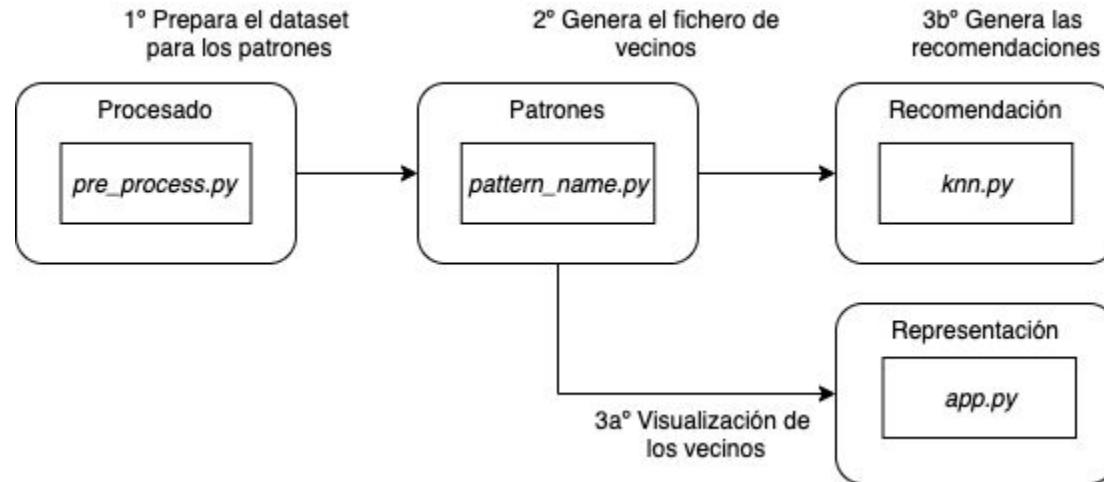
2. DISEÑO E IMPLEMENTACIÓN

Implementación - Flujo de la aplicación

1. python3 src/Processing/pre_process.py --input_file entradas/Rome10K/romeTempTrain.txt --coords_file entradas/POIS_rome_Coords.txt --output_file romePOISCompleto.txt
2. python3 src/Patterns/Convoy/ConvoyTrajectory.py --filename romePOISCompleto.txt --output similarity_output_convoy.txt --minpoints 3 --lifetime 2 --distance_max 0.1 --partials False
3. python3 src/Recommender/knn.py --train_file entradas/Rome10K/romeTempTrain.txt --test_file entradas/Rome10K/romeTest.txt --k 1 --neighbors_classified similarity_output_convoy.txt --output_file salida_knn_custom_rome.txt
4. python3 app.py

2. DISEÑO E IMPLEMENTACIÓN

Implementación - Flujo de la aplicación





2. ESTADO DEL ARTE

2. ESTADO DEL ARTE

Patrones de movimiento - ST-DBSCAN

Parámetros:

ϵ	minEps	temporal_threshold (th)
Radio espacial que delimita los puntos	Número mínimo de puntos para definir el cluster	Ventana temporal que delimita los instantes de tiempo

2. ESTADO DEL ARTE

Patrones de movimiento - ST-DBSCAN

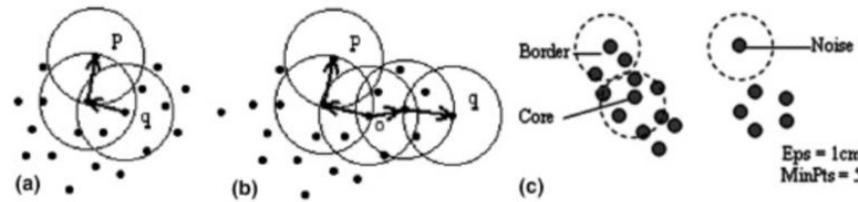


Figura 2.3: Ejemplo de cluster de puntos ST-DBSCAN [14], donde se aprecian los círculos delimitando el radio a buscar y cada uno de los puntos para definir cuáles entran en el criterio.

