

---

Análisis de Conceptos Formales en el área de  
recomendación. Aplicación experimental basada  
en RecSys 2019

---



**Trabajo Fin de Máster**

**Carlos Ismael Ortega Sánchez**

Trabajo de investigación para el

Máster en Lenguajes y Sistemas Informáticos

Universidad Nacional de Educación a Distancia

Dirigido por el

**Dr. Juan Manuel Cigarrán Recuero**

Junio 2023



# Resumen

Con el auge de internet y la creciente cantidad de datos disponibles, los sistemas de recomendación se han convertido en una herramienta esencial para ayudar a los usuarios a encontrar información relevante. En particular, la explosión de contenido generado por los usuarios ha aumentado la necesidad de sistemas de recomendación que puedan manejar grandes cantidades de datos y proporcionar recomendaciones personalizadas.

En los últimos años, ha surgido una tendencia en la utilización de sistemas basados en Aprendizaje Profundo para abordar una amplia gama de problemas. Si bien estos sistemas han demostrado ser altamente efectivos en diversas tareas, también presentan desafíos en términos de su interpretación y predictibilidad. Esta falta de transparencia y comprensión en los modelos de Aprendizaje Profundo plantea preocupaciones significativas en aplicaciones críticas, como la atención médica, la justicia penal o la toma de decisiones financieras. La opacidad de estos modelos dificulta la confianza y la aceptación de los resultados, lo que puede limitar su utilidad en determinados contextos.

Una posible alternativa puede ser la teoría del Análisis de Conceptos Formales (FCA) que permite la organización automática de distintos datos en una estructura llamada retículo con relaciones generalización-especificación. Esta representación revela relaciones ocultas que pueden escapar a la percepción humana y puede ser útil para mejorar la precisión y la interpretación de los sistemas de recomendación. Al proporcionar una representación más entendible y navegable de la información, el enfoque del FCA puede ayudar a los usuarios a comprender mejor las recomendaciones y a confiar en los resultados.

El presente trabajo trata de analizar la utilidad del FCA en el contexto de la recomendación basándose en la convención de RecSys 2019. Para ello, primero se ha tenido que explorar la creación del retículo y su estudio.

Referente a esto, se proponen distintas aproximaciones de posibles usos que se pueden dar al retículo de conceptos formales generados para así poder realizar una recomendación. Posteriormente se han realizado distintas aproximaciones para la extracción de la información representada en el retículo y así poder usarla en la recomendación. Por ultimo, se han comparado las distintas aproximaciones entre sí para comprobar su utilidad.

Los resultados obtenidos en este experimento demostraron la utilidad de la FCA para mejorar los sistemas de recomendación, obteniéndose una mejoría de un 20.12% en una de las propuestas. Esto favorece la idea de que la FCA puede ser una teoría útil y efectiva en el campo de la recomendación.

# Abstract

With the rise of the internet and the increasing amount of data available, recommender systems have become an essential tool to help users find relevant information. In particular, the explosion of user-generated content has increased the need for recommender systems that can handle large amounts of data and provide personalised recommendations.

In recent years, a trend has emerged in the use of deep learning-based systems to address a wide range of problems. While these systems have proven to be highly effective in a variety of tasks, they also present challenges in terms of their interpretability and predictability. This lack of transparency and understanding in deep learning models raises significant concerns in critical applications such as healthcare, criminal justice or financial decision-making. The opacity of these models makes it difficult to trust and accept the results, which may limit their usefulness in certain contexts.

A possible alternative may be the theory of Formal concept analysis (FCA), which allows the automatic organisation of different data into a structure called a lattice with generalisation-specificity relationships. This representation reveals hidden relationships that may escape human perception and can be useful for improving the accuracy and interpretability of recommender systems. By providing a more understandable and navigable representation of information, the FCA approach can help users to better understand recommendations and trust the results.

This paper attempts to analyse the usefulness of FCA in the context of recommendation based on the RecSys 2019 convention. In order to do so, first the creation of the lattice and its study had to be explored, and with reference to this, we offer different approaches of possible uses that can be given to the lattice of formal concepts generated in order to be able to make a recommendation are proposed. Subsequently, different approaches have been used to extract the information represented in the lattice in order to be able

to use it in the recommendation. Finally, the different approaches have been compared with each other to check their usefulness.

