

Universidad Autónoma de Madrid  
Escuela Politécnica Superior



Proyecto para la obtención del título de  
Máster en Data Science  
por la Universidad Autónoma de Madrid



Tutor del trabajo fin de máster:  
Alejandro Bellogín Kouki



## **Estudio de redes neuronales basadas en grafos aplicado a recomendación del siguiente punto de interés**

Miguel Carreras Lahoz

**Todos los derechos reservados.**

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

**DERECHOS RESERVADOS**

© 29 de febrero de 2024 por UNIVERSIDAD AUTÓNOMA DE MADRID

Francisco Tomás y Valiente, nº 1

Madrid, 28049

Spain

**Miguel Carreras Lahoz**

**Estudio de redes neuronales basadas en grafos aplicado a recomendación del siguiente punto de interés**

**Miguel Carreras Lahoz**

C\ Francisco Tomás y Valiente Nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

# AGRADECIMIENTOS

---

En primer lugar, quiero agradecer a mi tutor, Alejandro Bellogín, por su constante apoyo y orientación en el desarrollo de este trabajo. Su conocimiento y dedicación han sido fundamentales para el éxito de este trabajo.

A mi familia y amigos, por su cariño y confianza plena en mí durante este año y medio. En especial, a mis compañeros de piso Mario y Joselu, por acompañarme en todo momento.



# RESUMEN

---

Debido al crecimiento significativo de la información en la era del big data, cada vez es más complicado que los usuarios encuentren contenidos o servicios de su interés, generando un problema grave de "sobrecarga de información". Ante este desafío, los sistemas de recomendación han surgido para aprovechar los datos de comportamiento histórico del usuario y ofrecer servicios personalizados, abordando diversos campos, como aplicaciones de Puntos de Interés (POI), servicios multimedia o sitios web de comercio electrónico.

En el caso de los recomendadores de Puntos de Interés (POI), estos explotan esencialmente el historial de visitas o check-ins de los usuarios y otra información multimodal, como los atributos de los locales o la red de amistades, para recomendar el siguiente conjunto de POI adecuado para un usuario. Sus modelos, que inicialmente estaban basados en técnicas tradicionales, como la factorización de matrices o las cadenas de Markov, pronto estuvieron limitados por el reto de explotar plenamente las complejas correlaciones y transiciones entre POIs en las secuencias de check-ins. Sin embargo, la introducción de modelos de aprendizaje profundo ha marcado un hito en la mejora de la precisión de las recomendaciones de POIs.

Estos modelos, como las Redes Neuronales basadas en Grafos (GNN) entre otros, han demostrado su capacidad para abordar las limitaciones previas al capturar hábilmente las enrevesadas relaciones en las interacciones entre usuarios y puntos de interés. Concretamente estas GNN han conseguido destacar al explotar al mismo tiempo las preferencias a corto y largo plazo de las visitas, así como aprovechar la riqueza de datos multimodales, incluyendo aspectos espaciales, sociales y textuales, desde una perspectiva más eficiente que otros tipos de redes.

En particular, este trabajo presenta una revisión del estado del arte de los modelos de recomendación de POIs que aplican técnicas de aprendizaje profundo, centrándonos en los modelos GNN, algo bastante novedoso puesto que no se ha encontrado un revisión anterior. Además, se plantean experimentos para mejorar un modelo GNN existente en la recomendación del siguiente punto de interés, con el fin de completar parte del estudio de estos modelos, al igual que abrir la puerta a futuras investigaciones en este área.

# PALABRAS CLAVE

---

Sistemas de recomendación, Recomendación secuencial, Recomendación del siguiente punto de interés, Red Neuronal con Grafo, Mecanismo de Atención



# ABSTRACT

---

Due to the significant growth of information in the big data era, it is becoming increasingly difficult for users to find content or services of interest to them, generating a serious problem of “information overload”. Faced with this challenge, recommender systems have emerged to take advantage of historical user behavior data and offer personalized services, addressing various fields, such as Point of Interest (POI) applications, multimedia services, or e-commerce websites.

In the case of POI recommenders, they essentially exploit users’ visit history or check-ins and other multimodal information, such as venue attributes or friendship network, to recommend the next set of POIs suitable for a user. Their models, which were initially based on traditional techniques, such as matrix factorization or Markov chains, were soon limited by the challenge of fully exploiting the complex correlations and transitions between POIs in sequences of check-ins. However, the introduction of deep learning models has marked a milestone in improving the accuracy of POI recommendations.

These models, such as Graph Neural Networks (GNNs) among others, have demonstrated their ability to address previous limitations by deftly capturing the convoluted relationships in interactions between users and points of interest. Specifically, these GNNs have managed to stand out by simultaneously exploiting both short- and long-term preferences of visits, as well as taking advantage of the richness of multimodal data, including spatial, social, and textual aspects, from a more efficient perspective than other types of networks.

In particular, this work presents a state-of-the-art review of POI recommendation models applying deep learning techniques, focusing on GNN models, which is quite novel since no previous review has been found on this topic. In addition, experiments are proposed to improve an existing GNN model focused on the recommendation of the next POI, in order to complete part of the study of these models, as well as to open the door to future research in this area.

# KEYWORDS

---

Recommender Systems, Sequential Recommendation, Next Point of Interest Recommendation, Graph Neural Network, Attention Mechanism



# ÍNDICE

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Objetivos y enfoque	3
1.2	Estructura del documento	4
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	Sistemas de Recomendación	5
2.1.1	Notación y conceptos básicos	5
2.1.2	Estrategias de recomendación	6
2.2	Sistemas de Recomendación del siguiente punto de interés	11
2.2.1	Definición del problema	11
2.2.2	Problemas y desafíos	12
2.2.3	Procesamiento de datos	14
2.2.4	Modelos de recomendación NEXT-POI	14
2.3	Redes neuronales basadas en grafos	17
2.3.1	Redes Neuronales artificiales	17
2.3.2	Redes Neuronales de Grafos	17
2.4	Redes neuronales de grafos en recomendación de POI	19
2.4.1	Selección de artículos	19
2.4.2	Discusión	22
2.5	Evaluación de los Sistemas de Recomendación	22
<b>3</b>	<b>Diseño e implementación</b>	<b>25</b>
3.1	Entorno de trabajo	25
3.2	Descripción del modelo	26
3.2.1	Construcción del grafo de la secuencia de check-ins	26
3.2.2	Aprendizaje de la representación de características	27
3.2.3	Captura de las preferencias a corto y largo plazo	28
3.2.4	Recomendación de POI	29
3.3	Implementación	30
3.3.1	Variaciones	30
3.3.2	Estudio de hiperparámetros	31
3.3.3	Justificación de las épocas de entrenamiento	33
<b>4</b>	<b>Experimentos</b>	<b>35</b>

4.1	Conjunto de datos .....	35
4.2	Variaciones .....	36
4.3	Estudio de hiperparámetros .....	38
4.4	Resultados .....	41
<b>5</b>	<b>Conclusión y trabajo futuro</b>	<b>43</b>
5.1	Conclusión .....	43
5.2	Trabajo futuro .....	44
	<b>Bibliografía</b>	<b>49</b>

# LISTAS

---

## Lista de algoritmos

## Lista de ecuaciones

2.1	Similitud coseno .....	7
2.2	Error absoluto medio .....	22
2.3	Error cuadrático medio .....	22
2.4	Precisión .....	23
2.5	Recall .....	23
2.6	Rango recíproco medio .....	23
3.4	Preferencias personalizadas del usuario $u$ .....	29
3.5	Predicción entre un usuario $u$ y el POI $l_i$ .....	29
3.6	Predicción entre un usuario $u$ y el POI $l_i$ .....	29

## Lista de figuras

1.1	Ejemplo de check-ins de POI asociados a dos usuarios .....	2
2.1	Matriz de valoraciones .....	6
2.2	Comparación de las principales estrategias de recomendación .....	10
2.3	Representación gráfica de la información de una LBSN .....	12
2.4	Red de grafos .....	18
3.1	Arquitectura general del modelo ASGNN .....	26
3.2	Esquema de la propagación de información en una GNN .....	27
3.3	Rango Recíproco Medio para 30 épocas .....	34
3.4	Recall para 30 épocas .....	34

## Lista de tablas

2.1	Estado del arte del Deep Learning en recomendación de POI .....	20
-----	---	----

3.1	Especificaciones del entorno de trabajo .....	25
3.2	Resumen de las variaciones a realizar en el modelo base .....	30
3.3	Parámetros iniciales del modelo .....	32
3.4	Resumen del estudio de hiperparámetros.....	33
4.1	Comparación del rendimiento de las variantes .....	36
4.2	Resumen de las nuevas versiones propuestas.....	38
4.3	Comparación del rendimiento de las tres versiones propuestas .....	38
4.4	Comparación del rendimiento del estudio de hiperparámetros .....	39
4.5	Comparación del rendimiento de la combinación de hiperparámetros .....	40
4.6	Comparación del rendimiento de la unión de ambas pruebas .....	41

# INTRODUCCIÓN

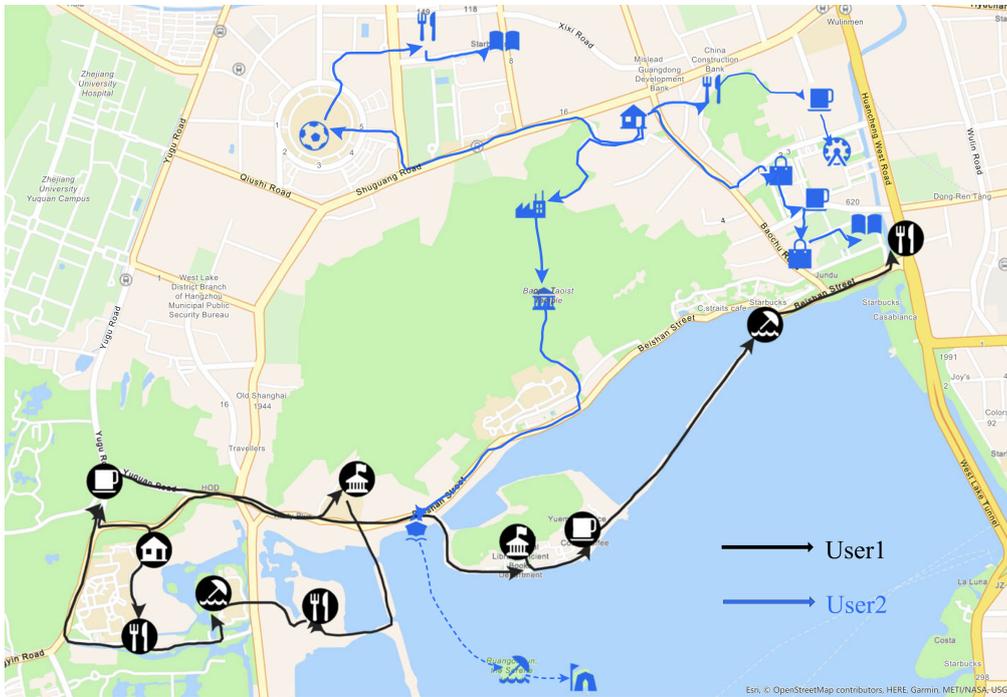
---

En los últimos años, hemos sido testigos de un notable y continuo crecimiento de la popularidad de las redes sociales con un enfoque en la localización (LBSN, de *Location-Based Social Network* en inglés). Estas se distinguen de las redes sociales convencionales por permitir a los usuarios compartir sus experiencias sobre los distintos puntos de interés que visitan, los cuales quedan guardados en vectores de información llamados check-ins. Plataformas como Foursquare, Yelp y Gowalla han liderado este cambio al permitir a los usuarios no solo comunicar su ubicación, sino también compartir sus experiencias, recomendaciones y descubrimientos en diversos puntos de interés, como restaurantes, hoteles y atracciones turísticas, obteniendo así grandes bases de datos de usuarios. El crecimiento vertiginoso de las redes sociales basadas en la localización ha suscitado un profundo interés en la investigación y desarrollo de sistemas de recomendación enfocados en los puntos de interés (POI).

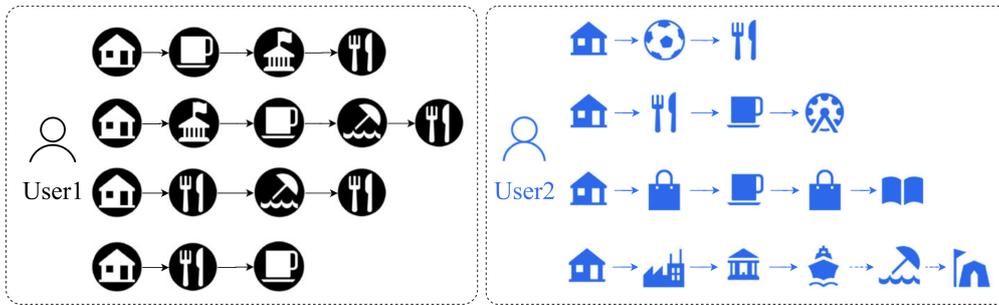
El objetivo principal de estos sistemas no es solo presentar a los usuarios nuevas y emocionantes opciones de lugares por descubrir, sino también comprender en profundidad sus preferencias individuales. Al hacerlo, buscan enriquecer significativamente la experiencia del usuario al proporcionar recomendaciones altamente personalizadas y oportunas. Dado que el tamaño y la tipología de los datos, así como las demandas de los usuarios, varían enormemente, esta oportunidad de disponer de grandes cantidades de datos multimodales plantea nuevos desafíos a los investigadores, incentivándolos a diseñar técnicas novedosas con el objetivo de capturar de manera más efectiva los patrones de movilidad y otros tipos de características como las espaciales, sociales o textuales.

Los primeros estudios en este ámbito se han centrado principalmente en técnicas convencionales de aprendizaje automático, como la factorización matricial [1–3], cadenas de Markov [4–7] o Ranking Bayesiano Personalizado [8–11], respectivamente. A pesar de su utilidad, la recomendación de POI plantea tres grandes retos [12, 13]:

- Escasez de datos: A diferencia de otras áreas de recomendación, como la música o las películas, donde los datos suelen estar más disponibles y abundantes, este área se enfrenta a importantes limitaciones en cuanto a la disponibilidad de datos para construir sistemas de recomendación precisos. Entre los diferentes motivos, está que los usuarios tienden a visitar solo un pequeño subconjunto de los numerosos lugares de interés disponibles y que visitar



(a) Trayectorias de los check-in



(b) Secuencias de los check-in.

**Figura 1.1:** Un ejemplo de la secuencia de check-ins de POI asociados a dos usuarios. Las preferencias del usuario 1 suelen ser recurrentes, mientras que las del usuario 2 son totalmente arbitrarias, seguramente condicionadas por otro tipo de influencias externas. Obtenido de [14].

diferentes puntos de interés es un proceso más costoso y menos frecuente que otros tipos de recomendaciones.

- **Preferencias a corto y largo plazo:** Los comportamientos de check-in de los usuarios dependen generalmente de preferencias a corto y largo plazo, que pueden desempeñar papeles diferentes en la recomendación de POI. Por un lado, las preferencias a corto plazo pueden estar influenciadas por el estado de ánimo del usuario, el clima actual, eventos especiales o actividades recientes. Por otro lado, las preferencias a largo plazo reflejan intereses más arraigados y patrones de comportamiento consistentes a lo largo del tiempo. Por ejemplo, como se muestra en la Figura 1.1, el usuario 1 tiende a visitar los mismos tipos de establecimientos continuamente (largo plazo). Por el contrario, el usuario 2 no sigue un ningún patrón y cambia continuamente de preferencias (corto plazo).
- **Influencias externas:** mientras en otros tipos de recomendación solo se tiene en cuenta la valoración de usuarios y objetos, la recomendación del punto de interés está fuertemente condicionada por otros factores: geográficos (distancia entre POIs visitados), sociales (relaciones entre los usuarios) y temporales (según el día de la semana o el momento del día), entre otros. Además, existe información implícita como las secuencias de check-ins que capturan de manera efectiva los comportamientos periódicos de los usuarios. Todas estas influencias alivian el problema de la escasez de datos, pero implican una mayor complejidad en los recomendadores de POI.

Sin embargo, la aparición de modelos de aprendizaje profundo en los últimos años ha permitido avances sustanciales en la precisión de las recomendaciones de POI [14–20]. En concreto, algunos de los modelos de la vanguardia utiliza redes neuronales basadas en grafos [21–23], los cuales captan hábilmente las complejas relaciones entre usuarios y POI, facilitando la incorporación fluida de nuevos factores al modelo. Este enfoque ha demostrado ser clave para mitigar los retos mencionados.

En este trabajo, introducimos una revisión del estado del arte de los modelos de recomendación de POI basados en redes neuronales, incluyendo buena parte de la literatura actual de redes neuronales basadas en grafos (GNN), ya que no hemos encontrado ningún otro trabajo donde se hayan analizado y categorizado en detalle. Además, proponemos una serie de experimentos con los que completar y mejorar un modelo del estado del arte de GNN para recomendación del siguiente punto de interés.

## 1.1. Objetivos y enfoque

Las principales contribuciones de este trabajo son:

- Una revisión detallada del estado del arte centrada en los métodos que aplican redes neuronales basadas en grafos para la recomendación.