2. ESTADO DEL ARTE

Patrones de movimiento - Librerías utilizadas

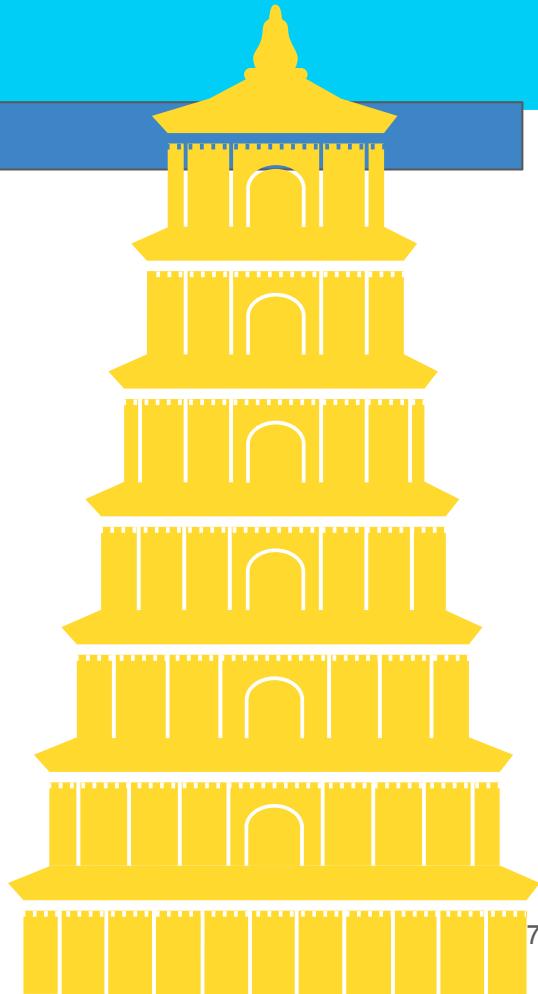
LCM

FPFlock

Pygmaps

py-st-dbscan

**Coherent Moving
Cluster Algorithm**



2. ESTADO DEL ARTE

Obtención de Trayectorias - Partición de trayectorias por instantes de tiempo

Timestamp:

Convert epoch to human-readable date and vice versa

1591653145

Timestamp to Human date

[batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **seconds**:

GMT: Monday, 8 June 2020 21:52:25

Your time zone: lunes, 8 de junio de 2020 23:52:25 **GMT+02:00 DST**

Relative: A few seconds ago

2. ESTADO DEL ARTE

Sistemas de recomendación

Filtrado Colaborativo:

Métodos basados en modelos

1. Factorización de matrices

Métodos basados en memoria

1. Rating y Similitud basado en usuario
2. Rating y similitud basado en ítem



2. ESTADO DEL ARTE

Sistemas de recomendación - Métodos basados en modelos

Factorización de matrices

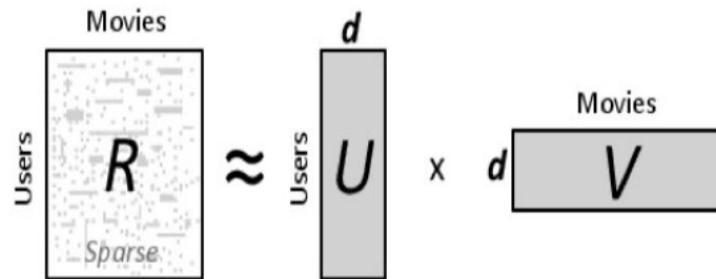


Figura 2.8: Representación de la factorización de matrices [24].

$$\hat{r}(u, i) = x_u^T \cdot y_i$$

2. ESTADO DEL ARTE

Sistemas de recomendación - Métodos basados en memoria

Predicción de rating y similitud entre usuarios

$$\hat{r}(u, i) = C \times \sum_{v \in N_k(u), r(v, i) \neq \emptyset} sim(u, v) r(v, i)$$