The results obtained in this experiment demonstrated the usefulness of the FCA to improve the recommendation systems, obtaining an improvement of 20.12 % in one of the proposals. This supports the idea that FCA can be a useful and effective theory in the field of recommendation.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	2
1.2. Propuesta y objetivos . . . . .	5
<b>2. Estado del arte</b>	<b>9</b>
2.1. Representación basada en conocimiento . . . . .	9
2.2. Detección de temas . . . . .	12
2.2.1. Clasificación . . . . .	12
2.2.2. Probabilístico . . . . .	13
2.2.3. Basado en grafo . . . . .	14
2.2.4. Análisis de conceptos formales . . . . .	15
2.3. Sistemas de recomendación . . . . .	16
2.3.1. Algoritmos y evaluación . . . . .	18
2.3.2. Temas relacionados . . . . .	28
<b>3. Solución propuesta</b>	<b>39</b>
3.1. Teoría de análisis de conceptos formales . . . . .	40
3.2. Descripción de la tarea . . . . .	45
3.3. Configuración experimental . . . . .	52
3.4. Experimentos y Resultados . . . . .	57
3.4.1. Preprocesamiento y creación del reticulo . . . . .	58
3.4.2. Generación del ranking y evaluación . . . . .	59
3.4.3. Línea Base . . . . .	64
3.4.4. Experimento 1: Popularidad FCA . . . . .	65
3.4.5. Experimento 2: Distancia . . . . .	66
3.4.6. Experimento 3: Penalización por visita . . . . .	67
3.4.7. Experimento 4: Bonificación por visita . . . . .	68
3.4.8. Experimento 5: Mixto distancia - Penalización por visita . . . . .	69

3.4.9. Experimento 6: Mixto distancia - Bonificación por visita	70
3.4.10. Discusión general y resultados finales . . . . .	71
<b>4. Conclusiones y trabajo futuro</b>	<b>75</b>
<b>Bibliografía</b>	<b>79</b>



# Índice de Figuras

2.1. Taxonomía de sistemas híbridos (extraído de ( <a href="#">Aggarwal, 2016</a> ))	23
2.2. Ejemplo de factorización de matrices de rango 2 . . . . .	31
3.1. Ejemplo de diagrama de Hasse . . . . .	43
3.3. Muestra del fichero <code>item_metadata.csv</code> . . . . .	49
3.2. Muestra de la sesión de un usuario . . . . .	50
3.4. Esquema de la tarea de RecSys . . . . .	51
3.5. Algoritmo Próximos vecinos . . . . .	53
3.6. Recuerdo del ejemplo de diagrama de Hasse . . . . .	60



# Índice de Tablas

3.1. Ejemplo de Contexto Formal . . . . .	40
3.2. Ejemplo de Conceptos Formales generados . . . . .	42
3.5. Resultados de la línea base . . . . .	64
3.6. Resultados de experimento 1 . . . . .	65
3.7. Resultados de experimento 2 . . . . .	66
3.8. Resultados de experimento 3 . . . . .	67
3.9. Resultados de experimento 4 . . . . .	68
3.10. Resultados de experimento 5 . . . . .	69
3.11. Resultados de experimento 6 . . . . .	70
3.12. Resultados de los experimentos . . . . .	71



# Capítulo 1

## Introducción

En la era actual de la información, estamos inmersos en un océano de datos, donde la cantidad de información disponible es vasta y crece a un ritmo vertiginoso. Internet y la digitalización de la información han generado una explosión de datos en diversos formatos, desde texto e imágenes hasta vídeos y transacciones. Sin embargo, esta abundancia de información también ha planteado un desafío significativo: cómo encontrar y acceder a la información relevante en medio de este mar de datos.

En este contexto, los buscadores y los sistemas de recomendación han surgido como soluciones clave para abordar este desafío. Los buscadores, como Google, han sido fundamentales para permitir a los usuarios buscar información específica utilizando palabras clave y algoritmos de indexación eficientes. Estos buscadores han revolucionado la forma en que encontramos información al proporcionar resultados precisos y relevantes en cuestión de segundos.