- Nuevas variantes de un modelo del estado del arte actual, el cual representa los comportamientos de registro del usuario con grafos y aplica GNN para aprender los patrones de comportamiento del usuario y sus preferencias para la recomendación del siguiente POI.
- Una discusión de los retos y deficiencias de los trabajos existentes y una lista de recomendaciones futuras para la investigación de POI en esta área.

## 1.2. Estructura del documento

El resto de este documento se organiza como sigue. En el Capítulo 2 se expone los conceptos claves del estado del arte actual, incluyendo nuestra revisión sobre los trabajos relacionados existentes. El Capítulo 3 describe el modelo de recomendación del siguiente POI junto con las hipótesis de los nuevos experimentos propuestos. El Capítulo 4 incluye tanto los resultados como un análisis experimental desarrollado. Finalmente, concluimos nuestro trabajo en el Capítulo 5 con las reflexiones y próximos retos de investigación más importantes.

# ESTADO DEL ARTE

---

En primer lugar, es importante contextualizar este proyecto, el cual se enfoca en el ámbito de la recomendación del siguiente punto de interés (NEXT POI) utilizando un modelo de Red Neuronal basado en grafos (GNN).

Este capítulo presenta tanto el marco teórico como un estudio bibliográfico de nuestro enfoque en los sistemas de recomendación NEXT POI, con una atención especial a los algoritmos de aprendizaje profundo, los cuales desempeñan un papel fundamental en el estado del arte. Además, se ofrece una descripción detallada de las redes neuronales basadas en grafos, ya que constituyen uno de las principales áreas de estudio en el estado del arte más reciente y son una parte fundamental de nuestra investigación.

## 2.1. Sistemas de Recomendación

En primer lugar, se detallan algunos conceptos básicos de los sistemas de recomendación (Subsección 2.1.1) y seguidamente, se profundiza en las principales estrategias de recomendación (Subsección 2.1.2).

### 2.1.1. Notación y conceptos básicos

Un sistema de recomendación es un modelo que proporciona sugerencias personalizadas a los usuarios, generalmente en el contexto de productos, servicios, contenido o información. Estos sistemas se utilizan en una gran variedad de aplicaciones, como plataformas de streaming de video, tiendas en línea, redes sociales, motores de búsqueda y más. El objetivo principal de un sistema de recomendación es ayudar a los usuarios a descubrir elementos o items que puedan ser de su interés y mejorar su experiencia de usuario.

Algunos de los conceptos y notaciones clave en sistemas de recomendación incluyen:

- Usuarios ( $u_i \in \mathcal{U}$ ): Representa al conjunto de individuos que participan en el sistema de

recomendación.

- Elementos ( $i_j \in \mathcal{I}$ ): Engloba al conjunto de elementos que se pueden recomendar: una película, un producto, una canción, un artículo, etc.
- Interacciones ( $r(u, i) \in \mathcal{R}$ ): Correspondería a la información sobre cómo un usuario interactúa con un elemento. Esto puede incluir calificaciones, reseñas, compras, clics, visitas, etc.
- Matriz de interacciones ( $\mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}$ ): Se forma a raíz del total de todas las posibles interacciones entre los usuarios y los elementos. Cada celda puede contener información sobre la interacción entre un usuario y un elemento, como una calificación o un valor binario.

La Figura 2.1 muestra un ejemplo de matriz de valoraciones. Como se puede observar, a pesar de desconocer muchas de las valoraciones de los usuarios a los diferentes items, el objetivo de un sistema de recomendación es obtener una buena sugerencia de la futura valoración de los usuarios. Para ello, existen diferentes estrategias de recomendación que difieren según se dé mayor peso a las características de los objetos o a las interacciones pasadas de usuarios. Por último, para poder comprobar los resultados, se aplican múltiples métricas de evaluación. Entre ellas, algunas de las más utilizadas son *Accuracy* [16, 17] y *Recall* [14, 20, 24].

		Items $\mathcal{I}$											
													
Users $\mathcal{U}$				1	3		2		4			3	
		1	2	5		4		1		2	4		5
		4		?	3	5			5			2	
			2			5	4	4		5			4
			3	4		5			4	3	5		4
		3		2		1	5		3			5	
		3			2			3		5		1	

**Figura 2.1:** Matriz de valoraciones. Las filas corresponderían a los usuarios, las columnas a los items y las puntuaciones se encuentran dentro del rango [1,5]. Obtenido de las transparecias de la asignatura de información no estructurada.

### 2.1.2. Estrategias de recomendación

A día de hoy existen diferentes estrategias a la hora de realizar una recomendación personalizada, cada una con sus propios desafíos y ventajas, sin embargo, la gran mayoría de los algoritmos de

recomendación se engloban en dos grandes categorías: los filtrados basados en el contenido y los filtrados colaborativos.

### Basadas en contenido

Los sistemas de recomendación basados en el contenido (CB) examinan tanto los elementos como las características del usuario (contenido), empleándolos para generar perfiles tanto del usuario como del ítem, cuya utilidad se utilizan para sugerir al usuario objetivo elementos que guardan similitudes con aquellos que ha apreciado en experiencias previas. Este enfoque se centra en comprender las preferencias individuales del usuario, ofreciendo recomendaciones personalizadas que se alinean estrechamente con sus intereses y gustos anteriores. Para hacer recomendaciones, este tipo de sistema utiliza tres componentes principales [25]:

- **Analizador de contenido:** corresponde al input de los otros dos componentes. Su objetivo es preprocesar la información disponible de los ítems para sacar conceptos claves u otra información de interés mediante técnicas de extracción de características.
- **Aprendiz de perfiles:** este módulo se encarga de recopilar y concretizar la información representativa de cada ítem, con el fin de construir un perfil para cada usuario basándose en sus preferencias pasadas.
- **El filtrado:** esta última componente explota el perfil del usuario, comparándolo con los ítems del sistema, para generar nuevas recomendaciones. Para lograr este objetivo, se usan técnicas de similitud como por ejemplo, la ecuación de similitud coseno (Ecuación 2.1).

Muchos algoritmos basados en el contenido recurren a enfoques simples de recuperación de información (IR), siendo el modelo de espacio vectorial (VSM) [26] uno de los más utilizados. Este modelo representa documentos (en este caso, descripciones de ítems) y consultas (preferencias del usuario) como vectores en un espacio multidimensional. El perfil del usuario se crea a partir de la agregación (ponderada) de los ítems puntuados por dicho usuario y, utiliza medidas de similitud para determinar la similitud entre los vectores de documentos y consultas. Aunque es un enfoque más clásico, sigue siendo habitual en este tipo de sistemas.

Para construir los vectores, se suelen utilizar esquemas basados en la frecuencia, como la ponderación TF-IDF o BM25 [27], una combinación de la "importancia" de los términos y su especificidad dentro del texto. Una vez que hemos transformado todos los ítems en vectores, se puede aplicar una métrica de similitud para obtener un ranking de ítems similares con respecto a otros que el usuario ha consumido previamente. Un ejemplo frecuente es la similitud coseno:

$$\hat{r}(u, x) = \text{sim}(u, x) = \text{cos}(u, x) = \frac{\sum_{k \in K} u_k x_k}{|u| |x|} \quad (2.1)$$

donde  $x_k$  correspondería al vector de palabras claves y  $u$  el promedio de los vectores de los ítems

puntuados por  $u$ , ponderados por sus calificaciones.

Además, obtener nuevas relaciones latentes entre los documentos y los términos asociados es uno de los principales objetos de estudio, y su investigación ha propiciado muchas otras técnicas de recomendación [25]. Algunos de estos algoritmos tienen como objetivo clasificar cada ítem  $i$  en una clase  $c$ . Un enfoque relevante sería el de los algoritmos basados en vecindarios, en los cuales se evalúa la similitud entre ítems en función de características compartidas y recomienda aquellos con la mayor proximidad. Otro enfoque son los modelos probabilísticos, como los Modelos de Tema Latente (LDA) [28], que asignan probabilidades a la presencia de temas específicos en documentos, permitiendo una representación más abstracta del contenido (por ejemplo,  $P(d|c)$  será la probabilidad de clasificar un documento  $d$  en la clase  $c$  para el usuario objetivo). Estos métodos, al considerar la proximidad o la probabilidad, ofrecen formas alternativas y complementarias de modelar las características de los ítems, permitiendo sistemas de recomendación más flexibles y adaptativos. Existen además opciones más complejas como las representaciones de palabras preentrenadas, aplicando modelos de procesamiento de lenguaje natural (NLP) [29], como Word2Vec o BERT, que ofrecen una comprensión más profunda de las relaciones semánticas y contextuales en el texto de los ítems, o el análisis de sentimientos, el cual agrega una capa adicional al modelado de características de los ítems al evaluar la tonalidad emocional asociada al texto, más allá de la temática del ítem y consiguiendo así una mejor personalización de las recomendaciones.

La adopción del enfoque de recomendación basado en el contenido ofrece diversas ventajas en comparación con otros paradigmas [25]. En primer lugar, destaca la independencia del usuario, ya que utilizan exclusivamente las valoraciones proporcionadas por el usuario activo para construir su propio perfil. Otra ventaja es la transparencia del sistema, ya que las explicaciones sobre el funcionamiento del sistema de recomendación pueden ofrecerse de manera explícita al enumerar las características y permiten al usuario evaluar y decidir si confiar en una recomendación. Por último, otros recomendadores sufren del problema del primer evaluador, no obstante, los recomendadores basados en el contenido tienen la capacidad de sugerir artículos que aún no han sido valorados por ningún usuario.

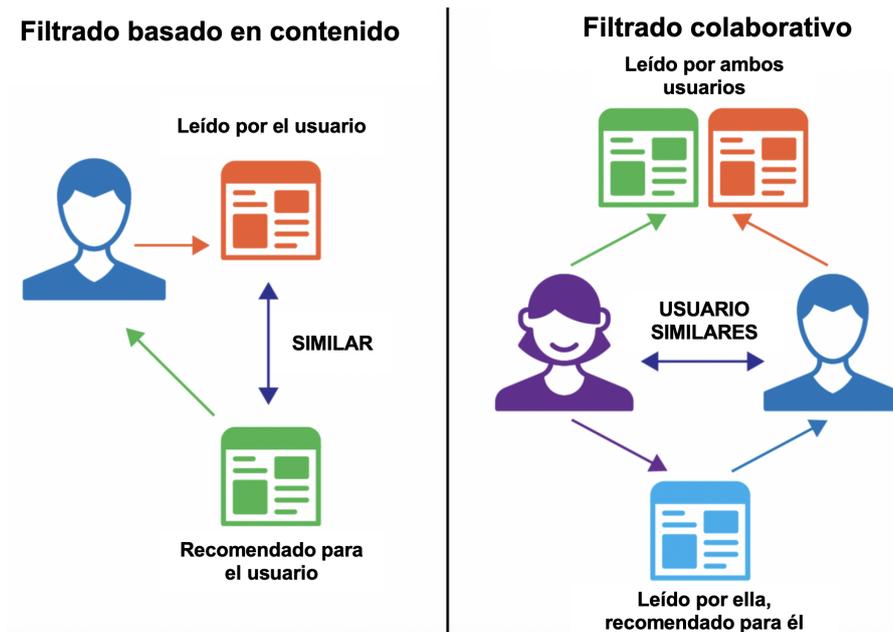
Sin embargo, estas técnicas enfrentan algunos desafíos [25], como el análisis de contenido limitado, donde las características asignadas a cada usuario o ítem puede ser insuficiente para comprender los intereses de los usuarios, la sobrespecialización, ya que los sistemas basados en el contenido tienden a sugerir elementos similares a los ya valorados, limitando la novedad de las recomendaciones y en el caso de usuarios nuevos, la falta de valoraciones suficientes dificulta la comprensión precisa de preferencias y la entrega de recomendaciones fiables.

## Filtrado colaborativo

El filtrado colaborativo (CF) es otra técnica popular en sistemas de recomendación que se centra en prever las preferencias de un usuario basándose en la información de usuarios similares. La pre-

misa fundamental es que si dos usuarios han compartido preferencias en el pasado, es probable que compartan intereses en el futuro. Este método aprovecha el comportamiento colectivo de usuarios para ofrecer recomendaciones personalizadas. Generalmente, los algoritmos de filtrado colaborativo se suelen englobar en dos familias: las basadas en memoria, las cuales se centran en comparar directamente la similitud entre usuarios o elementos utilizando métricas como la correlación de Pearson o la similitud del coseno (Ecuación 2.1), y los basados en modelos, los cuales en lugar de depender directamente de la similitud entre usuarios o elementos, utilizan modelos predictivos y algoritmos para hacer recomendaciones. A continuación explicamos algunos de los conceptos fundamentales relacionados con estas dos familias de algoritmos de CF [25]:

- **Basado en memoria:** la idea detrás de estos algoritmos es la similitud entre usuarios o elementos para realizar recomendaciones. En el enfoque usuario-usuario, se compara la similitud entre los gustos de un usuario y otros usuarios, mientras que en el enfoque ítem-ítem, se evalúa la similitud entre los elementos en función de cómo han sido calificados por diferentes usuarios. Estos métodos utilizan métricas como la correlación de Pearson o la similitud del coseno para cuantificar la relación entre las preferencias. Además, son llamados vecinos más cercanos o k-NN, debido a que construyen vecindarios con los usuarios/ítems con mayores similitudes. Una ventaja clave de este enfoque es su simplicidad y facilidad de implementación, siendo intuitivo para entender y utilizar, aunque, por otro lado, puede ser sensible a la falta de datos y tener dificultades con usuarios o elementos que tienen historiales de calificación limitados. A pesar de estos desafíos, el filtrado colaborativo basado en memoria sigue siendo una opción eficaz, especialmente cuando hay suficiente información histórica disponible.
- **Basado en modelos:** evita la comparación directa de similitud para utilizar algoritmos y modelos predictivos para hacer recomendaciones. Dentro de esta familia, destaca el área de la factorización de matrices (MF), la cual representa la información de usuarios y elementos en forma de matrices y la descompone en factores latentes. Dentro de este área, se incluyen muy diferentes algoritmos, como los que aplican LSA probabilística, que introduce un componente probabilístico para modelar la incertidumbre en las relaciones entre usuarios y elementos; Singular Value Decomposition (SVD), que logra una aproximación de la matriz original de ratings de usuarios-ítems mediante la descomposición de la misma; o incluso redes neuronales, cada uno con sus propias ventajas en términos de eficacia y manejo de datos. En general, aunque estos algoritmos pueden ser más complejos de implementar y requerir más recursos computacionales, el filtrado colaborativo basado en modelo destaca por su capacidad para abordar problemas de sobreajuste y ofrecer recomendaciones más robustas en entornos con datos ruidosos o escasos.



**Figura 2.2:** Comparación gráfica de las principales estrategias de recomendación: filtrado basado en el contenido y filtrado colaborativo.

## Otras estrategias

Más allá de la popularidad de estas dos principales estrategias (las cuales comparamos esquemáticamente en la Figura 2.2), existen otros métodos de recomendación que ofrecen ventajas considerables según la finalidad del recomendador. Entre ellos, encontramos los sensibles al contexto (Capítulo 7 del libro [25]), los cuales reconocen que las preferencias y necesidades de los usuarios pueden variar según el contexto en el que se encuentren, y tienen en cuenta otros factores como la ubicación, la hora del día, situación actual del usuario,... Otro grupo de gran importancia son los basados en dominios cruzados [30], los cuales utilizan información y patrones aprendidos en un contexto para mejorar la recomendación en un contexto diferente, siendo así especialmente útiles cuando se enfrentan a falta de datos en un nuevo dominio.

Por último, cada algoritmo de recomendación por separado puede presentar algunas desventajas en determinadas situaciones, por lo que es habitual combinar varios modelos para paliar esos problemas [31]. Esta combinación puede ser realizada de varias formas, como la fusión de resultados de diferentes algoritmos o la implementación de un enfoque híbrido donde los métodos individuales colaboran de manera coordinada. Inicialmente, proliferaron los métodos híbridos que combinan los enfoques de CF y CB, aunque debido a la gran expansión de los sistemas de recomendación, pronto han surgido otras técnicas que combinan algoritmos del mismo tipo de modelo de recomendación.