Predicción de rating y similitud entre ítems

$$\hat{r}(u, i) = C \times \sum_{r(u, j) \neq \emptyset} sim(i, j) r(u, j)$$

$$C = \frac{1}{\sum_{v \in N_k(u), r(v, i) \neq \emptyset} |sim(u, v)|}$$

$$C = \frac{1}{\sum_{r(u, j) \neq \emptyset} |sim(i, j)|}$$

2. ESTADO DEL ARTE

Sistemas de recomendación - Funciones de similitud

Similitud Coseno

$$sim(u, v) = \frac{\sum_{v \in i: r(u, i) \neq \emptyset, r(v, i) \neq \emptyset} r(u, i)r(v, i)}{\sqrt{\sum_{i: r(u, i) \neq \emptyset} r(u, i)^2 \sum_{i: r(v, i) \neq \emptyset} r(v, i)^2}} \epsilon [0, 1]$$

Correlación de Pearson

$$sim(u, v) = \frac{\sum_{v \in i: r(u, i) \neq \emptyset, r(v, i) \neq \emptyset} (r(u, i) - \bar{r}_u)(r(v, i) - \bar{r}_v)}{\sqrt{\sum_{i: r(u, i) \neq \emptyset, r(v, i) \neq \emptyset} (r(u, i) - \bar{r}_u)^2 \sum_{i: r(u, i) \neq \emptyset, r(v, i) \neq \emptyset} (r(v, i) - \bar{r}_v)^2}} \epsilon [0, 1]$$

2. ESTADO DEL ARTE

Sistemas de recomendación - Métricas de evaluación

MAE

$$MAE = \frac{1}{|test|} \sum_{(u,i) \in test} |\hat{r}(u, i) - r(u, i)|$$

RMSE

$$RMSE = \sqrt{\frac{1}{|test|} \sum_{(u,i) \in test} (\hat{r}(u, i) - r(u, i))^2}$$

Precisión

$$P = \frac{TP}{TP + FP}$$

Recall

$$R = \frac{TP}{TP + FN}$$

2. DISEÑO E IMPLEMENTACIÓN

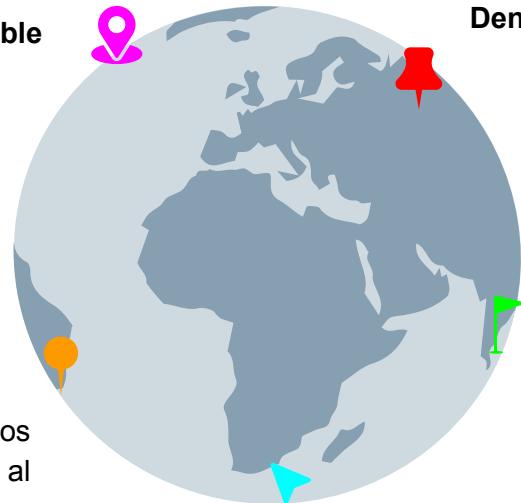
Patrones de movimiento - Convoy

Density-reachable

Un punto p es alcanzable desde un punto q si el punto p está a una distancia ϵ del punto q , y q tiene un número suficiente de puntos en sus vecinos a una distancia ϵ .

Core object

Un punto en el cual sus vecinos conectados deben satisfacer la condición de contener al menos minPts .



Búsqueda Convoy

Convoy devuelve todos los grupos de objetos posibles tal que cada grupo consista en un grupo máximo de puntos densamente conectados con respecto a ϵ y m durante al menos k puntos.

Density-Connected

Un punto $p \in S$ está densamente conectado a un punto $q \in S$ con respecto a ϵ y m si existe un punto $x \in S$ tal que ambos p y q son alcanzables desde x .

Border-object

Un objeto p es un border-object si no es un core-object pero es density-reachable desde otro core-object.