Por otro lado, los sistemas de recomendación han ganado predominancia al aprovechar los datos existentes para sugerir productos, servicios o contenido personalizados. Estos sistemas se basan en algoritmos sofisticados que analizan los patrones de comportamiento de los usuarios, como sus historiales de compras, preferencias o interacciones en redes sociales. Al aprender de estas señales, los sistemas de recomendación pueden ofrecer recomendaciones precisas y adaptadas a los intereses y necesidades individuales de cada usuario.

Aunque los buscadores y los sistemas de recomendación han demostrado ser herramientas eficaces para lidiar con la gran cantidad de información disponible, todavía existen desafíos significativos por superar. El problema

de la sobrecarga de información persiste, ya que la cantidad de datos sigue aumentando exponencialmente. Además, la calidad de las recomendaciones y la capacidad de los buscadores para comprender la intención del usuario y ofrecer resultados más precisos también son áreas que requieren mejoras continuas.

En el presente trabajo, presentamos una propuesta que se basa en la teoría del Análisis de Conceptos Formales (FCA) con el objetivo de mejorar los resultados en el ámbito de la recomendación. Nuestra solución aborda el problema en dos fases distintas: la generación del retículo y la utilización de dicho retículo para realizar las recomendaciones.

En la primera fase, se lleva a cabo la generación del retículo, una estructura fundamental en el análisis de conceptos formales. Al construir el retículo, se obtiene una representación estructurada y organizada de los datos, lo que facilita el análisis y la exploración de la información. En la segunda fase, se exploran diversas formas de utilizar el retículo generado para realizar las recomendaciones. La estructura del retículo proporciona información valiosa sobre las relaciones descubiertas entre los elementos del dominio. Esta información puede ser aprovechada de diferentes maneras para mejorar la precisión y la relevancia de las recomendaciones realizadas. En esta fase, se investigarán y propondrán métodos y técnicas específicas para explorar el retículo y utilizar eficientemente la información derivada de las relaciones descubiertas. Para llevar a cabo este trabajo, utilizaremos una colección de datos proporcionada por el ([RecSys, 2019](#)). Estos datos serán la base sobre la cual construiremos y evaluaremos nuestra solución basada en el análisis de conceptos formales.

Primero veremos la motivación que nos ha llevado a realizar un trabajo de investigación en el contexto de la recomendación y por qué hemos usado FCA y no otras soluciones más usadas como el Aprendizaje Profundo. Posteriormente veremos en detalle la pregunta de investigación que intentamos responder en este trabajo y cuales son los objetivos a conseguir para poder resolverla.

## 1.1. Motivación

En la era actual de la información digital, nos encontramos con una abrumadora cantidad de datos disponibles. En nuestra vida cotidiana, nos en-

frentamos a una amplia gama de decisiones y opciones, desde elegir una película para ver, hasta seleccionar un producto para comprar en línea. Ante esta avalancha de opciones, las técnicas de recomendación se han vuelto indispensables para ayudar a los usuarios a tomar decisiones informadas y descubrir nuevas opciones que se ajusten a sus preferencias.

La sobrecarga de información sigue siendo un desafío significativo para los usuarios; la búsqueda manual y la evaluación exhaustiva de todas las opciones disponibles consumen tiempo y pueden resultar abrumadoras. Es en este contexto donde entran en juego las técnicas de recomendación, que buscan filtrar y personalizar la información para brindar recomendaciones relevantes y útiles a los usuarios.

Dentro de los sistemas de recomendación, existen diversas soluciones, como los recomendadores basados en filtros colaborativos, que agrupan a los usuarios con gustos similares; los recomendadores basados en contenido, que se basan en las preferencias del propio usuario para recomendar elementos relevantes; o los recomendadores híbridos, que fusionan distintas técnicas. Además, los objetivos de los recomendadores pueden variar, desde recomendar un elemento hasta ordenar una lista de elementos según el interés del usuario. Para lograr estos objetivos, se utilizan diversas técnicas, como el Aprendizaje Profundo, el método de los k-vecinos más cercanos, las técnicas basadas en grafos, entre otras.

En los últimos años, ha surgido una tendencia en la utilización de sistemas basados en Aprendizaje Profundo para abordar una amplia gama de problemas. Si bien estos sistemas han demostrado ser altamente efectivos en diversas tareas, también presentan desafíos en términos de su interpretación y predictibilidad.