## 2.2. Sistemas de Recomendación del siguiente punto de interés

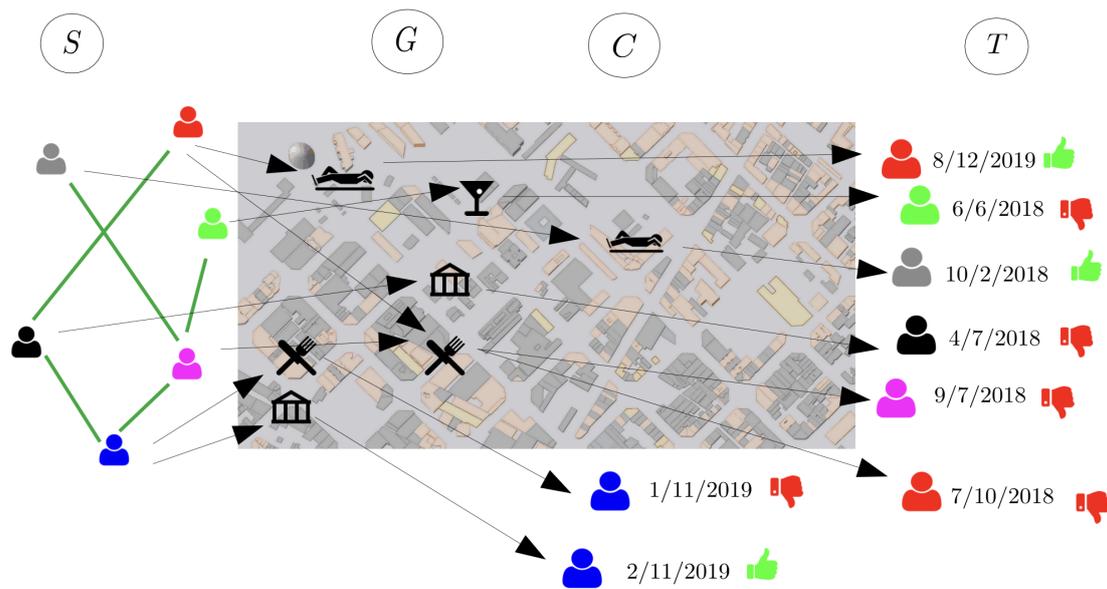
En este apartado, comenzaremos detallando en la Subsección 2.2.1 las claves de la recomendación de puntos de interés. Posteriormente, ampliaremos en la Subsección 2.2.2 las principales dificultades o retos que encontramos en estos recomendadores y seguidamente, los conceptos claves del procesamiento de este tipo de información (Subsección 2.2.3). Finalmente, terminamos repasando en la Subsección 2.2.4 la trayectoria del estado del arte, desde los métodos tradicionales hasta modelos mucho más vanguardistas como los algoritmos de aprendizaje profundo.

### 2.2.1. Definición del problema

El problema de la recomendación de Point of Interest (POI) se refiere a la tarea de sugerir a los usuarios nuevos lugares para visitar cuando llegan a una ciudad o región. En este contexto, los lugares pueden incluir museos, restaurantes, hoteles u otros puntos de interés. Afortunadamente, la mayoría de los datos que aplican estos sistemas de recomendación son recogidas por las LBSN, redes sociales orientadas en la localización, en las cuales los usuarios establecen vínculos sociales con otros usuarios, comparten información y registran visitas a lugares específicos de sus alrededores. Estas interacciones incluyen diversos tipos de datos, como la ubicación de los usuarios, los lugares que han visitado, las interacciones sociales entre ellos y sus recomendaciones, entre otros, como se puede observar en la Figura 2.3.

Adicionalmente, dentro estos recomendadores existen diversos enfoques según la finalidad de nuestra recomendación [32]. Un ejemplo fácil de visualizarlo es pensando en las posibles situaciones cotidianas donde podríamos tener la necesidad de una recomendación. Por un lado, podríamos querer ir a un tipo de local concreto a una hora determinada, o bien planear un viaje para toda la familia dentro de un tiempo. Por otro lado, igual nos interesa más las actividades que se puedan realizar o los objetos que se puedan comprar en según qué sitios, o los establecimientos más populares de una ciudad.

En nuestro estudio, veremos el estado de arte de todos estos diferentes enfoques, sin embargo, para nuestro modelo nos centraremos en el objetivo de recomendación del siguiente punto de interés, la cual se centra en ofrecer orientación en tiempo real, sugiriendo de manera dinámica el siguiente destino más adecuado mientras el usuario se encuentra en movimiento. Este objetivo es especialmente útil para entornos urbanos o durante viajes, donde la optimización del tiempo y la adaptación a las preferencias individuales son fundamentales.



**Figura 2.3:** Representación gráfica de la información generada por una LBSN, donde S correspondería a las relaciones sociales entre usuarios, G y C muestran las características geográficas y categóricas de los POIs y T guarda los ratings junto con su fecha. Obtenido de [12].

## 2.2.2. Problemas y desafíos

Aunque la tarea de recomendación de POI comparte similitudes con el problema clásico de recomendación, este presenta particularidades únicas. Algunas de estas limitaciones y características distintivas incluyen desafíos relacionados con la falta de datos, la complejidad de introducción de nuevos elementos o usuarios o el constante cambio de las tendencias y modas [12, 13].

### Escasez de datos

La escasez de datos es uno de los problemas más cruciales a la hora de crear un sistema de recomendación de lugares, debido a que las valoraciones o visitas son mucho más escasas que en la mayoría de los sistemas de recomendación. Desde el punto de vista del usuario, una persona viaja a muy pocos lugares a lo largo de su vida en comparación con el enorme número de puntos de interés que existen y que puede visitar. Además, explorar distintos POI cuesta mucho más que explorar distintas opciones en otros campos de la recomendación, lo que agrava aún más el problema de la escasez de datos. En consecuencia, las relaciones entre usuarios y POI descritas en los conjuntos de datos son realmente escasas. Por ejemplo, como se indica en [12], la densidad de los datasets de Netflix o Movielens (generalmente usados en otros recomendadores) es de 1.77 % y 0.537 %, respectivamente, mientras la de los datasets más empleados por los recomendadores de POI, Foursquare y Gowalla [33], es de 0.0034 % y 0.0047 % respectivamente. Por lo tanto, diseñar un sistema de recomendación eficaz con este conjunto de datos dispersos siempre supone un reto.

### Arranque en frío

Cuando un usuario se une a una red LBSN, la falta de una caracterización adecuada de ese usuario da como resultado un pobre rendimiento inicial de la recomendación. Del mismo modo, cuando se crea un nuevo POI para ser explorado, se queda muy por detrás de otros POI ya existentes en términos de recomendación debido a la falta de datos históricos asociadas con ese POI. Dado que este problema es común a la mayoría de los modelos de recomendación actuales, la eliminación del problema del arranque en frío (*cold start*) es una dirección de investigación prometedora en la recomendación de POI.

### Escasez de conjuntos de datos de referencia

Aunque en los trabajos anteriores se han utilizado en gran medida conjuntos de datos de check-ins populares, como Foursquare, Gowalla, Yelp o Weeplaces, la enorme escala de estos conjuntos de datos hace que sea imposible trabajar con todos a la vez. En consecuencia, la mayoría de los trabajos anteriores toman sólo un subconjunto (por ejemplo, los check-ins de una ciudad dentro de un marco temporal específico) de estos conjuntos de datos para evaluar los modelos propuestos. Además, los últimos avances en la recomendación de POI muestran que, además de los registros, los datos de otras modalidades como la descripción textual, las opiniones de los usuarios, etc., pueden mejorar el rendimiento de la recomendación. Sin embargo, los conjuntos de datos populares no contienen estos datos, lo que lleva a proponer nuevos conjuntos de datos (por ejemplo, el conjunto de datos de la red social Instagram) que los contengan. La variedad de conjuntos de datos utilizados en los modelos hace que sea muy difícil comparar el rendimiento entre los diferentes modelos más avanzados [12]. Un conjunto de datos de referencia que contenga datos de varias modalidades puede resolver estos problemas y crear una base sólida para evaluar el rendimiento de los nuevos modelos.

### Falta de aprendizaje en línea

La mayoría de los modelos de recomendación de POI del estado del arte utilizan el aprendizaje fuera de línea, es decir, modelos que se entrenan una sola vez con los datos disponibles en ese momento antes de su aplicación. En el mundo real, esta estrategia no es óptima, ya que los usuarios generan a diario toneladas de nuevos datos de registro que contienen información crucial sobre los cambios en las preferencias de los usuarios. Una estrategia de aprendizaje en línea que pueda actualizar los modelos a medida que se disponga de nuevos datos es, por tanto, de gran importancia para que el modelo pueda ofrecer un rendimiento óptimo de recomendación a lo largo del tiempo, incluso en circunstancias cambiantes.

### 2.2.3. Procesamiento de datos

Como ya se ha expuesto, la densidad de la mayoría de los conjuntos de datos de recomendación de POI es muy baja. Por este motivo, la gran mayoría de los recomendadores de POI utilizan más de una fuente de información, o bien emplean técnicas de aumento de datos.

Por un lado, las redes LBSN son de gran valor para los recomendadores de POIs debido a la gran cantidad de datos que pueden generar. Si bien los check-ins que registra la red ya informan de las preferencias de los usuarios, la hora y fecha de la interacción, la ubicación, archivos multimedia o las posibles reviews que escriban acerca de los ítems, estas redes también son capaces de ofrecer información adicional como la popularidad de una zona en general o en momentos específicos del día, la semana o el año. Todo esto nutre los recomendadores de POIs, los cuales se han sabido adaptar a las diferentes fuentes de información relacionadas con los ítems.

Por otro lado, aunque toda la información que rodea a un check-in es extensa, desgraciadamente la cantidad de check-ins registrados es muy inferior a otros sistemas de recomendación. Por ello, es frecuente que en los recomendadores de POIs se apliquen técnicas de aumento de datos con los cuales conseguir modelos más robustos [34]. En el caso de que nuestro input sean series de puntos de interés, podemos aplicar el uso de subsecuencias. Dado una secuencia  $l = (l_1, l_2, l_3, l_4, l_5)$ , dividimos la misma en 3 subsecuencias  $(l_1, l_2), (l_2, l_3, l_4), (l_3, l_4, l_5)$ , cuyo objetivo consiste en recomendar el último elemento de cada secuencia. De esta forma, obtendremos 4 secuencias en lugar de una sola, consiguiendo así reducir el sobreajuste y dar más peso a los elementos intermedios [35]. Otra estrategia para conseguir modelos más robustos, consiste en añadir ruido a nuestros datos, de tal forma que dada una serie  $(l_1, l_2, l_3, l_4, l_5)$ , podemos sustituir o bien añadir información con elementos neutros. Por ejemplo, al sustituir el tercer elemento de la serie por un elemento neutro como un vector nulo, tal que  $(l_1, l_2, 0, l_4, l_5)$ , conseguimos un modelo más robusto a la falta de información.

### 2.2.4. Modelos de recomendación NEXT-POI

A lo largo de las décadas, la evolución de los algoritmos de recomendación del siguiente Punto de Interés (NEXT POI) ha estado marcada por la adaptación a nuevas tecnologías y la disponibilidad de datos.

#### Métodos tradicionales

En comparación con la recomendación convencional de POI, la recomendación del siguiente POI se centra más en la influencia temporal de la trayectoria reciente para predecir los próximos movimientos de un usuario. En sus inicios, los sistemas se centraban en aquellos basados en funciones de similitud, o bien aplicando la similitud entre usuarios para proporcionar recomendaciones (Filtrado Colaborativo) [36], o bien centrándose en las características de los ítems y los perfiles de usuario, más

que en los usuarios (Filtrado basado en Contenido). Por ejemplo, Jia et al. [37] proponen un sistema recomendador de FC basado en el usuario para la recomendación de atracciones turísticas, que consta de tres pasos: representación de la información del usuario (turista), generación de usuarios vecinos (otros turistas) y generación de recomendaciones de ubicaciones. Otro ejemplo lo proponen Linden et al. [38], donde utilizan una variante de FC basado en ítems cuya recomendación final es una lista de productos de Amazon. Este método mide la similitud entre los productos basándose en los comportamientos de compra o valoración del usuario, y recomienda los productos que son similares.

Adicionalmente, los primeros estudios adoptaron otros métodos ampliamente utilizados en tareas de recomendación secuencial, como las cadenas de Markov [4, 5, 39]. Estos algoritmos, se basan en que las preferencias y comportamientos de los usuarios pueden ser modelados como un proceso estocástico, donde la probabilidad de transición entre diferentes estados (en este caso, POIs) depende únicamente del estado actual y no de la secuencia completa de estados anteriores. Como ejemplo, Zhang et al. [40] propusieron una cadena de Markov aditiva para modelar la influencia secuencial de las ubicaciones en los comportamientos de los usuarios.

### Factorización y probabilísticos

El surgimiento de las Máquinas de Soporte Vectorial (SVM) introdujo algoritmos de clasificación y regresión, seguidos por la Factorización Matricial, que transforma la matriz de usuarios e ítems en un espacio de baja dimensión que caracteriza la información en factores latentes. Esta última, es una de los enfoques más utilizados y por ejemplo, Lian et al. [41] presentan un método de recomendación de POI llamado GeoMF, que explota los registros de movilidad del usuario en las redes sociales basadas en la localización (LBSN) y modela esos datos de feedback implícito con factorización matricial ponderada para la recomendación de POI.

Los métodos probabilísticos, por otro lado, emplean modelos basados en la teoría de la probabilidad para modelar y prever las preferencias y comportamientos de los usuarios. Estos enfoques incluyen modelos de redes bayesianas que representan relaciones probabilísticas entre usuarios, POIs y características contextuales; cadenas de Markov ocultas que modelan las preferencias como secuencias de estados ocultos; o procesos gaussianos, los cuales ofrecen enfoques flexibles para gestionar la incertidumbre en las elecciones de los usuarios, permitiendo una recomendación más adaptativa y realista. Estos últimos no solo generan recomendaciones, sino que también cuantifican la incertidumbre asociada, brindando mayor robustez y precisión a medida que evolucionan las preferencias de los usuarios. Algunos ejemplos pueden ser encontrados en [9, 11, 42, 43].

### Aprendizaje profundo

A medida que la capacidad computacional aumentó, las Redes Neuronales se convirtieron en una opción poderosa para modelar relaciones complejas. Estas técnicas de aprendizaje profundo descu-

bren automáticamente las mejores características latentes de cada componente del sistema, evitando enormemente el tiempo y conocimientos necesarios para conseguirlas previamente en el aprendizaje automático tradicional. Esta ventaja les ha permitido tener aplicaciones en muy diversas áreas como los sistemas de recomendación, el reconocimiento de imágenes, el procesamiento de lenguaje natural, asistencia sanitaria, etcétera.

Aunque existen muchos tipos de redes neuronales dentro de la recomendación de punto de interés, algunas de las más usadas han sido recopiladas en [13]. Estas son: los modelos LSTM o redes neuronales de memoria a corto y largo plazo (por sus siglas en inglés, Long Short-Term Memory) [19], capaces de recordar información relevante de largo plazo y, al mismo tiempo, manejar y olvidar información menos importante a corto plazo; los modelos GRU [44], cuyas puertas de actualización les permiten decidir qué información retener y qué información descartar (computacionalmente más eficientes que los LSTM); Redes Neuronales Recurrentes (RNN) [15, 16], diseñadas para procesar secuencias de datos, donde la información se propaga a través de conexiones recurrentes, permitiendo la captura de dependencias temporales; y los modelos GAN [45], que mejora mediante la correcta discriminación entre los datos reales y datos generados sintéticamente.

### Modelos híbridos y de atención

Actualmente, los modelos híbridos, que combinan varios enfoques, han ganado popularidad al aprovechar las fortalezas individuales de cada algoritmo, marcando así una continua evolución en la búsqueda de recomendaciones más precisas y personalizadas. Por ejemplo, Zhao et al. [46] diseñan una red espacio-temporal (STGN) para incorporar información espacial y temporal y capturar las preferencias a corto y largo plazo para la recomendación de POI secuenciales. Chang et al. [47], por otro lado, propusieron un enfoque algo distinto aprovechándose de que la mayoría de los modelos no utilizan contenido textual debido a que no queda registrado en muchos conjuntos de datos. Por ello, el equipo de Chang generaron un nuevo conjunto de datos de la red social Instagram con la descripción textual de los POI escrita por los usuarios. El modelo que crearon, CAPE, consta de una capa de contexto de check-in que captura la influencia geográfica de los puntos de interés, y una capa de contenido textual, con la que captura las características de los mismos.

Por otro lado, algunos estudios han adaptado los mecanismos de atención a esta tarea de recomendación, como DeepMove y STAN [48, 49], los cuales se centran en partes específicas de los datos de entrada, mejorando así su capacidad de captar patrones relevantes y haciendo predicciones más precisas. DeepMove [48] propuso un modelo de atención para capturar el patrón de periodicidad multinivel y una red neuronal recurrente que modela las transiciones secuenciales para la recomendación final. STAN [49] empleó el mecanismo de autoatención para extraer las interacciones entre POI no adyacentes, algo que los modelos secuenciales no consiguen.

## 2.3. Redes neuronales basadas en grafos

A continuación, en la Subsección 2.3.1 describimos conceptos básicos acerca de las redes neuronales, con el fin de, seguidamente, entender mejor el contexto de las redes neuronales de grafos (Subsección 2.3.2).

### 2.3.1. Redes Neuronales artificiales

Las redes neuronales son modelos inspirados en el cerebro humano que se utilizan en la inteligencia artificial para realizar tareas complejas. Formadas por “neuronas” artificiales, estas redes presentan capas: una de entrada para datos, capas ocultas para aprender patrones, y una de salida para producir resultados. Cada conexión entre neuronas tiene un peso que se ajusta durante el entrenamiento para mejorar la precisión mediante la propagación hacia delante y hacia atrás.

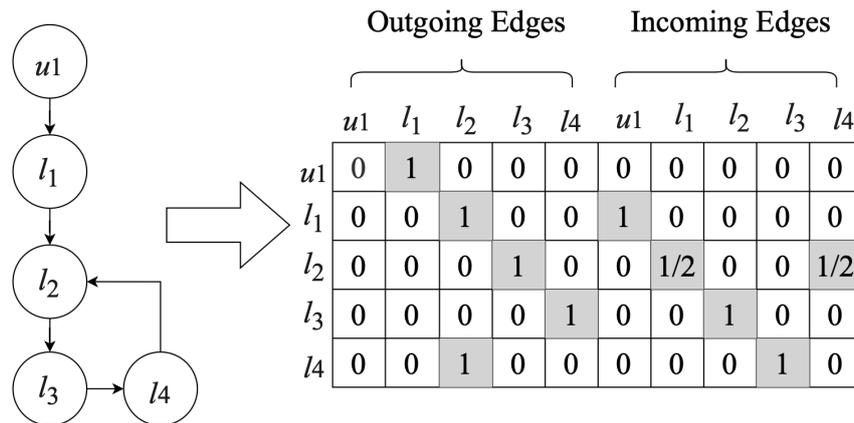
Durante la propagación hacia adelante, los datos se mueven a través de la red, y durante la propagación hacia atrás, la red ajusta sus pesos basándose en la diferencia entre sus predicciones y los resultados reales. La función de activación introduce no linealidades, permitiendo a la red aprender patrones complejos. Además, se utiliza una función de pérdida para medir la discrepancia entre las predicciones y los resultados reales, y la tasa de aprendizaje controla el tamaño de los ajustes durante el entrenamiento. Este proceso de entrenamiento, que implica iteraciones a través de los datos y ajustes de pesos, les permite aprender y generalizar a partir de patrones en los datos.

Inicialmente inspiradas en la estructura y función del cerebro, las perceptrones y las redes neuronales de una sola capa fueron los primeros modelos. Sin embargo, su capacidad para abordar problemas complejos era limitada. La introducción de las redes neuronales multicapa y el algoritmo de propagación hacia atrás permitieron superar estas limitaciones, posibilitando el aprendizaje de representaciones jerárquicas. En los últimos años, las redes neuronales profundas, o deep learning, han ganado prominencia, impulsadas por el aumento de datos y la mejora de la capacidad computacional. Estas redes, como las Redes Neuronales Convolucionales (CNN) para datos espaciales, las Redes Neuronales Recurrentes (RNN) para datos secuenciales, y las Redes Neuronales Generativas (GAN) para generación de datos, han demostrado tener éxito en tareas de clasificación de imágenes, detección de objetos, procesamiento de lenguaje natural, y más [50].

### 2.3.2. Redes Neuronales de Grafos

Las redes neuronales de grafos o basadas en grafos (GNN, por sus siglas en inglés: *Graph Neural Networks*) son una potente variante de las redes neuronales tradicionales, diseñadas para modelar y entender datos estructurados en forma de grafos. Estas redes aprovechan la estructura espacial de la información (como mapas o ubicaciones geográficas) para realizar tareas específicas, como la predic-

ción de propiedades en ubicaciones específicas, la clasificación de áreas geográficas o la generación de representaciones vectoriales para objetos en entornos espaciales. Los nodos representan las entidades o elementos, y las aristas capturan las relaciones entre ellos. La clave se encuentra en las capas de agregación de la red, donde la información de los nodos vecinos se procesa para actualizar las características de cada nodo, y las capas de actualización, posteriormente ajustan estas características. La capacidad de aprendizaje de GNNs radica en su habilidad para generalizar a nuevas tareas y adaptarse a diversas estructuras de grafo. Un ejemplo de una red de grafos junto con su matriz de conexiones se muestra en la Figura 2.4.



**Figura 2.4:** Ejemplo de una red de grafos simple junto con su matriz de conexiones  $A^{u_1}$ . Obtenido de [14].

Con la proliferación de datos estructurados en grafos, como los de redes sociales, han surgido muchos trabajos de investigación sobre GNN. En especial, las GNN y sus variantes, como las redes convolucionales de grafos (GCN) y las redes recurrentes de grafos (GRN), se han utilizado ampliamente en muchas tareas, como la predicción y la recomendación. Por ejemplo, Wang et al. [51] representan los datos de interacción entre usuarios y POI como grafos de conexión de orden superior y aprenden las preferencias personalizadas del usuario a partir de las interacciones complejas y no lineales de los grafos. Li et al. proponen GGNN [52], un modelo que utiliza puertas GRU en el paso de propagación. Este, aplica la red neuronal durante un número fijo de  $T$  pasos y seguidamente, se propaga hacia atrás en el tiempo para calcular los gradientes. Además, este modelo también se aplica en el campo de la recomendación de sesiones, en la cual se pretende predecir las preferencias de un usuario en función de su comportamiento reciente en una sesión de navegación o interacción online. Un estudio de ello fue el que adoptaron Wu et al. [53], el cual utiliza una red de atención para modelar la preferencia global del usuario y sus intereses actuales.

Asimismo, otras variantes incluyen en los sistemas de recomendación las redes gráficas de atención, que combinan GNN con redes de atención, dándole de esta forma mayor valor a partes específicas de los datos de entrada según el contexto. Por ejemplo, Song et al. [54] proponen un sistema de recomendación basado en una red neuronal de atención gráfica dinámica para comunicaciones online,

que modela separadamente los comportamientos dinámicos del usuario con una RNN, y la influencia social dependiente del contexto con una red de atención basada en grafos. En otro ejemplo, Wang et al. [55] proponen una Red de Atención de Grafo de Conocimiento (KGAT) para enlazar duplas usuario-artículo mediante atributos y fusionar el grafo usuario-artículo y el grafo de conocimiento de atributos, para formar una estructura de red unificada para la recomendación.

## 2.4. Redes neuronales de grafos en recomendación de POI

En el contexto de recomendación del siguiente punto de interés, los usuarios y elementos a recomendar (POI) suelen ser considerados nodos mientras que las conexiones Usuario-POI y las relaciones entre POIs serían las aristas del grafo. La idea principal está en explotar diversas influencias, como geográficas, sociales y temporales, para establecer y ponderar los vínculos entre estos nodos.

En general, este tipo de modelos aporta ventajas cruciales al representar de manera natural las relaciones espaciales entre diferentes POI, modelar relaciones complejas entre nodos de manera eficiente o permitir cambios en el tiempo mediante la inclusión de aristas temporales, entre otras. A continuación, hemos seleccionado y detallado algunos de los trabajos con redes neuronales más representativos de los últimos años, centrándonos en aquellos que dan acceso al propio modelo en un repositorio público, independientemente o no de que sean basados en grafos. Mostramos los resultados de este estudio del estado del arte en la Tabla 2.1.

Por un lado, podemos observar cómo, a pesar de haber realizado el estudio entre los años 2016 y 2023, muchos de los trabajos cuyos modelos están basados en grafos corresponden a los últimos años revisados, de 2020 en adelante. Esto es debido a que existe una tendencia cambiante a lo largo de este periodo de tiempo, donde los modelos más representativos de los primeros años fueron las Redes Neuronales Recurrentes (RNN) y modelos más básicos como los Autoencoders o las LSTM, y posteriormente se han ido sustituyendo por modelos más complejos, los cuales incluyen mecanismos de atención, o Redes Neuronales de Grafos.

Por otro lado, cabe destacar cómo los datasets de Gowalla y Foursquare son utilizados en la mayoría de trabajos. Esto es debido a que actualmente, junto a Brightkite o Yelp, son los mayores conjuntos de datos de POIs existentes y, por tanto, los que más se han estudiado dentro del estado del arte.

### 2.4.1. Selección de artículos

Después de revisar la literatura para llevar a cabo este trabajo, presentamos a continuación algunos proyectos que nos han llamado la atención. Los tres modelos se han tenido bajo consideración como estudio de referencia, sin embargo, hemos seleccionado únicamente el más completo de ellos.

Cita	Año	Acrónimo	Modelo	Dataset	Código (siempre en GitHub)
[15]	2016	ST-RNN	RNN	Gowalla	yongqyu/STRNN
[56]	2017	DCPR	RNN, CNN, MF	Foursquare, Yelp, TIST	fengjiaowang7/DCPR
[57]	2017	VPOI	CNN	ImageNet	NO
[58]	2017	PACE	NN	Gowalla, Yelp	yangji9181/PACE2017
[59]	2017	DRCF	RNN y MLP	Foursquare, Brightkite, Yelp	qnfwnkd/DRCF
[60]	2018	RECNET	DNN	Gowalla, Foursquare	Mrdongsonny/POI-recommendation
[61]	2018	SAE-NAD	AE	Gowalla, Foursquare, Yelp	allenjack/SAE-NAD
[18]	2018	CARA	RNN	Foursquare, Brightkite and Yelp	feay1234/CARA
[62]	2018	TMCA	LSTM, AE	Gowalla, Yelp	zhenqi/TMCA
[63]	2019	TCENR	RNN + CNN	Yelp	omer-tal/TCENR
[45]	2019	APOIR	GRU, GAN, MF	Gowalla, Foursquare, Yelp	APOIR2018/APOIR
[64]	2020	GeoSAN	Attention AE	Gowalla, Foursquare, Brightkite	libertyeagle/GeoSAN
[20]	2020	LSTPM	RNN	Gowalla, Foursquare	NLPWM-WHU/LSTPM
[14]	2021	<b>ASGNN</b>	Attention GNN	Gowalla, Foursquare	HduDBSI/ASGNN
[49]	2021	STAN	AN (Attention Network)	Gowalla	yingtaoluo/STAN
[65]	2021	DAN-SNR	Deep AN	Gowalla, Brightkite	drhuangliwei/DAN-SNR
[21]	2021	<b>GEAPR</b>	Graph AN, MF	Yelp	zyli93/GEAPR
[66]	2021	LightMove	AN	Foursquare, Taxi dataset	Jinsung-Jeon/LightMove
[67]	2021	ATST-LSTM	Attention LSTM	Gowalla, Brightkite	drhuangliwei/ATST-LSTM
[68]	2022	VGMF	CNN, MF	No especifica	socialsnail/VGMF
[22]	2022	PPR	Graph LSTM	Gowalla, Foursquare, Brightkite	dsj96/PPR-master
[44]	2022	SelfTrip	GRU	Flickr	NO
[16]	2022	<b>HMT-GRN</b>	Graph RRN	Gowalla, Foursquare	poi-rec/HMT-GRN
[17]	2022	GETNext	Attention GNN	Gowalla, Foursquare	songyangme/GETNext
[69]	2022	NSSR	AN	Foursquare	conquerSelf/NSSR
[23]	2023	DisenPOI	GNN	Foursquare, Meituan	Yifang-Qin/DisenPOI
[19]	2023	DeepAltTrip	GNN, LSTM	Flickr	smrashid20/DeepAltTrip

**Tabla 2.1:** Resumen de los trabajos más representativos de recomendación de POI con redes neuronales en los últimos años ordenados por año de publicación. En negrita, los más destacados.

## Red neuronal recurrente gráfica jerárquica para multitarea

En este trabajo [16] se aborda el problema de la escasez de datos en la matriz Usuario-POI, obstáculo común en trabajos anteriores de recomendación del siguiente punto de interés. Para superar esta limitación, los autores proponen un enfoque denominado Red Recurrente Gráfica Jerárquica Multitarea (HMT-GRN), que se centra en el aprendizaje de diferentes matrices Usuario-Región con menor escasez en un entorno multitarea.

En resumen, el modelo que proponen supera la escasez de relaciones en la matriz Usuario-POI mediante la incorporación de matrices Usuario-Región de diferentes niveles de granularidad. El enfoque utiliza una búsqueda jerárquica de haces (HBS) para reducir el espacio de búsqueda, mejorando así la eficiencia en la predicción del próximo POI. Además, el modelo incorpora un módulo de Red Recurrente de Grafos (GRN) para capturar dependencias secuenciales dentro de las secuencias de visitas de los POI, considerando las relaciones POI-Región y POI-Tiempo. Por último, para equilibrar la personalización y la exploración, también proponen una capa balanceadora que decide si el siguiente POI ha sido visitado históricamente por el usuario o no.

Concluyen remarcando cómo las dos nuevas implementaciones, las matrices Usuario-Región de diferentes niveles de granularidad y la capa de selectividad de explotación-exploración, alivian el problema de la escasez de datos, mejorando los resultados de los baseline principales.

### Red de atención mejorada por grafos para la recomendación de POI explicables

Por otro lado, el modelo GEAPR [21] fue diseñado para solventar dos limitaciones de muchos modelos de recomendación de POI: la falta de consideración de los atributos individuales de los usuarios y la carencia de explicaciones claras sobre por qué se hacen ciertas recomendaciones. Para ello, el modelo tiene en cuenta diferentes factores, como el contexto estructural, el impacto de los vecinos, los atributos del usuario y la influencia de la geolocalización, unido a una capa de atención triple para cuantificar estas influencias y garantizar la interpretabilidad del modelo.

Para finalizar, demuestran mediante los experimentos cómo esa división de la motivación de los POIs en 4 factores distintos, ayudan a mejorar los resultados de otros modelos del estado del arte. Además, instan en futuros trabajos, a explotar los atributos de los puntos de interés mediante grafos de semántica y a estudiar formas de preservar la no linealidad de las características geográficas.

### Modelo secuencial de atención basado en redes neuronales gráficas

Mediante este proyecto, Wang et al. [14] emplean ASGNN, una mezcla de modelo de atención secuencial y red neuronal de grafos, para disuadir el problema de la sobrecarga de información en la era de la tecnología, donde la gran cantidad de datos disponibles en Internet dificulta que los usuarios encuentren contenido o servicios relevantes. Este modelo se beneficia de las redes de atención para capturar las preferencias a largo y corto plazo del usuario y, posteriormente, de las redes neuronales basadas en grafos para aprender los vectores de características latentes informativas de baja dimensión (embeddings) de los POI. Esto además solventa otro problema relacionado con que los métodos de recomendación tradicionales, a menudo, descuidan los cambios dinámicos en las preferencias a corto plazo, limitando su rendimiento.

EL modelo ASGNN consta de cuatro pasos principales:

- 1.– Construcción de un grafo de secuencia de check-ins de POI.
- 2.– Aprendizaje de representación de características.
- 3.– Captura de preferencias a largo/corto plazo.
- 4.– Recomendación del siguiente punto de interés.

En los experimentos se prueba y compara el modelo bajo diferentes escenarios y con otros métodos tradicionales y del reciente estado del arte, y tras ello, los resultados demuestran que ASGNN supera a los modelos de referencia en la tarea de recomendación del próximo POI. Finalmente, dejan abierta la puerta para que en el futuro se utilicen GNN heterogéneas para explorar completamente diversos tipos de información auxiliar, como contextos temporales, metadatos, correlaciones secuenciales, etc., con el objetivo de mejorar aún más el rendimiento de la recomendación.

## 2.4.2. Discusión

A pesar de que los tres trabajos son bastante innovadores dentro de los sistemas de recomendación de POI, hemos seleccionado el tercero de ellos puesto que hemos concluido que era el más completo de ellos. Por un lado, el primer artículo mezcla información secuencial captada por la red recurrente considerando diferentes distribuciones geográficas y la posibilidad de cambiar la personalización o exploración del modelo. Por otro lado, el segundo afronta la tarea de recomendación desde una perspectiva bastante amplia, incorporando varios tipos de información adicionales. Sin embargo, el trabajo seleccionado completa la información anterior, teniendo en cuenta no solo las preferencias a largo plazo si no también a corto plazo, y modelando dichas secuencias como grafos, de forma que la red neuronal de grafos pueda captar completamente relaciones complejas del sistema.

## 2.5. Evaluación de los Sistemas de Recomendación

La evaluación de los sistemas de recomendación es esencial para medir su eficacia y precisión en proporcionar sugerencias relevantes a los usuarios. Dada la diversidad de enfoques, desde el filtrado colaborativo hasta los modelos híbridos y los sensibles al contexto, la evaluación se convierte en un proceso clave para comprender cómo estos sistemas se desempeñan en entornos del mundo real. A continuación se describen algunas de las métricas más importantes de estas estrategias:

Error absoluto medio (MAE o Mean Square Error en inglés): esta métrica mide el promedio de los valores absolutos entre las valoraciones predichas y las reales.

$$MAE = \frac{1}{|K|} \sum_{(u,i) \in K} r_{ui} - \hat{r}_{ui} \quad (2.2)$$

Error cuadrático medio (RMSE o Root Mean Squared Error): se diferencia del MAE por usar la raíz cuadrada del error medido en lugar del valor absoluto de este. Además, en términos generales penaliza más cuanto mayor sea la diferencia.

$$RMSE = \frac{1}{|K|} \sum_{(u,i) \in K} \sqrt{(r_{ui} - \hat{r}_{ui})^2} \quad (2.3)$$

Concretamente, el RMSE y el MAE son métricas útiles en problemas de regresión, especialmente en sistemas de recomendación que pronostican valoraciones numéricas de ítems ya visitados. No obstante, esto no refleja correctamente problemas del mundo real, y dentro del contexto de recomendación de NEXT POI, trabajaremos generalmente con un ranking de nuevas recomendaciones, independientemente de si han sido vistas previamente o no, por lo que serán más comunes aquellas métricas que comparen los elementos según se han recomendado correcta o incorrectamente. Algunas de ellas son:

Precisión (Accuracy en inglés): Mide la proporción de elementos recomendados que son relevantes para el usuario. Se calcula como el número de elementos relevantes recomendados dividido por el número total de elementos recomendados. Dentro del contexto de recomendación de NEXT POI, tenemos que tener en cuenta que generalmente trabajaremos con un ranking de recomendaciones. Es por eso que únicamente se evalúan los  $k$  primeros lugares recomendados.

$$P@k = \frac{1}{|U|} \sum_{u \in U} \frac{Rel_u@k}{k} \quad (2.4)$$

Recall (Recuperación): esta métrica evalúa la proporción de elementos relevantes que fueron recomendados correctamente. Se calcula como el número de elementos relevantes recomendados dividido por el número total de elementos relevantes. De nuevo, debemos tener en cuenta únicamente los  $k$  primeros elementos recomendados.

$$R@k = \frac{1}{|U|} \sum_{u \in U} \frac{Rel_u@k}{Rel} \quad (2.5)$$

En las ecuaciones 2.4 y 2.5,  $Rel_u@N$  corresponde al número de ítems relevantes para el usuario  $u$  y recomendamos hasta la posición  $k$  del ranking y  $Rel_u$  el número de sitios relevantes para el usuario  $u$ .