Los modelos de Aprendizaje Profundo, con su arquitectura compleja y capacidad para aprender representaciones de alto nivel de los datos, pueden capturar patrones y relaciones sofisticadas en los conjuntos de datos. Sin embargo, a medida que estos modelos se vuelven más profundos y complejos, su capacidad de explicar cómo y por qué toman decisiones se vuelve más limitada.

Esta falta de transparencia y comprensión en los modelos de Aprendizaje Profundo plantea **preocupaciones significativas en aplicaciones críticas, como la atención médica, la justicia penal o la toma de decisiones financieras**. La opacidad de estos modelos dificulta la confianza y

la adopción generalizada, ya que los usuarios y las partes interesadas desean entender cómo se llega a una determinada predicción o recomendación.

FCA es una técnica adecuada para este tipo de tareas, ya que permite la organización automática de los usuarios y los elementos a recomendar en una estructura llamada retículo con relaciones generalización-especificación (Una estructura similar a una estructura jerárquica). Esta representación revela relaciones ocultas que pueden escapar a la percepción humana. Además, FCA se basa en un enfoque no supervisado, lo que significa que no requiere un conjunto de entrenamiento inicial para construir el modelo; solo se necesita el historial de un usuario para crear el retículo. Otra ventaja de este enfoque es que su representación es fácilmente comprensible para los usuarios, ya que el retículo puede ser visualizado mediante un diagrama de Hasse, lo que permite una comprensión intuitiva del sistema. Esto contrasta con el Aprendizaje Profundo, donde el usuario no tiene un conocimiento completo de cómo la red realiza las recomendaciones y las salidas pueden resultar impredecibles.

En el retículo generado por FCA, los usuarios están ordenados en base a generalización-especificación, y varios usuarios pueden estar ubicados en el mismo nodo, lo que permite tratar los diferentes nodos como perfiles de usuario y aproximarse a un recomendador basado en filtros colaborativos. Además, el mismo grafo representa las relaciones entre los usuarios y las características de los elementos a ordenar, lo que permite tener en cuenta estas características para realizar un recomendador basado en contenido. De esta manera, utilizando un grafo, podemos desarrollar un recomendador híbrido que aproveche lo mejor de ambos enfoques. Sin embargo, a pesar de sus múltiples ventajas, FCA también presenta desafíos. El algoritmo de creación del retículo es complejo y lento. Además, es posible que se generen retículos de gran tamaño, lo que dificulta su almacenamiento y recorrido.

En relación al campo de la recomendación, existe una abundancia de estudios y datos disponibles, así como la organización de congresos dedicados a los últimos avances en este ámbito. En el presente trabajo, hemos decidido basarnos en una propuesta presentada en el congreso ([RecSys, 2019](#)), la cual nos proporciona un conjunto de datos y un marco de referencia para dar inicio a nuestra investigación. Siguiendo dicha propuesta, nuestro enfoque se centra en la aplicación de la teoría del Análisis de Conceptos Formales en los sistemas de recomendación, con el objetivo de evaluar su utilidad.



## 1.2. Propuesta y objetivos

El presente trabajo tiene como objetivo aplicar eficientemente FCA en el contexto de recomendación para abordar la propuesta en (RecSys, 2019). Este objetivo puede traducirse en la siguiente pregunta de investigación: **”¿Es FCA una teoría útil en el contexto de la recomendación?”**. Para responder a esta pregunta es necesario cumplir los siguientes objetivos de investigación:

1. **Analizar y comprender la teoría de Análisis de Conceptos Formales (FCA)**. Se realizará un estudio exhaustivo de la teoría del FCA. Se examinarán los fundamentos teóricos y los conceptos clave, que se utiliza para analizar relaciones entre objetos y atributos en conjuntos de datos. Comprender esta teoría es fundamental para aplicarla en la generación de recomendaciones.
2. **Realizar un análisis detallado del problema propuesto por (RecSys, 2019) y examinar el conjunto de datos proporcionado**. Se llevará a cabo un análisis detallado del problema planteado por (RecSys, 2019), que proporciona un contexto específico para la aplicación de técnicas de recomendación. Se examinará en profundidad el conjunto de datos proporcionado, lo que implicará comprender su estructura, características y posibles desafíos asociados.
3. **Determinar qué información del conjunto de datos será utilizada en la generación del retículo de conceptos formales**. Se determinará qué información del conjunto de datos será relevante y útil para el proceso de generación del retículo de conceptos formales.
4. **Diseñar y evaluar diferentes enfoques para extraer la información relevante del retículo y utilizarla en la generación de recomendaciones**. Se explorarán diferentes técnicas y algoritmos para aprovechar eficazmente la estructura y las relaciones presentes en el retículo.
5. **Seleccionar los métodos de evaluación adecuados y realizar la evaluación de las propuestas desarrolladas**. Se seleccionarán los métodos de evaluación adecuados para medir la eficacia de las propuestas desarrolladas en el proyecto. La evaluación se realizará para determinar la calidad y el rendimiento de las soluciones propuestas.