Rango recíproco medio (MRR o Mean Reciprocal Rank): en este último caso, MRR representa el promedio del inverso de los rangos de los POIs objetivo en una lista de recomendaciones. Es decir, únicamente evaluamos el puesto del primer elemento recomendado correctamente dentro de los  $k$  primeros elementos devueltos. Cuanto mayor sea esta métrica, mejor serán las recomendaciones.

$$MRR@k = \frac{1}{k} \sum \frac{1}{Rango(k_i)} \quad (2.6)$$

Finalmente, existen otras métricas menos utilizadas que reconocen la importancia de factores como la originalidad y la variedad de las recomendaciones (Capítulo 8 del libro [25]), como la novedad que destaca la importancia de sugerir elementos no explorados previamente por el usuario, o la diversidad, que se enfoca en ofrecer una gama amplia de sugerencias, asegurando que la selección de elementos no esté limitada a una categoría o tipo específico.



# DISEÑO E IMPLEMENTACIÓN

---

En este capítulo detallamos los pasos necesarios previos a la aplicación del modelo, así como una descripción de las cuatro componentes principales del modelo y sus variantes. En primer lugar, comenzamos detallando el entorno de trabajo sobre el que trabajamos (Sección 3.1). A continuación, en la Sección 3.2 describimos en mayor profundidad las cuatro componentes principales que tiene nuestro modelo y, por último, presentamos las diferentes variaciones del modelo a estudiar junto con sus hipótesis previas a su ejecución en la Sección 3.3.

## 3.1. Entorno de trabajo

La implementación y la realización de todas las pruebas se han realizado bajo un mismo entorno de trabajo, el que ofrece la máquina virtual de Google a través de Google Colaboratory. A pesar de que los requerimientos de nuestro modelo están recogidos en el repositorio público del trabajo de referencia <sup>1</sup>, se ha elegido este entorno debido a su fácil configuración basada en la nube y a que incluye gran parte de las librerías más comunes de Python. Esto ha permitido la ejecución en paralelo desde dos sesiones diferentes de Google Colab, reduciendo considerablemente el tiempo de ejecución de las pruebas. Las especificaciones de este entorno están recogidas en la Tabla 3.1.

Recursos	Características
CPU	Intel Xeon
GPU	Tesla K80
RAM	13 GB
S.O.	Ubuntu 22.04
Python	3.10

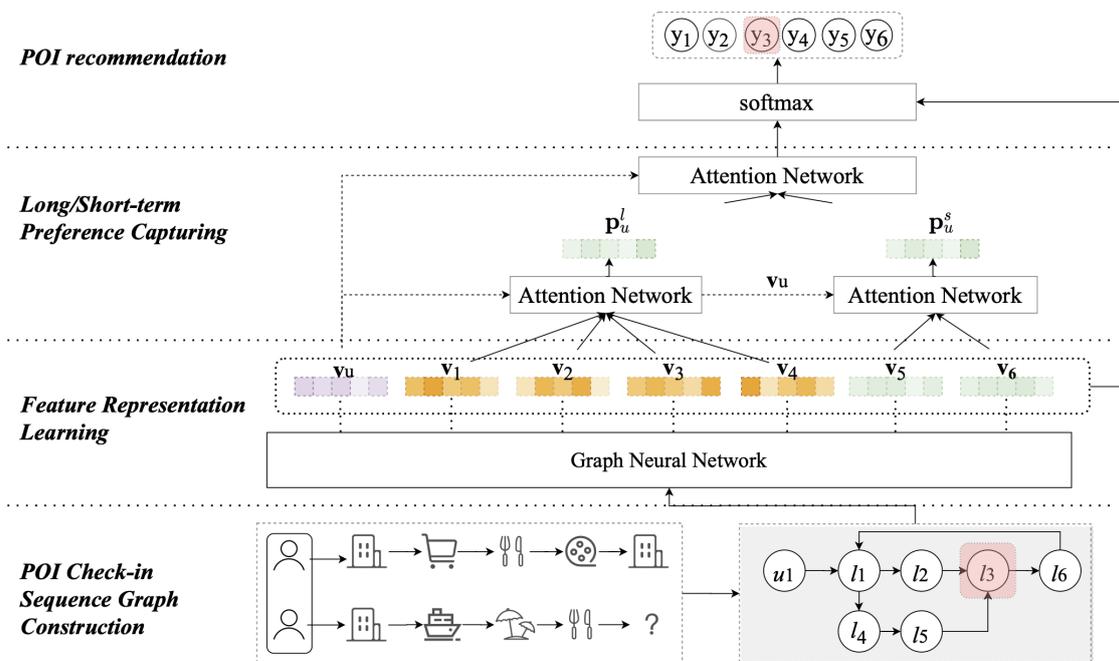
**Tabla 3.1:** Recursos utilizados para los experimentos.

---

<sup>1</sup> <https://github.com/HduDBSI/ASGNN>

## 3.2. Descripción del modelo

Como se muestra en la Figura 3.1, el modelo ASGNN [14] se divide en cuatro componentes bien diferenciadas: 1) Construcción de un grafo de secuencia de check-ins de POI, 2) Aprendizaje de representación de características, 3) Captura de preferencias a largo/corto plazo y 4) Recomendación del siguiente punto de interés.



**Figura 3.1:** Arquitectura general del modelo ASGNN [14]. Las cuatro divisiones son: 1) Construcción de un grafo de secuencia de check-ins de POI, 2) Aprendizaje de representación de características, 3) Captura de preferencias a largo/corto plazo y 4) Recomendación del siguiente punto de interés.

### 3.2.1. Contrucción del grafo de la secuencia de check-ins

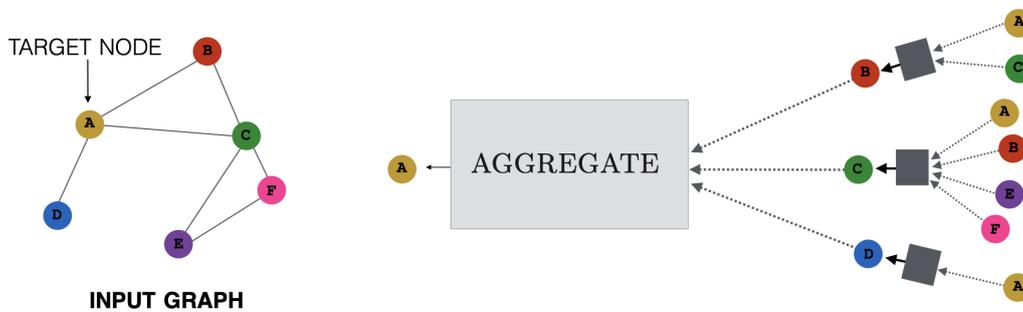
En primer lugar, el modelo representa los comportamientos de check-in mediante la construcción de un grafo de secuencia de check-ins de puntos de interés en lugar de utilizar secuencias directas, como en otros métodos. Esta elección ofrece varios beneficios claves. Primeramente, al no truncar las secuencias de comportamiento del usuario a una longitud determinada, se conserva más información secuencial imprescindible para las tareas de predicción y recomendación. Además, la representación en grafo permite capturar patrones repetitivos de comportamiento al igual que una propagación más efectiva de preferencias y características locales de check-in del usuario, mejorando en general la recomendación.

Concretamente, el grafo modela cada secuencia de check-ins ( $S$ ) como un grafo dirigido  $G(V, E)$  con conjuntos de vértices ( $V$ ), que representan los usuarios ( $U$ ) y POIs ( $L$ ), y las aristas ( $E$ ), que

incluyen conexiones usuario-POI y POI-POI. Los pesos de las aristas indican las frecuencias de check-in de los usuarios en los POIs, normalizadas por el grado de salida del vértice de inicio.

### 3.2.2. Aprendizaje de la representación de características

Después de la construcción del grafo de secuencia de check-ins de puntos de interés, el modelo aprende la representación vectorial de baja dimensión de POI y usuario a través de una Red Neuronal de Grafos (GNN). Cada POI  $l_{u,i} \in L$  y usuario  $u \in U$  se incorpora en un espacio de características latentes, y el vector del nodo  $v \in R^d$  representa un vector de características latentes/preferencias (o embedding en inglés) de un POI o usuario, donde  $d$  es la dimensión de los vectores. Los embeddings aprendidos pueden capturar las características intrínsecas de los POIs y representar las preferencias del usuario por los POIs. En la Figura 3.2 se muestra cómo se propagaría la información en una red neuronal de grafos.



**Figura 3.2:** Visión general de cómo un único nodo agrega mensajes de su vecindario local. La red agrega mensajes de los vecinos del grafo local de A (es decir, B, C y D) y, a su vez, los mensajes procedentes de estos vecinos se basan en la información agregada de sus respectivos vecindarios, y así sucesivamente. Obtenida de [70]

En los modelos GNN convencionales, es común adoptar el proceso de decisión de Markov, el cual es un modelo matemático de aprendizaje por refuerzo que permite describir la toma de decisiones en situaciones secuenciales y estocásticas. No obstante, la clave de este modelo reside en que la elección de una acción depende únicamente del estado actual y no de la secuencia completa de estados anteriores y, en nuestro caso, es computacionalmente ineficiente actualizar iterativamente los estados ocultos de todos los nodos. Para superar estas limitaciones, se adopta una estrategia que utiliza una variante de las redes neuronales de grafos llamada GGNN (Gated Graph Neural Network), la cual incorpora un mecanismo de puerta inspirado en las Redes Neuronales Recurrentes (RNN) para mejorar la eficacia de la propagación de información a lo largo del grafo. Este enfoque permite aprovechar de manera más efectiva los datos de check-in del usuario y capturar sus preferencias a largo plazo de manera más precisa.

Concretamente, la función de actualización de las características del nodo  $v_i$  de nuestro grafo está definida como:

$$a_{s,i}^t = A_i^u \cdot [v_1^{t-1}, \dots, v_n^{t-1}]^T H + b \quad (3.1a)$$

$$z_{s,i}^t = \sigma(W_z a_{s,i}^t + U_z v_i^{t-1}) \quad (3.1b)$$

$$r_{s,i}^t = \sigma(W_r a_{s,i}^t + U_r v_i^{t-1}) \quad (3.1c)$$

$$\tilde{v}_l^t = \tanh(W_o a_{s,i}^t + U_o (r_{s,i}^t \odot v_i^{t-1})) \quad (3.1d)$$

$$v_i^t = (1 - z_{s,i}^t) \odot v_i^{t-1} + z_{s,i}^t \odot \tilde{v}_l^t \quad (3.1e)$$

La Ecuación 3.1a describe la propagación de información, donde  $A$  es la matriz de adyacencia generada como en el ejemplo de la Figura 2.4,  $[v_1, \dots, v_n]$  son los embeddings de los nodos en el tiempo  $t - 1$ ,  $H$  correspondería a la matriz de pesos y  $b$  la componente que introduce el sesgo. Asimismo, las Ecuaciones 3.1b y 3.1c calculan las componentes de actualización y reinicio, ponderadas por las matrices de pesos  $W$  y  $U$  entrenadas durante el entrenamiento del modelo y la función sigmoide  $\sigma$ . En 3.1d se predice el posible siguiente estado del nodo  $i$ , de nuevo multiplicado por matrices de pesos provenientes del proceso de entrenamiento, y finalmente, la Ecuación 3.1e define cómo actualizar el estado de cada nodo basándose en el estado oculto anterior y el estado candidato con la puerta de actualización.

### 3.2.3. Captura de las preferencias a corto y largo plazo

Numerosos enfoques de recomendación existentes [49, 66] incorporan mecanismos de atención para capturar las preferencias a corto y largo plazo. Sin embargo, estos métodos a menudo descuidan las necesidades personalizadas y los patrones de comportamiento del usuario sumando ambas preferencias de manera lineal. Específicamente, asignan típicamente un peso fijo a las preferencias a corto y largo plazo del usuario, a pesar de que estas pueden desempeñar roles distintos en las tareas de predicción y recomendación. Para evitar esta limitación, el modelo emplea una red de atención jerárquica personalizada para obtener las preferencias a corto y largo plazo del usuario y combinarlas de forma adaptativa.

En concreto, el mecanismo de atención de dos capas que se aplica en esta propuesta puede captar información importante, como las preferencias del usuario con precisión, asignando diferentes pesos a los registros de entrada en secuencias de comportamiento y aprovechar las preferencias a corto y largo plazo del usuario de forma adaptativa para mejorar la recomendación. Por un lado, las preferencias a largo plazo para un usuario  $u$  se determinan mediante las siguientes ecuaciones:

$$h_i^l = \phi(W_l v_i + b_l) \quad (3.2a)$$

$$\alpha_{u,i}^l = \frac{\exp(v_u^T h_i^l)}{\sum_i \exp(v_u^T h_i^l)} \quad (3.2b)$$

$$p_u^l = \sum_i \alpha_{u,i}^l v_i \quad (3.2c)$$

donde, en primer lugar, las características latentes de los nodos que recorre  $u$ ,  $v_i$ , pasan por un simple perceptrón multicapa (MLP) con el fin de obtener la representación oculta  $h$  ( $W$  y  $b$  son parámetros del modelo y  $\phi$  la función de activación ReLU). Seguidamente, se calcula el vector de atención  $\alpha_{u,i}^l$  como la similitud entre  $h$  y  $u$  normalizada por la función softmax y finalmente, se suma ponderada de los nodos da lugar a las preferencias a largo plazo.

Siguiendo el mismo procedimiento, podemos calcular las preferencias a corto plazo, esta vez, utilizando únicamente el último check-in registrado en lugar de la secuencia de POIs completa.

$$h_i^s = \phi(W_s v_i + b_s) \quad (3.3a)$$

$$\alpha_{u,i}^s = \frac{\exp(v_u^T h_i^s)}{\sum_i \exp(v_u^T h_i^s)} \quad (3.3b)$$

$$p_u^s = \sum_i \alpha_{u,i}^s v_i \quad (3.3c)$$

Para terminar, combinamos ambas preferencias a corto y largo plazo mediante dos pesos independientes ( $\beta_l$  y  $\beta_s$ ) con el fin de obtener las preferencias personalizadas del usuario  $u$ .

$$p_u = \beta_l p_u^l + \beta_s p_u^s \quad (3.4)$$

### 3.2.4. Recomendación de POI

Finalmente, el modelo aplica una capa softmax al producto de las preferencias del usuario y los factores latentes en los nodos, para calcular la posibilidad de que un usuario objetivo  $u$  interactúe con el POI  $l_i$  seleccionado:

$$\hat{y} = \text{softmax}(p_u^T v_i) \quad (3.5)$$

Para evaluar completamente cómo está aprendiendo y mejorando nuestro modelo, se aplica la función de pérdida de entropía cruzada:

$$\mathcal{L}(\hat{y}) = - \sum_{i=1}^m y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (3.6)$$

donde  $y$  denota las preferencias reales e  $\hat{y}$  nuestras estimaciones según la Ecuación 3.5.

### 3.3. Implementación

En esta última sección planteamos la serie de experimentos que aplicamos al modelo ASGNN con el fin de estudiar su comportamiento, e incluso mejorar los resultados basándonos en las hipótesis presentadas.

#### 3.3.1. Variaciones

En la primera serie de experimentos buscamos modificar algunas partes del sistema de recomendación de POI con el fin de encontrar una variante que mejore el modelo actual. Es decir, manteniendo la mayoría de las partes iniciales del modelo empleado en [14], realizamos un estudio de posibles modificaciones cuyas combinaciones generen variantes al modelo inicial con resultados mejores a los obtenidos.

Lo primero que hacemos es identificar las posibles modificaciones al modelo, ya que es posible que, dependiendo de la componente modificada, el modelo no esté preparado para entrenar los datos de la misma forma y acabe generando errores. Tras ello, el siguiente paso es analizar los cambios a estas componentes junto con las hipótesis sobre las implicaciones de cada modificación. En la Tabla 3.2, mostramos las componentes que finalmente se van a modificar junto con sus posibles modificaciones.