6. **Comparar los resultados obtenidos utilizando las propuestas basadas en FCA con diversas líneas base establecidas para validar su eficacia.** Se llevará a cabo una comparación de los resultados obtenidos mediante la implementación de las propuestas basadas en el FCA con respecto a las líneas base establecidas previamente. Esto permitirá validar la eficacia y el valor agregado de los enfoques propuestos.
7. **Realizar una comparación y análisis de las diferentes propuestas, extrayendo conclusiones sobre su desempeño y efectividad.** Se realizará una comparación y análisis exhaustivo de las diferentes propuestas desarrolladas. Se extraerán conclusiones sobre su desempeño y efectividad en la generación de recomendaciones basadas en el FCA. Estas conclusiones proporcionarán información valiosa para comprender el impacto y las aplicaciones potenciales de las técnicas del FCA en el campo de la recomendación.

Además de estos objetivos de investigación nos encontramos con los siguientes objetivos técnicos:

1. **Identificar los requisitos técnicos y organizativos necesarios para la implementación adecuada del sistema de recomendación basado en FCA.** Se identificarán y analizarán los requisitos técnicos y organizativos necesarios para implementar de manera efectiva el sistema de recomendación. Esto implica considerar aspectos como la infraestructura tecnológica requerida, las habilidades y conocimientos necesarios, así como los aspectos organizativos relacionados con la integración del sistema.
2. **Resolver las dificultades técnicas surgidas debido al rendimiento del proceso de generación del retículo.** Durante la implementación del sistema, es posible que surjan dificultades técnicas relacionadas con el rendimiento del proceso de generación del retículo de conceptos. Estas dificultades podrían incluir problemas de consumo de recursos o tiempos de ejecución prolongados. Se buscarán soluciones para abordar y resolver eficazmente estas dificultades.

Estos objetivos específicos se plantean con el propósito de abordar de manera integral el problema de investigación y alcanzar una comprensión

sólida de la utilidad del FCA en el contexto de la recomendación, así como proponer soluciones y enfoques innovadores para mejorar su aplicabilidad.



## Capítulo 2

# Estado del arte

El avance y desarrollo de la tecnología ha tenido un impacto significativo en diversas áreas del conocimiento humano. En particular, el estudio y comprensión del estado actual de un campo específico de investigación se ha convertido en una herramienta esencial para entender su evolución, identificar las tendencias emergentes y establecer las bases para futuras investigaciones.

El objetivo de este estado del arte es analizar y sintetizar los conocimientos más relevantes, estableciendo conexiones entre los diversos estudios y resaltando las principales contribuciones realizadas hasta la fecha. Para lograrlo, se ha realizado una extensa revisión bibliográfica, consultando fuentes académicas, revistas especializadas y otros medios confiables.

En primer lugar, realizaremos un análisis exhaustivo de diversas metodologías para representar conocimiento. A continuación, abordaremos el campo de la detección de temas, el cual constituye un área de investigación relevante en el ámbito de la recomendación. Examinaremos distintos enfoques y perspectivas relacionadas con este tema. Por último, nos adentraremos en los sistemas de recomendación, profundizando en diversos algoritmos y métodos de evaluación utilizados en este contexto.

### 2.1. Representación basada en conocimiento

Según ([Gradmann, 2010](#)), el conocimiento se define como información más contexto, y la organización del conocimiento se basa en organizar la información dentro de un contexto específico. Esto puede hacerse manual o automáticamente. La organización manual implica la participación de expertos y suele ser costosa, mientras que los enfoques automáticos son más populares

debido a su facilidad de creación y uso.