Nº	Componente	Modelo básico	Variante
V1	Red neuronal de grafos (GGNN)	Celda GRU	Celda RNN básica
V2	Función de activación	F. Softmax	F. Sigmoid
V3			F. Tanh
V4			F. ReLU
V5	Optimizador	ADAM	Descenso por gradiente
V6			RMSProp
V7			Momentum
V8			Adagrad
V9			Adadelata
V10	Capa Dropout	No incluida	Dropout (ratio de 0.3)
V11	Regularizador	L2	L1
V12			L1 + L2
V13			Sin nada
V14	Mecanismo de atención	Dos niveles	Red Neuronal de Atención
V15			Atención Multinodal (2 nodos)

**Tabla 3.2:** Resumen de las principales componentes del modelo base, junto con las variantes seleccionadas.

A continuación, analizamos brevemente las implicaciones que tiene cada una de las modificaciones. Para poder comprobar cómo de acertadas o erróneas han sido, en el siguiente capítulo se van a probar todos los cambios mostrados, independientemente de si mejoran o no el resultado:

- **V1:** Este cambio (utilizar celda RNN básica en lugar de celda GRU) puede simplificar el modelo y reducir su complejidad computacional. En conjuntos de datos pequeños, una celda RNN básica puede ser suficiente. Mejoraría el coste computacional, pero puede verse afectada la capacidad de modelado de la información a largo plazo.
- **V2:** La función de activación de una neurona artificial permite encender o apagar las neuronas para obtener la respuesta deseada. En este caso, la sigmoide puede ser útil para predecir una preferencia binaria. Los datos no se asemejan por lo que esperamos peor rendimiento.
- **V3:** Útil para respuestas centradas alrededor de cero, la función tangente puede ser efectiva en problemas de recomendación donde las preferencias de los usuarios pueden ser tanto positivas como negativas. Al igual que para V2, no esperamos mejor rendimiento ya que estudiamos frecuencias de POI.
- **V4:** La función ReLU es conocida por su simplicidad y eficiencia en términos de cómputo. Puede ser útil cuando se trabaja con modelos profundos y se desea una función de activación que permita el aprendizaje más rápido (naturaleza no lineal). Funcionará bien en los problemas sin extrema complejidad.
- **V5:** El optimizador de una red neuronal corresponde a la estrategia tomada para ajustar gradualmente los pesos del modelo y, en consecuencia, los resultados obtenidos. El descenso por gradiente es lento pero estable. Puede funcionar bien si los datos son lineales o tienen características simples.
- **V6:** Este optimizador adapta el aprendizaje de la red en función de la diferencia de resultados. Más complejo que el descenso por gradiente, debería mejorar la convergencia.
- **V7:** Similar al descenso por gradiente, pero incorpora una especie de inercia que ayuda a acelerar la convergencia. Similar a V7.
- **V8:** Este optimizador adapta la tasa de aprendizaje de forma individual para cada peso según el gradiente. En determinados problemas puede llegar a la convergencia antes.
- **V9:** Versión mejorada del optimizador Adagrad. Parecido a Adagrad, pero con convergencia más rápida.
- **V10:** La capa de dropout previene el sobreajuste de la red apagando de forma temporal parte de las neuronas. Podría mejorar la capacidad del modelo para generalizar a datos de prueba no vistos y reducir el sobreajuste a los datos de entrenamiento.
- **V11:** El regularizador es una componente de la red que previene el sobreajuste al introducir penalizaciones adicionales según va aprendiendo la red. El L1 es muy simple y puede reducir la capacidad de captar relaciones complejas.
- **V12:** La combinación podría proporcionar un modelo más equilibrado en términos de complejidad, mejorando el rendimiento del modelo.
- **V13:** Eliminar el regularizador podría ser beneficioso si el modelo actual está infraajustado y se necesita una mayor capacidad de aprendizaje.
- **V14:** Introduce, con respecto al modelo de atención de dos niveles, un capa densa en la red, lo que permite captar mejor representaciones más complejas y no lineales. Dependiendo de si el modelo es muy simple, podría mejorar los resultados.
- **V15:** De nuevo, un mecanismo más complejo que podría capturar mejor relaciones complejas. Sin embargo, según los datos puede que empeore computacionalmente y no mejorem las recomendaciones.

### 3.3.2. Estudio de hiperparámetros

Una vez identificadas las posibles modificaciones al modelo base, la siguiente serie de experimentos consiste en realizar un estudio de hiperparámetros del modelo con el fin de optimizar los resultados obtenidos en las nuevas versiones. En general, los hiperparámetros son parámetros externos a un mo-

delo de aprendizaje automático que no se aprenden directamente del conjunto de datos durante el entrenamiento, si no que se establecen antes de iniciar el proceso de entrenamiento y afectan al comportamiento y rendimiento del modelo. Para ASGNN, definimos los siguientes parámetros (sus valores iniciales están agrupados en la Tabla 3.3) junto con sus principales implicaciones:

- **Tamaño oculto y tamaño de salida:** Controlan la complejidad de los datos de entrada y salida. Valores más grandes pueden mejorar la capacidad del modelo, pero también aumentarán la complejidad computacional.
- **Tamaño del batch:** Determina la cantidad de datos sobre los que entrena el modelo al mismo tiempo. Tamaños de lote más grandes pueden acelerar el entrenamiento, pero también requieren más memoria GPU.
- **Tasa de aprendizaje y tasa de aprendizaje en decaimiento:** Ambos determinan lo que aprende la red con el paso de las épocas. Un valor de aprendizaje demasiado alto puede llevar a la divergencia, mientras que uno demasiado bajo puede provocar una convergencia muy lenta.
- **Regularización L2:** La regularización L2 ayuda a prevenir el sobreajuste, pero un valor demasiado alto puede suprimir los pesos, llegando a un modelo demasiado generalizado.
- **Pasos de propagación de la red neuronal:** Determina cuántos pasos de tiempo se realizarán durante el entrenamiento y la inferencia de la red neuronal. Un número mayor de pasos puede permitir que la red capture relaciones más a largo plazo, pero también aumentará el tiempo de entrenamiento.
- **Añadir cálculo de atenciones:** Un modelo híbrido puede capturar mejor la información tanto de los nodos como de las conexiones entre ellos, pero puede ser más complejo. En nuestro caso la modificación es eliminar el mecanismo de atención del modelo.
- **Ratio de abandono (en el caso de que tuviera):** El abandono (o dropout) en una red neuronal se encarga de “apagar” parte de las neuronas para ayudar a prevenir el sobreajuste. Darle demasiado peso puede afectar la capacidad del modelo para aprender.
- **Número de nodos (para el modelo de atención multinodal):** Más nodos pueden capturar patrones más complejos, pero también aumentan la complejidad del modelo.
- **Pasos hasta el decaimiento de la tasa de aprendizaje:** De nuevo, el decaimiento programado de la tasa de aprendizaje puede ayudar a mejorar la convergencia.
- **Parámetros iniciales:** La inicialización de pesos adecuada puede influir en la velocidad de convergencia y el rendimiento del modelo.

Hiperparámetros	Valor actual
Tamaño oculto y tamaño de salida	80
Tamaño del batch	100
Tasa de aprendizaje y tasa de aprendizaje en decaimiento	0.002 y 0.1
Regularizador L2	1e-5
Pasos de propagación	1
Atención/No atención	True
Ratio de abandono	-
Número de cabezas (Atención Multinodal)	-
Tasa de decaimiento	3
Parámetros iniciales	0.11

**Tabla 3.3:** Resumen de los hiperparámetros inicial del modelo.

Seguidamente, en la Tabla 3.4 mostramos las posibles variaciones de cada hiperparámetro. Estas podemos dividir las en dos grandes grupos: por un lado aquellas que se deben multiplicar o dividir por algún término pero que mantienen el mismo orden de magnitud (por ejemplo los tamaños de salida o del batch, los pasos de propagación, el ratio de abandono,...). Por otro lado, están las que deben variar en algún orden de magnitud para poder notar sus efectos (como la tasa de aprendizaje, el regularizador L2 o los parámetros iniciales).

Nº	Hiperparámetros	Rango de variación
H1	Tamaño oculto y tamaño de salida	[64, 128, 256, 512]
H2	Tamaño del batch	[32, 64, 128, 256]
H3	Tasa de aprendizaje y tasa de aprendizaje en decaimiento	[0.001, 0.01, 0.1]
H4	Regularizador L2	[1e-5, 1e-4, 1e-3]
H5	Pasos de propagación	[1, 2, 3]
H6	Atención/No atención	[True,False]
H7	Ratio de abandono	[0.2, 0.5]
H8	Número de cabezas (Atención Multinodal)	[1, 2, 4]
H9	Tasa de decaimiento	[1,2,5]
H10	Parámetros iniciales	[0.1, 0.01, 0.001]

**Tabla 3.4:** Resumen de las modificaciones que se van a realizar en los hiperparámetros del modelo base.

### 3.3.3. Justificación de las épocas de entrenamiento

Por último, Wang et al. [14] aplican el modelo para un total de 30 épocas. Sin embargo, debido al gran número de experimentos que se deben realizar y a nuestras limitaciones en el entorno de trabajo (como se muestran en la Sección 3.1 de este capítulo), nuestro primer objetivo es encontrar una aproximación que permita reducir considerablemente el tiempo de ejecución total de todos los experimentos. En la Figuras 3.3 y 3.4 se muestran dos ejecuciones con el modelo plano, sin modificaciones. En ella, se han representado los resultados finales de Rango Recíproco Medio (MRR) y Recall (R) para los 20 primeros elementos recomendados.

Como se puede ver, tras un pequeño pico en los resultados (hasta la época 5), el cual asociamos a un sobreajuste inicial, a medida que avanzan las épocas, el modelo comienza rápidamente a converger y estabilizarse. Es decir, en las primeras épocas el modelo estaría aprendiendo rápidamente y ajustándose a los datos de entrenamiento, memorizando los datos de entrenamiento en lugar de aprender patrones generales. Sin embargo, durante las siguientes épocas el modelo se adapta más lentamente, y con ayuda de la regularización evita un sobreajuste excesivo.

Tras esta dinámica general en los resultados, hemos decidido aproximar el número total de épocas a 10 épocas, al considerar que es un número más que suficiente para encontrar unos resultados estables del modelo, y al mismo tiempo permite ahorrar mucho tiempo de ejecución en los experimentos. Gracias a esta aproximación conseguimos reducir cada ejecución del modelo a unas 4 horas.

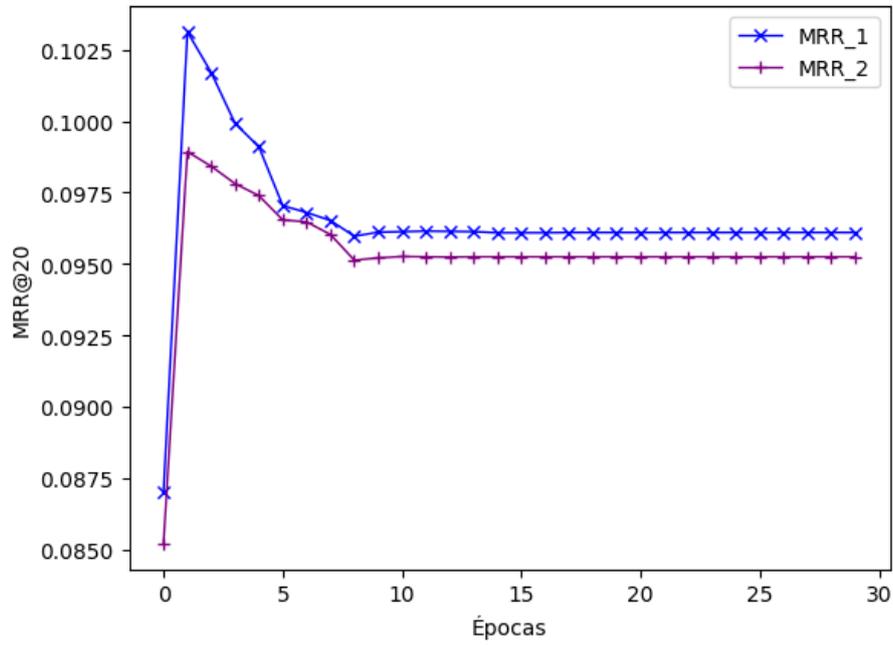


Figura 3.3: Representación gráfica de la métrica MRR para 30 épocas.

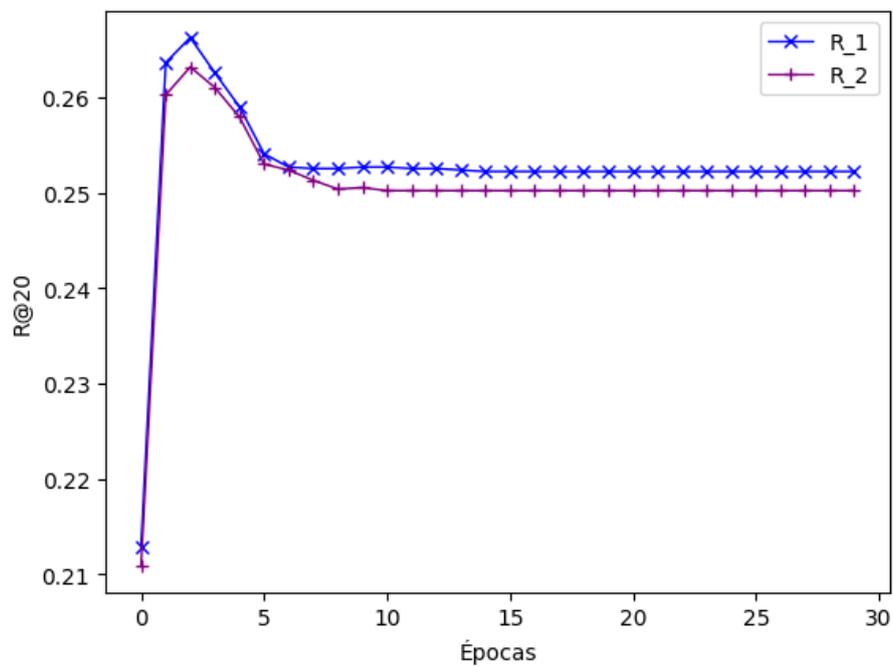


Figura 3.4: Representación gráfica de la métrica Recall para 30 épocas.

# EXPERIMENTOS

---

En este apartado comenzamos describiendo el conjunto de datos de los que partimos. Seguidamente, mostraremos los resultados obtenidos en cada uno de los experimentos, junto con una breve discusión de los resultados y la comprobación de cómo de acertadas han sido nuestras hipótesis previas a su implementación. Finalmente, unificaremos ambos experimentos mediante tres variantes del modelo, las cuales serán una combinación de una selección de modificaciones e hiperparámetros.

## 4.1. Conjunto de datos

Fundada en 2009, Foursquare se ha convertido en una de las redes sociales basadas en ubicación (LBSN) más destacadas, siendo ampliamente utilizada por investigadores [12, 13] para estudiar datos de localización. Los usuarios de la red social marcan sus visitas a lugares de interés (haciendo *check-in*), expresan preferencias mediante likes, dejan comentarios y reciben recomendaciones del sistema. Aunque la información de check-ins no está directamente disponible a través de la página web o API de Foursquare, muchos investigadores obtienen datos de interacciones a través de otras redes sociales como Twitter. Además, la propia red social implementa un sistema de puntos y recompensas que motiva a los usuarios a hacer check-in en lugares y contribuir activamente a la plataforma, donde los usuarios pueden recibir insignias y reconocimientos por sus contribuciones y actividades en ella.

En nuestro modelo, el conjunto de datos seleccionados de Foursquare proviene de [71]. En concreto, se eliminan los POI cuya frecuencia es inferior a cinco y conservamos las secuencias de comportamiento de los usuarios con más de un POI. El conjunto de datos final de Foursquare contiene 128.147 registros entre 1.809 usuarios y 5.514 POI.

Por otro lado, en el modelo de referencia [14] comparten los datos empleados en una carpeta de Google Drive <sup>1</sup> para los experimentos, por lo que, para poder obtener y comparar con los mismos resultados, empleamos los mismos datos para nuestras pruebas. Además, estos datos han pasado un preprocesamiento de datos que incluye una clasificación y limpieza de datos (desgraciadamente no aportan apenas información relacionada con esto en el trabajo original).

---

<sup>1</sup>[https://drive.google.com/drive/folders/1xf-dMzblfRTr3yE9SwdhR9vhWU4swhD\\_](https://drive.google.com/drive/folders/1xf-dMzblfRTr3yE9SwdhR9vhWU4swhD_)

## 4.2. Variaciones

Una vez hemos determinado las diferentes modificaciones a realizar en nuestro modelo, ejecutamos las 15 versiones planteadas en el capítulo anterior con el fin de obtener aquellas modificaciones que mejoren el rendimiento del modelo de referencia. Además, independientemente de si han resultado ser mejores recomendadores o no, posteriormente comprobamos cómo de cerca están los resultados de nuestras hipótesis iniciales.

La idea es, una vez realizados las pruebas, determinar 3 nuevas versiones adicionales que combinen varias de estas modificaciones, con lo cual podamos estudiar también el efecto combinado de los cambios. La Tabla 4.1 muestra la comparativa de los efectos de cada una de las modificaciones, para las dos métricas escogidas para nuestro análisis, Recall y MRR, para el ranking de los 5, 10 y 20 primeros puntos de interés. En concreto, se han resaltado en negrita aquellos resultados que son mejores a los de referencia.