La organización manual se relaciona con las ontologías, que son definiciones formales de un dominio de conocimiento. Las ontologías consisten en una jerarquía de conceptos que se describen mediante atributos y se vinculan mediante relaciones (Fellbaum, 1998). Por otro lado, las organizaciones automáticas se basan en representaciones conceptuales para organizar el conocimiento.

Las ontologías proporcionan una representación formal que permite el razonamiento sobre los datos para inferir nuevas relaciones o conocimientos. Sin embargo, su aplicabilidad se ve limitada en dominios generales o escenarios volátiles. En tales casos, es más adecuada una representación menos estricta que permita cambios y adaptaciones rápidas.

La representación conceptual busca crear una representación más abstracta del contenido mediante los conceptos que aborda. Estas representaciones se generan dinámicamente al analizar los contenidos a representar, inferir sus relaciones latentes y su organización. Esto da como resultado una representación más dinámica, fácil de modificar y más estrechamente relacionada con los contenidos, ya que se infiere directamente de ellos.

Se han propuesto diferentes metodologías para generar representaciones conceptuales. Una de ellas es el análisis de características especiales del contenido, que permite inferir los conceptos a partir de ellas, como se propone en los estudios (Abel et al., 2011) y (Tao et al., 2012). Además, (Meijer, Frasincar, y Hogenboom, 2014) presenta el análisis de los sentidos de las palabras para la extracción de taxonomías.

En cuanto a la detección de conceptos latentes, se han utilizado diversas metodologías, como el enfoque estadístico descrito en (Ramage, Dumais, y Liebling, 2010a), la factorización algebraica mencionada en los estudios (Diaz-Aviles et al., 2012) y (Ticha et al., 2014), así como el análisis formal de conceptos aplicado en este trabajo, según lo propuesto en (Valverde-Albacete y Peláez-Moreno, 2007) utilizando la teoría de Galois.

La representación conceptual se puede derivar automáticamente de los datos mediante técnicas estadísticas como LSA o LDA, que se basan en la frecuencia de términos y su distribución a lo largo de los datos. La agrupación jerárquica y el LDA se consideran en general como técnicas deseables para la organización del conocimiento (Hu, Fang, y Liang, 2014).

No obstante, un problema común en estos métodos probabilísticos es

que la representación resultante no tiene una interpretación evidente para los usuarios (Zheng, Yan, y Zhou, 2015). Aunque refleja conceptos latentes, carece de una representación clara del concepto definido en términos comprensibles para las personas. En cambio, estos conceptos latentes son abstracciones matemáticas de las relaciones entre términos y documentos.

En este contexto, FCA proporciona un marco matemático en el que llevar a cabo este proceso, que ha sido ampliamente aplicado en la literatura de organización de contenidos (Carpineto y Romano, 2004b), (Kim y Compton, 2004), (Poelmans et al., 2013), (Priss, 2000), (Rahman y Chow, 2010) y (Tian, 2006). FCA también crea la representación conceptual, pero los conceptos identificados corresponden a conceptos del mundo real. De hecho, los conceptos se describen mediante la combinación de rasgos que los definen. FCA puede verse como una forma de generar automáticamente representaciones similares a la ontología y ofrece una serie de ventajas:

1. No requiere de ninguna información previa sobre los contenidos para organizarlos.
2. Proporciona una organización basada en una estructura reticular, en lugar de una simple jerarquía. El retículo representa un formalismo que explora mejor las correlaciones, similitudes, anomalías o incluso incoherencias en las estructuras de datos (Carpineto y Romano, 2004b).
3. Ofrece una representación fácilmente legible para los usuarios de la estructura resultante, facilitando su navegación y comprensión.

Una revisión detallada de la aplicación del FCA para la representación del conocimiento es realizada por (Poelmans et al., 2013) y por (Codoceo y Napoli, 2015). Algunas otras investigaciones interesantes sobre este tema son los trabajos de (Tian, 2006) para la construcción de un menú jerárquico para la organización de contenidos; los de (Priss, 2000) y en (Carpineto y Romano, 2004b) para sistemas de Recuperación de Información; la investigación presentada en (Alam y Napoli, 2014) sobre el modelado de Resultados de Consultas SPARQL; o el trabajo de (Kim y Compton, 2004) centrado en la organización de documentos para la Adquisición de Conocimiento.