Variante	Recall@			MRR@			T. (min)
	5	10	20	5	10	20	
Original	0,1505	0,1999	0,2496	0,0838	0,0904	0,0938	118
V1	0,1213	0,1536	0,1894	0,0801	0,0844	0,0869	119
V2	<b>0,1521</b>	0,1988	0,2447	<b>0,0853</b>	<b>0,0915</b>	<b>0,0947</b>	126
V3	0,1446	0,1934	0,2442	0,0827	0,0893	0,0928	87
V4	0,1456	0,1988	0,2437	<b>0,0838</b>	<b>0,0909</b>	<b>0,0940</b>	88
V5	0,0429	0,0441	0,0451	0,0364	0,0365	0,0366	147
V6	0,1085	0,1411	0,1680	0,0633	0,0677	0,0696	149
V7	0,0517	0,0543	0,0576	0,0416	0,0420	0,0422	88
V8	0,0441	0,0449	0,0462	0,0380	0,0381	0,0382	94
V9	0,0432	0,0445	0,0454	0,0356	0,0358	0,0359	87
V10	0,0374	0,0395	0,0420	0,0328	0,0331	0,0333	93
V11	0,0292	0,0321	0,0335	0,0270	0,0274	0,0275	122
V12	0,0002	0,0003	0,0009	0,0000	0,0001	0,0001	119
V13	0,1488	0,1940	0,2391	<b>0,0847</b>	<b>0,0908</b>	<b>0,0939</b>	118
V14	<b>0,1524</b>	<b>0,2007</b>	<b>0,2512</b>	<b>0,0859</b>	<b>0,0923</b>	<b>0,0959</b>	121
V15	<b>0,1514</b>	<b>0,1988</b>	0,2454	<b>0,0847</b>	<b>0,0910</b>	<b>0,0943</b>	118

**Tabla 4.1:** Comparación del rendimiento de cada una de las variantes en Recall y MRR para 10 épocas. Resaltamos en negrita las variantes cuyos resultados mejoran al original. Además, añadimos las medidas de tiempo de una ejecución.

Como se puede ver en la Tabla 4.1, para las 15 modificaciones propuestas, encontramos resultados tanto beneficiosos como perjudiciales para nuestro modelo base. Siendo estos, generalmente, igual de

buenos o malos en ambas métricas al mismo tiempo. Existe una peculiaridad, y es que el único caso donde no se cumple esto es para la celda RNN básica (V1), donde al aplicar una celda más básica conseguimos (en un tiempo algo menor), que el porcentaje de aciertos sea bajo a pesar de haber identificado bien a los  $k$  primeros elementos, seguramente debido a que al disminuir la complejidad del modelo, muchos elementos se hayan considerado erróneamente relevantes.

Además, observamos cómo la función de activación (V2-4) apenas tiene implicación en los resultados del modelo (únicamente destaca la función ReLU la cual reduce considerablemente el tiempo de ejecución), mejorando ligeramente los resultados de MRR pero no los de Recall. Los nuevos optimizadores (V5-9) y aplicar una capa de Dropout (V10) traen peores resultados, principalmente debido a las complejas relaciones que nuestro modelo tiene que captar o a que no existe sobreajuste y se acaba generalizando demasiado, respectivamente.

Los regularizadores (V11-13) confirman nuestra teoría, y es que el modelo está infraajustado, es decir, cualquier restricción adicional produce peores resultados. Esto explicaría por qué las variantes V11 y V12 empeoran el recomendador, mientras que quitar el regularizador (V13) no. Por último, concluimos que los nuevos mecanismos de atención (V14-15) no solo captan correctamente las relaciones si no que incluso mejoran en cierta medida el rendimiento.

Tras esto, se determinan tres nuevas versiones del modelo. El objetivo es estudiar el impacto de varias de las modificaciones conjuntamente. Concretamente, estas versiones han sido creadas para responder a dos preguntas:

- **P1:** ¿Cuál es el efecto combinado si modificamos todas las componentes seleccionadas del modelo?
- **P2:** ¿Qué resultados obtendríamos si escogiéramos únicamente aquellas modificaciones cuyos resultados mejoran o igualan los del modelo base?

Para responderlas, se crean tres nuevos modelos cuyas combinaciones están recogidas en la Tabla 4.2. La primera de ellas responde a la primera pregunta, ya que hemos cambiado todas las componentes posibles. A priori, debido a que en la Tabla 4.1 no se han obtenido buenos resultados en todas las componentes, suponemos que los resultados pueden llegar a ser bastante peores a los del modelo original. Las dos siguientes son una mezcla de las nuevas variantes en función de los resultados anteriores (Tabla 4.1). Los resultados obtenidos de estas nuevas pruebas se incluyen en la Tabla 4.3.

Como intuimos, el primer modelo V16, el cual es una mezcla de todas las posibles variantes, obtiene el peor de los resultados. Seguramente debido a que la contribución global suele ser una combinación de todas las aportaciones individuales, y al considerar todas las opciones, estamos incluyendo variantes menos acertadas.

También observamos cómo los otros dos modelos, a priori seleccionados a mano con las modificaciones con mejores resultados, no llegan a superar el rendimiento del modelo original. En este caso,

Componente	Modelo original	Versión 16	Versión 17	Versión 18
Red neuronal de grafos (GGNN)	Celda GRU	Celda RNN básica	Celda GRU	Celda GRU
Función de activación	F. Softmax	F. Sigmoid	F. Sigmoid	F. Softmax
Optimizador	ADAM	RMSProp	ADAM	ADAM
Dropout	No Aplica	1 capa (0.3)	No Aplica	1 capa (0.3)
Regularizador	L2	Sin L2	L2	L2
Mecanismos de atención	2 niveles	ANN	ANN	Multinodal

**Tabla 4.2:** Resumen de las componentes de las nuevas versiones propuestas.

Variante	Recall@			MRR@		
	5	10	20	5	10	20
Original	0,1505	0,1999	0,2496	0,0838	0,0904	0,0938
V16	0,0565	0,0738	0,0912	0,0340	0,0364	0,0376
V17	0,1480	0,1982	0,2484	0,0830	0,0898	0,0933
V18	0,1380	0,1851	0,2312	0,0799	0,0862	0,0894

**Tabla 4.3:** Comparación del rendimiento de las tres versiones propuestas en Recall y MRR para 10 épocas.

entendemos que el modelo final está asimismo influenciado por la afinidad entre todas las componentes, ya que a pesar de no ser mejores resultados, obtenemos unos muy parecidos a los del original. No obstante, una posible razón para estos malos resultados podría ser que los modelos no están usando los mejores hiperparámetros, por ello, en la siguiente sección realizamos un estudio donde se varían los valores de los hiperparámetros en las ejecuciones.

### 4.3. Estudio de hiperparámetros

Adicionalmente, para nuestro segundo conjunto de pruebas, realizamos un análisis de 10 de los principales parámetros de nuestro modelo. En la Tabla 4.4 mostramos la comparación de los resultados obtenidos para las métricas anteriores, MRR y R, y los 5, 10 y 20 primeros POIs, cambiando los hiperparámetros según lo indicado en la Tabla 3.4.

Como se puede observar, aparecen todas las combinaciones propuestas, incluidas aquellas que han provocado errores en el modelo. Por ejemplo, para los hiperparámetros H1 y H2, los valores de 512 y 256, respectivamente, daban problemas debido a las enormes dimensiones de la red neuronal resultante. Esto último, sumado a nuestros limitados recursos (expuestos en la Tabla 3.1) ha hecho que el modelo fallase. Además, el parámetro H6 no se ha podido analizar debido a fallos en la definición del modelo; mientras que el H8, debido a que este cambio únicamente afecta a la variante del modelo que aplica el mecanismo de atención de múltiples cabezas, no al modelo base, el cual mantiene el mecanismo de atención de dos niveles, tampoco se ha probado con el modelo inicial.

Variante	Recall@			MRR@		
	5	10	20	5	10	20
Original	0,1505	0,1999	0,2496	0,0838	0,0904	0,0938
Variación H1: 64	0,1490	<b>0,2001</b>	<b>0,2505</b>	0,0823	0,0892	0,0927
Variación H1: 128	0,1416	0,1923	0,2437	0,0836	<b>0,0905</b>	<b>0,0941</b>
Variación H1: 256	0,1349	0,1781	0,2280	0,0837	0,0895	0,0929
Variación H1: 512	X	X	X	X	X	X
Variación H2: 32	0,1405	0,1866	0,2357	<b>0,0838</b>	0,0900	0,0934
Variación H2: 64	0,1471	0,1996	0,2458	<b>0,0841</b>	<b>0,0911</b>	<b>0,0943</b>
Variación H2: 128	0,1463	0,1994	<b>0,2513</b>	<b>0,0856</b>	<b>0,0927</b>	<b>0,0964</b>
Variación H2: 256	X	X	X	X	X	X
Variación H3: 0.001	<b>0,1514</b>	0,1971	0,2444	<b>0,0848</b>	<b>0,0909</b>	<b>0,0942</b>
Variación H3: 0.01	0,1477	0,1973	0,2459	<b>0,0888</b>	<b>0,0954</b>	<b>0,0987</b>
Variación H3: 0.1	0,0079	0,0091	0,0117	0,0057	0,0059	0,0060
Variación H4: 1e-4	<b>0,1541</b>	<b>0,2035</b>	<b>0,2555</b>	<b>0,0879</b>	<b>0,0945</b>	<b>0,0981</b>
Variación H4: 1e-3	0,1408	0,1879	0,2346	0,0813	0,0875	0,0908
Variación H5: 2	0,0923	0,1111	0,1321	0,0656	0,0681	0,0695
Variación H5: 3	0,1289	0,1701	0,2150	0,0776	0,0831	0,0863
Variación H6: False	X	X	X	X	X	X
Variación H7: 0.2	0,1440	0,1920	0,2385	0,0824	0,0889	0,0922
Variación H7: 0.4	0,1405	0,1808	0,2278	0,0802	0,0856	0,0888
Variación H7: 0.5	0,1309	0,1697	0,2104	0,0767	0,0818	0,0846
Variación H8: 1,2,3	X	X	X	X	X	X
Variación H9: 1	0,1491	0,1951	0,2379	<b>0,0855</b>	<b>0,0916</b>	<b>0,0945</b>
Variación H9: 2	<b>0,1579</b>	<b>0,2115</b>	<b>0,2572</b>	<b>0,0878</b>	<b>0,0950</b>	<b>0,0982</b>
Variación H9: 5	0,1311	0,1670	0,2126	0,0805	0,0854	0,0885
Variación H10: 0.01	<b>0,1551</b>	<b>0,2056</b>	<b>0,2572</b>	<b>0,0875</b>	<b>0,0943</b>	<b>0,0979</b>
Variación H10: 0.001	<b>0,1524</b>	<b>0,2045</b>	<b>0,2536</b>	<b>0,0871</b>	<b>0,0939</b>	<b>0,0974</b>

**Tabla 4.4:** Comparación del rendimiento de los nuevos valores de hiperparámetros en Recall y MRR para 10 épocas. En negrita las variantes cuyos resultados mejoran al original.

En primer lugar, notamos cómo los cambios en el estudio de hiperparámetros son bastante menores con respecto al modelo original, sin embargo, muchos más resultados se encuentran por encima de los de referencia. Por un lado, al aumentar el tamaño de la red y de nuestros grupos de entrenamiento (H1-2), estamos consiguiendo un aumento de la complejidad, mejorando el rendimiento del modelo. La tasa de aprendizaje (H3), por otro lado, confirma una buena elección del mismo, ya que una tasa algo mayor o algo menor mejoran muy ligeramente el MRR, pero el Recall lo mantenemos por encima.

Además, existen varios parámetros donde un valor intermedio podría mejorar las recomendaciones. Estos son el regularizador, la tasa de decaimiento y la modificación de los parámetros iniciales (H4,9 y 10). Estos valores intermedios afinan el modelo y aumentan la velocidad de convergencia. Como puntualización, los mejores resultados del análisis de hiperparámetros se han obtenido para estas variaciones (Variación H9, para una tasa de decaimiento de 2, y Variación H10, para los parámetros iniciales de 0.01). Finalmente, concluir que añadir una capa de dropout (H5) o variar los pasos de propagación de la red neuronal (H7), no mejora el rendimiento, debido seguramente a una disminución de la capacidad del modelo para aprender.

A continuación, siguiendo la dinámica del anterior conjunto de pruebas, realizamos un breve estudio del efecto conjunto de cambiar varios hiperparámetros al mismo tiempo en el modelo (Tabla 4.5). Para ello, determinamos dos posibles combinaciones: la primera de ellas incluye el mejor valor para cada uno de las posibles variables estudiadas, es decir, Modelo(H1=64, H2=128, H3=0.01, H4=1e-4, H5=3, H7=0.2, H9=2, H10=0.01), y al que llamaremos Modelo Opt. 1. La segunda combinación aplica únicamente aquellas modificaciones que generan resultados mejores o por lo menos similares a los de referencia, esto es, Modelo(H1=64, H2=128, H4=1e-4, H9=2, H10=0.01), llamado Modelo Opt. 2.

Variante	Recall@			MRR@		
	5	10	20	5	10	20
Original	0,1505	0,1999	0,2496	0,0838	0,0904	0,0938
Modelo Opt 1	0,1420	0,1874	0,2320	0,0835	0,0896	0,0927
Modelo Opt 2	<b>0,1605</b>	<b>0,2090</b>	<b>0,2575</b>	<b>0,0909</b>	<b>0,0973</b>	<b>0,1007</b>

**Tabla 4.5:** Comparación del rendimiento para la combinación de hiperparámetros en Recall y MRR para 10 épocas. Resaltamos en negrita las variantes cuyos resultados mejoran al original.

Para el primero de los modelos ocurre algo parecido a lo que sucedía en la Variante 16 del experimento anterior (Sección 4.2). Esto es, que la combinación de varias modificaciones produce un efecto combinado, el cual podría considerarse, bajo algunos matices, como una media de todos ellos. Es por eso que los resultados en este experimento no son malos, pero tampoco destacan entre los obtenidos hasta ahora.

Sin embargo, al seleccionar únicamente los cambios que mejoran el recomendador, hemos obtenido los mejores resultados de todos los experimentos. Cabe destacar que las métricas aumentan por encima de cualquier otro modelo ejecutado, por lo que podemos confirmar, que en estos casos, no solo se tiene en cuenta el efecto individual de cada modificación, si no que esta combinación produce

un efecto agregado, el cual mejora todavía más los resultados.

## 4.4. Resultados

Finalmente, concluimos nuestros experimentos con la unión de ambas pruebas. Esto es, seleccionamos la mejor combinación de hiperparámetros, aquellos que igualaban o mejoraban el rendimiento de nuestro recomendador (Modelo(H1=64, H2=128, H4=1e-4, H9=2, H10=0.01) de la Sección 4.3), junto con las tres versiones finales obtenidas para las modificaciones del modelo base (Sección 4.2). Los resultados se muestran en la Tabla 4.6, para las métricas ya mencionadas de MRR y Recall en diferentes rankings de POI.

Variante	Recall@			MRR@		
	5	10	20	5	10	20
Original	0,1505	0,1999	0,2496	0,0838	0,0904	0,0938
Versión 16	0,0395	0,0562	0,0766	0,0235	0,0257	0,0271
Versión 17	<b>0,1524</b>	<b>0,2042</b>	<b>0,2542</b>	<b>0,0877</b>	<b>0,0947</b>	<b>0,0981</b>
Versión 18	0,1379	0,1829	0,2320	0,0792	0,0853	0,0887

**Tabla 4.6:** Comparación del rendimiento de la unión de ambos experimentos en Recall y MRR para 10 épocas. Resaltamos en negrita las variantes cuyos resultados mejoran al original.

En primer lugar, la Versión 16 sigue bastante por debajo en cuanto a métricas con respecto de las otras dos. Esto es debido a que, a pesar de tener una buena combinación de hiperparámetros, la selección de componentes no permite que el sistema de recomendación funcione correctamente.

Por otro lado, las versiones 17 y 18 tienen resultados mucho más prometedores, destacando la Versión 17 como la mejor de las tres. Esto es debido a la selección previa de los componentes del sistema. Como nota, la capa de Dropout que incluimos en la Versión 18 perjudicaba fuertemente el modelo con esta componente individual, y es seguramente el motivo por el cual esta versión obtiene resultados peores.

Como conclusión final, comentar que, a pesar de haber seleccionado la mejor combinación de componentes del modelo y el mejor conjunto de hiperparámetros, seguimos obteniendo recomendadores peores a los obtenidos en el análisis de hiperparámetros. Por ende, comprobamos cómo, por muy bien seleccionados que estén los parámetros del modelo, estos son solo óptimos para un modelo concreto. Por consiguiente, concluimos que se podría mejorar la Versión 17 aún más mediante un nuevo estudio de hiperparámetros.



# CONCLUSIÓN Y TRABAJO FUTURO

---

En este capítulo se expone un resumen de todos los resultados y conclusiones que se han obtenido y se proponen una serie de ideas que podrían plantearse como futuras investigaciones en base a este trabajo.

## 5.1. Conclusión

En este trabajo, abordamos el desafío de la recomendación personalizada, específicamente en el ámbito turístico, proponiendo un modelo basado en red neuronal. Nuestra propuesta se centra en mejorar de manera integral un modelo de Redes Neuronales basadas en Grafos (GNN) en el contexto de recomendación del siguiente punto de interés (NEXT POI). Para ello, realizamos un estudio del estado del arte en este área, con el fin de seleccionar en primer lugar un modelo del estado del arte actual que sea lo más completo posible. El artículo elegido [14] destaca, no solo por usar las redes neuronales de grafos mencionadas, si no por aplicar diferentes mecanismos de atención, los cuales le permiten capturar las preferencias a corto y largo plazo, y captar adecuadamente relaciones complejas en el sistema de POI-usuarios.