## 2.2. Detección de temas

La detección de temas es una tarea crucial para analizar grandes corpus de datos y comprender, organizar y resumir su contenido. Se centra en descubrir grupos de contenidos similares que comparten un tema semántico común (Mcauliffe y Blei, 2007).

La investigación en detección de temas comenzó hace varios años, principalmente motivada por la necesidad de gestionar la información en flujos de datos. Uno de los primeros eventos relevantes en esta área fue el *Topic Detection and Tracking* (TDT), celebrado en el marco de TREC (Fiscus y Doddington, 2002). El objetivo de TDT era descubrir y agrupar contenidos relacionados temáticamente en flujos de datos, como las noticias. Los trabajos desarrollados en el contexto de TDT y otros estudios de vanguardia han demostrado ser relativamente efectivos en la detección de temas en contenidos textuales (Allan et al., 1998).

Sin embargo, la atención se ha desplazado hacia las fuentes de datos en redes sociales, especialmente en Twitter. En el estado del arte se han propuesto diversas metodologías, algunas de las cuales presentaremos a continuación.

### 2.2.1. Clasificación

Los enfoques iniciales para la detección de temas utilizaron técnicas de clasificación con éxito en diferentes escenarios (Becker, Naaman, y Gravano, 2011), (Bengel et al., 2004), (Kumaran y Allan, 2004) y (Wayne, 2000). Sin embargo, al tratar con tweets, surgen consideraciones que pueden limitar el rendimiento de los algoritmos de clasificación tradicionales, como se explican en (Anta et al., 2013). Estas consideraciones incluyen la presencia de signos especiales (abreviaturas, emoticonos, hashtags), el uso de argot, la brevedad de los contenidos impuesta por Twitter y las posibles faltas de ortografía.

A pesar de estas consideraciones, se han propuesto enfoques basados en clasificación en el estado del arte. En (Sriram et al., 2010), se propone un algoritmo de clasificación para categorizar los tweets en noticias, opiniones, mensajes personales, eventos y ofertas. Este algoritmo utiliza metadatos relacionados con Twitter sobre los autores de los tweets, como su nombre e información del perfil, para adaptar los métodos de clasificación a los datos de Twitter.

En el trabajo de (Phan, Nguyen, y Horiguchi, 2008), también se abor-



dan las limitaciones de la clasificación mediante el uso de fuentes externas como Wikipedia y MEDLINE para enriquecer el contenido de los tweets y aumentar las características disponibles para el algoritmo de clasificación. Otras soluciones propuestas incluyen la tokenización de los tweets (Laboreiro et al., 2010), el stemming, el análisis ortográfico y el uso de diccionarios (Agarwal et al., 2011).

En el ámbito del TASS (Taller de Análisis de Sentimiento en SEPLN) (Villena-Román y Garcia-Morera, 2013), se presentan enfoques novedosos para la detección de temas basados en metodologías de clasificación. En (Cordobés et al., 2013), se propone una técnica de clasificación basada en el análisis de grafos, argumentando que cualquier texto se puede representar como un grafo y que los temas pueden extraerse de la estructura de dicho grafo. Por otro lado, en (Pla y Hurtado, 2013), se aplica un enfoque clásico de SVM (Máquinas de Vectores de Soporte) basado en características dependientes de Twitter y en el contenido de los tweets expandido léxica y morfológicamente. Para obtener más detalles sobre otros enfoques recientes en este sentido, así como un análisis detallado del modelado de temas desde el punto de vista de los sistemas supervisados, se puede consultar el trabajo de (Blei y MCAULIFFE, 2010).

### 2.2.2. Probabilístico

En la actualidad, los métodos probabilísticos, principalmente el Modelado de Temas Probabilístico (pTM) utilizando LDA, se han vuelto estándar en la detección de temas. Estos métodos buscan identificar el espacio subyacente de temas en base a técnicas probabilísticas. Algunos trabajos relevantes en esta área incluyen (AlSumait, Barbará, y Domeniconi, 2008), (Anthes, 2010), (Guo et al., 2013), (Huang et al., 2012) y (Godin et al., 2013).