A través de dos experimentos, el primero orientado al estudio de las componentes principales del modelo de referencia y el segundo hacia un análisis de hiperparámetros, conseguimos, por un lado, entender completamente nuestro complejo modelo de partida, y por otro lado, presentar nuevas versiones del mismo, capaces de superar algunas limitaciones observadas y optimizar la capacidad predictiva del modelo.

Además, este trabajo ha contribuido al avance del conocimiento en el estado del arte de los algoritmos de aprendizaje profundo aplicado a la recomendación de próximos puntos de interés. La revisión exhaustiva del panorama actual en este objetivo, centrándose específicamente en los modelos de GNN, ha permitido identificar las tendencias emergentes, los desafíos actuales y las posibles direcciones futuras. Este análisis también es de especial importancia, puesto que no se han encontrado revisiones previas enfocadas en estos algoritmos basados en grafos bastante innovadores. Nuestro estado del arte proporciona un marco sólido para la comprensión del contexto en el que se sitúa el

modelo desarrollado y destaca la importancia de la innovación continua en este campo dinámico.

Por último, el trabajo demuestra la relevancia y el potencial de las Redes Neuronales de Grafos en la mejora de los sistemas de recomendación de POI. A medida que la tecnología avanza, se espera que las contribuciones de este estudio impulsen aún más el desarrollo de modelos más precisos y eficientes, proporcionando soluciones cada vez más personalizadas y relevantes para los usuarios en el ámbito de la recomendación de puntos de interés.

## 5.2. Trabajo futuro

Con perspectiva al futuro, podríamos ampliar estos experimentos y extenderlos con otros conjuntos de datos, como Yelp o Brightkite y ver cuáles son las características más determinantes a la hora de trabajar con ellos. Cada dataset presenta diferentes desafíos y oportunidades y su estudio conjunto proporcionaría una comprensión más completa de la capacidad del modelo en diferentes escenarios, lo que permitiría una evaluación más sólida de su rendimiento en diversas situaciones, asegurando su aplicabilidad en entornos del mundo real. Además, ampliar los experimentos a varios conjuntos de datos permitiría ampliar la validez de las hipótesis incluidas en este trabajo al utilizar datos diversos y heterogéneos.

Otras investigaciones por desarrollar serían las que incluyen el análisis completo de todas las combinaciones de versiones e hiperparámetros, en concreto, una validación cruzada podría encontrar nuevas versiones con mejores recomendaciones. No obstante, hay que tener en cuenta el coste de estos experimentos, por lo que sería interesante apostar por variantes que, además de efectivas, sean eficientes.

Además, para enriquecer aún más el poder predictivo del recomendador de POI basado en GNN, sugerimos la incorporación de fuentes de datos adicionales con una variedad de tipos de información. Dado que las Redes Neuronales de Grafos tienen la capacidad de explotar las relaciones sociales y otros atributos, la inclusión de datos que abarquen aspectos como preferencias sociales, historial de interacciones y comentarios de usuarios puede ser fundamental. Integrar estos elementos en el modelo puede proporcionar una representación más completa y contextualizada de las preferencias individuales, mejorando la capacidad del sistema para ofrecer recomendaciones más personalizadas y relevantes.

Finalmente, queremos destacar que la investigación en este campo prometedor está en constante evolución y requiere un esfuerzo continuo para avanzar en el diseño de recomendadores de POI basados en GNN. Abordar estos aspectos propuestos no solo contribuirá a mejorar los resultados de los recomendadores, sino que también abrirá nuevas oportunidades para innovaciones en la recomendación personalizada en el ámbito de los puntos de interés.

# BIBLIOGRAFÍA

---

- [1] H. A. Rahmani, M. Aliannejadi, M. Baratchi, and F. Crestani, "Joint geographical and temporal modeling based on matrix factorization for point-of-interest recommendation," in *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I* (J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, and F. Martins, eds.), vol. 12035 of *Lecture Notes in Computer Science*, pp. 205–219, Springer, 2020.
- [2] Z. Pan, L. Cui, X. Wu, Z. Zhang, X. Li, and G. Chen, "Deep potential geo-social relationship mining for point-of-interest recommendation," *IEEE Access*, vol. 7, pp. 99496–99507, 2019.
- [3] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014* (J. Li, X. S. Wang, M. N. Garofalakis, I. Soboroff, T. Suel, and M. Wang, eds.), pp. 739–748, ACM, 2014.
- [4] Y. Wen, J. Zhang, Q. Zeng, X. Chen, and F. Zhang, "Loc2vec-based cluster-level transition behavior mining for successive POI recommendation," *IEEE Access*, vol. 7, pp. 109311–109319, 2019.
- [5] R. Mao, Z. Han, Z. Liu, Y. Liu, X. Lv, and P. Xuan, "PRME-GTS: A new successive POI recommendation model with temporal and social influences," in *Advanced Data Mining and Applications - 15th International Conference, ADMA 2019, Dalian, China, November 21-23, 2019, Proceedings* (J. Li, S. Wang, S. Qin, X. Li, and S. Wang, eds.), vol. 11888 of *Lecture Notes in Computer Science*, pp. 266–274, Springer, 2019.
- [6] X. Xiong, S. Qiao, Y. Li, N. Han, G. Yuan, and Y. Zhang, "A point-of-interest suggestion algorithm in multi-source geo-social networks," *Eng. Appl. Artif. Intell.*, vol. 88, 2020.
- [7] R. Yue-Qiang, W. Ze, S. Xiao-Na, and S. Shi-Min, "A multi-element hybrid location recommendation algorithm for location based social networks," *IEEE Access*, vol. 7, pp. 100416–100427, 2019.
- [8] Q. Cui, Y. Tang, S. Wu, and L. Wang, "Distance2pre: Personalized spatial preference for next point-of-interest prediction," in *Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part III* (Q. Yang, Z. Zhou, Z. Gong, M. Zhang, and S. Huang, eds.), vol. 11441 of *Lecture Notes in Computer Science*, pp. 289–301, Springer, 2019.
- [9] J. He, X. Li, and L. Liao, "Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking," pp. 1837–1843, 08 2017.
- [10] J. He, X. Li, L. Liao, and M. Wang, "Inferring continuous latent preference on transition intervals for next point-of-interest recommendation," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part II* (M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Iffrim, eds.), vol. 11052 of *Lecture Notes in Computer Science*, pp. 741–756, Springer, 2018.
- [11] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "STELLAR: spatial-temporal latent ranking for successive point-of-interest recommendation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA* (D. Schuurmans and M. P. Wellman, eds.), pp. 315–322, AAAI Press, 2016.
- [12] P. Sánchez and A. Bellogín, "Point-of-interest recommender systems based on location-based social networks: A survey from an experimental perspective," *ACM Comput. Surv.*, vol. 54, no. 11s, pp. 223:1–223:37, 2022.
- [13] M. A. Islam, M. M. Mohammad, S. S. S. Das, and M. E. Ali, "A survey on deep learning based point-of-interest (POI) recommendations," *Neurocomputing*, vol. 472, pp. 306–325, 2022.
- [14] D. Wang, X. Wang, Z. Xiang, D. Yu, S. Deng, and G. Xu, "Attentive sequential model based on graph neural network for next poi recommendation," *World Wide Web*, vol. 24, no. 6, pp. 2161–2184, 2021.
- [15] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA* (D. Schuurmans and M. P. Wellman, eds.), pp. 194–200, AAAI Press, 2016.

- [16] N. Lim, B. Hooi, S. Ng, Y. L. Goh, R. Weng, and R. Tan, "Hierarchical multi-task graph recurrent network for next POI recommendation," in *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022* (E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, eds.), pp. 1133–1143, ACM, 2022.
- [17] S. Yang, J. Liu, and K. Zhao, "Getnext: Trajectory flow map enhanced transformer for next POI recommendation," in *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022* (E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, eds.), pp. 1144–1153, ACM, 2022.
- [18] J. Manotumruksa, C. Macdonald, and I. Ounis, "A contextual attention recurrent architecture for context-aware venue recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, (New York, NY, USA), p. 555–564*, Association for Computing Machinery, 2018.
- [19] S. M. M. Rashid, M. E. Ali, and M. A. Cheema, "Deepaltrip: Top-k alternative itineraries for trip recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9433–9447, 2023.
- [20] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, and H. Yin, "Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 214–221, AAAI Press, 2020.
- [21] Z. Li, W. Cheng, H. Xiao, W. Yu, H. Chen, and W. Wang, "You are what and where you are: Graph enhanced attention network for explainable POI recommendation," in *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021* (G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, eds.), pp. 3945–3954, ACM, 2021.
- [22] S. Dai, Y. Yu, H. Fan, and J. Dong, "Spatio-temporal representation learning with social tie for personalized poi recommendation," *Data Science and Engineering*, vol. 7, 03 2022.
- [23] Y. Qin, Y. Wang, F. Sun, W. Ju, X. Hou, Z. Wang, J. Cheng, J. Lei, and M. Zhang, "Disenpoi: Disentangling sequential and geographical influence for point-of-interest recommendation," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM 2023, Singapore, 27 February 2023 - 3 March 2023* (T. Chua, H. W. Lauw, L. Si, E. Terzi, and P. Tsaparas, eds.), pp. 508–516, ACM, 2023.
- [24] S. Zhang, Z. Bai, P. Li, and Y. Chang, "Multi-graph convolutional network for fine-grained and personalized poi recommendation," *Electronics*, vol. 11, p. 2966, 09 2022.
- [25] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, *Semantics-Aware Content-Based Recommender Systems*, pp. 119–159. 01 2015.
- [26] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval the Concepts and Technology Behind Search*. 01 2011.
- [27] I. Cantador, A. Bellogín, and D. Vallet, "Content-based recommendation in social tagging systems," in *Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, (New York, NY, USA), p. 237–240*, Association for Computing Machinery, 2010.
- [28] X. Ren, M. Song, H. E, and J. Song, "Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation," *Neurocomput.*, vol. 241, p. 38–55, jun 2017.
- [29] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A framework and graphical development environment for robust NLP tools and applications.," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 2002*.
- [30] I. Tobías, "Matrix factorization models for cross-domain recommendation: Addressing the cold start in collaborative filtering," 2017.
- [31] R. Burke, *Hybrid Web Recommender Systems*, pp. 377–408. 01 2007.
- [32] Z. Ding, X. Li, C. Jiang, and M. Zhou, "Objectives and state-of-the-art of location-based social network recommender systems," *ACM Computing Surveys*, vol. 51, pp. 1–28, 01 2018.
- [33] D. Yang, D. Zhang, and B. Qu, "Participatory cultural mapping based on collective behavior data in location-based social networks," *ACM Trans. Intell. Syst. Technol.*, vol. 7, jan 2016.

- [34] R. S. G. Hitti, "Explotación de características de secuencias para su uso en sistemas de recomendación," 2020.
- [35] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016* (A. Karatzoglou, B. Hidasi, D. Tikk, O. S. Shalom, H. Roitman, B. Shapira, and L. Rokach, eds.), pp. 17–22, ACM, 2016.
- [36] Y. Blanco-Fernández, J. Pazos-Arias, A. Gil, M. Cabrer, and M. López-Nores, "Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems," *Consumer Electronics, IEEE Transactions on*, vol. 54, pp. 727 – 735, 06 2008.
- [37] Z. Jia, Y. Yang, W. Gao, and X. Chen, "User-based collaborative filtering for tourist attraction recommendations," *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015*, pp. 22–25, 04 2015.
- [38] G. Linden, B. Smith, and J. York, "Linden g, smith b and york j: 'amazon.com recommendations: item-to-item collaborative filtering', internet comput. ieee, , 7," *Internet Computing, IEEE*, vol. 7, pp. 76 – 80, 02 2003.
- [39] C. Cheng, H. Yang, M. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 2605–2611, 01 2013.
- [40] J.-D. Zhang, C.-Y. Chow, and Y. Li, "Lore: exploiting sequential influence for location recommendations," pp. 103–112, 11 2014.
- [41] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2014.
- [42] L. Huang, Y. Ma, Y. Liu, and A. Kumar, "Multi-modal bayesian embedding for point-of-interest recommendation on location-based cyber-physical-social networks," *Future Generation Computer Systems*, vol. 108, pp. 1119–1128, 07 2020.
- [43] W. Wang, J. Chen, J. Wang, J.-x. Chen, and Z. Gong, "Geography-aware inductive matrix completion for personalized point of interest recommendation in smart cities," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 10 2019.
- [44] Q. Gao, W. Wang, K. Zhang, X. Yang, C. Miao, and T. Li, "Self-supervised representation learning for trip recommendation," *Knowl. Based Syst.*, vol. 247, p. 108791, 2022.
- [45] F. Zhou, R. Yin, K. Zhang, G. Trajcevski, T. Zhong, and J. Wu, "Adversarial point-of-interest recommendation," in *The World Wide Web Conference, WWW '19, (New York, NY, USA)*, p. 3462–34618, Association for Computing Machinery, 2019.
- [46] P. Zhao, A. Luo, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou, "Where to go next: A spatio-temporal gated network for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2512–2524, 2022.
- [47] B. Chang, Y. Park, D. Park, S. Kim, and J. Kang, "Content-aware hierarchical point-of-interest embedding model for successive poi recommendation," in *International Joint Conference on Artificial Intelligence*, 2018.
- [48] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: Predicting human mobility with attentional recurrent networks," pp. 1459–1468, 04 2018.
- [49] Y. Luo, Q. Liu, and Z. Liu, "Stan: Spatio-temporal attention network for next location recommendation," pp. 2177–2185, 04 2021.
- [50] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 5:1–5:38, 2019.
- [51] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, (New York, NY, USA)*, p. 165–174, Association for Computing Machinery, 2019.
- [52] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 11 2015.
- [53] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 346–353, AAAI Press, 2019.
- [54] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic

- graph attention networks,” in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019* (J. S. Culpepper, A. Moffat, P. N. Bennett, and K. Lerman, eds.), pp. 555–563, ACM, 2019.
- [55] X. Wang, X. He, Y. Cao, M. Liu, and T. Chua, “KGAT: knowledge graph attention network for recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019* (A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, eds.), pp. 950–958, ACM, 2019.
- [56] F. Wang, Y. Qu, L. Zheng, C.-T. Lu, and P. Yu, “Deep and broad learning on content-aware poi recommendation,” pp. 369–378, 10 2017.
- [57] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, “What your images reveal: Exploiting visual contents for point-of-interest recommendation,” in *Proceedings of the 26th International Conference on World Wide Web, WWW '17, (Republic and Canton of Geneva, CHE), p. 391–400, International World Wide Web Conferences Steering Committee, 2017.*
- [58] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, “Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, (New York, NY, USA), p. 1245–1254, Association for Computing Machinery, 2017.*
- [59] J. Manotumruksa, C. Macdonald, and I. Ounis, “A deep recurrent collaborative filtering framework for venue recommendation,” *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017.*
- [60] R. Ding and Z. Chen, “Recnet: a deep neural network for personalized poi recommendation in location-based social networks,” *International Journal of Geographical Information Science*, vol. 32, pp. 1–18, 04 2018.
- [61] C. Ma, Y. Zhang, Q. Wang, and X. Liu, “Point-of-interest recommendation: Exploiting self-attentive autoencoders with neighbor-aware influence,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018* (A. Cuzzocrea, J. Allan, N. W. Paton, D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, and H. Wang, eds.), pp. 697–706, ACM, 2018.
- [62] R. Li, Y. Shen, and Y. Zhu, “Next point-of-interest recommendation with temporal and multi-level context attention,” pp. 1110–1115, 11 2018.
- [63] O. Tal and Y. Liu, “A joint deep recommendation framework for location-based social networks,” *Complexity*, vol. 2019, pp. 1–11, 03 2019.
- [64] D. Lian, Y. Wu, Y. Ge, X. Xie, and E. Chen, “Geography-aware sequential location recommendation,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, (New York, NY, USA), p. 2009–2019, Association for Computing Machinery, 2020.*
- [65] L. Huang, Y. Ma, Y. Liu, and K. He, “DAN-SNR: A deep attentive network for social-aware next point-of-interest recommendation,” *ACM Trans. Internet Techn.*, vol. 21, no. 1, pp. 2:1–2:27, 2021.
- [66] J. Jeon, S. Kang, M. Jo, S. Cho, N. Park, S. Kim, and C. Song, “Lightmove: A lightweight next-poi recommendation fortaxicab rooftop advertising,” in *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021* (G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, and H. Tong, eds.), pp. 3857–3866, ACM, 2021.
- [67] L. Huang, Y. Ma, S. Wang, and Y. Liu, “An attention-based spatiotemporal lstm network for next poi recommendation,” *IEEE Transactions on Services Computing*, vol. 14, pp. 1585–1597, 11 2021.
- [68] B. Liu, Q. Meng, H. Zhang, K. Xu, and J. Cao, “Vgmf: Visual contents and geographical influence enhanced point-of-interest recommendation in location-based social network,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, 06 2022.
- [69] G. Chen, G. Zhao, L. Zhu, Z. Zhuo, and X. Qian, “Combining non-sampling and self-attention for sequential recommendation,” *Inf. Process. Manage.*, vol. 59, mar 2022.
- [70] W. L. Hamilton, “Graph representation learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159.
- [71] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, “Time-aware point-of-interest recommendation,” in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 363–372, 2013.