A pesar de la amplia aplicación de las técnicas de clustering, especialmente las probabilísticas, en la detección de temas, todavía existen algunas preguntas sin respuesta. Una de las más importantes es determinar la cantidad de temas y, por lo tanto, cuántos grupos se pueden generar (Guo et al., 2013). Para abordar este problema, se han utilizado enfoques como el análisis de la matriz kernel para algoritmos de clustering (Honarkhah y Caers, 2010), los Procesos Jerárquicos de Dirichlet (HDP) en algoritmos LDA (Honarkhah y Caers, 2010) y (Teh et al., 2006), versiones supervisadas de LDA, como LDA etiquetado (*Labelled LDA*) (Ramage et al., 2009) y su aplicación

en Twitter, TwitterLDA (Quercia, Askham, y Crowcroft, 2012), así como el uso de Inferencia Bayesiana en algoritmos probabilísticos (Cheng, He, y Liu, 2014). En este contexto, (Petkos et al., 2014) proponen un enfoque de clustering basado en *Soft Frequent Pattern Mining* (SFPM), una versión menos restrictiva de los algoritmos tradicionales de minería de patrones que no requiere que todos los términos de un conjunto aparezcan juntos con frecuencia. Aunque estas técnicas ofrecen soluciones interesantes y pueden obtener buenos resultados, requieren cierto grado de supervisión (es decir, entrenamiento, parametrización o análisis previo de los datos) y no son de aplicación sencilla.

Además de la cuestión del número de temas, existe otra limitación relacionada con estas metodologías: ¿Cómo pueden los sistemas incorporar el conocimiento previo sobre los temas? En otras palabras, si se dispone de datos de entrenamiento anotados previamente, ¿los sistemas deben estar totalmente determinados por los datos de entrenamiento o deben mostrar cierto grado de adaptabilidad?

### 2.2.3. Basado en grafo

Las técnicas de clustering y LDA basadas en probabilidades han empezado a ser reemplazadas por otras propuestas debido a los problemas mencionados. Algunos enfoques novedosos aplican técnicas de análisis de grafos, como el trabajo de (Figuerola et al., 2005) en Replab 2013 y el trabajo de (Cataldi, Caro, y Schifanella, 2010). Estos métodos utilizan las relaciones entre los términos de los tweets para construir una representación de los datos. Algunos enfoques basados en grafos conectan los tweets mediante aristas que representan la inclusión léxica de frases cortas en lugar de términos (Leskovec, Backstrom, y Kleinberg, 2009). La estructura del grafo se utiliza posteriormente para identificar temas. Por ejemplo, en (Figuerola et al., 2005), las agrupaciones más densas se consideran temas, donde la densidad se calcula en función de todas las relaciones posibles y las que realmente existen. De manera similar, en (Cataldi, Caro, y Schifanella, 2010), se aplica el concepto de "energía del contenido" para encontrar palabras clave emergentes que puedan definir nuevos temas. Además, en (Sayyadi, Hurst, y Maykov, 2009), se utiliza la "centralidad de la interrelación" en el grafo para llevar a cabo el proceso de agrupación.

Los enfoques basados en grafos abordan los problemas relacionados con

la detección de temas, como la adaptación de temas y la necesidad de parametrización. Este tipo de metodología no restringe el número de temas a detectar, a diferencia de los algoritmos de clustering, clasificación o probabilísticos que requieren establecer el número  $K$  de agrupaciones, el número de clases aprendidas durante el entrenamiento o el parámetro  $K$  en LDA. Por lo tanto, el número de temas se determina exclusivamente en función de las características de los datos y se adapta a diferentes conjuntos de datos. Los enfoques basados en grafos también abordan el problema de la detección de nuevos temas teniendo en cuenta el conocimiento previo disponible: el grafo puede generarse utilizando el conocimiento previo y luego expandirse fácilmente con el nuevo conocimiento adquirido a lo largo del tiempo. Estas técnicas también proporcionan una representación visual de las relaciones entre los datos y muestran de manera explícita la estructura de los temas detectados.

#### 2.2.4. Análisis de conceptos formales

El Análisis de conceptos formales (FCA) es una alternativa a los enfoques basados en grafos para abordar los problemas de selección de número de temas y adaptabilidad. El FCA organiza automáticamente los contenidos en una estructura reticular, aprovechando las relaciones entre las características de los datos. Al igual que los enfoques basados en grafos, el FCA no limita el número de temas generados y se adapta a las características específicas de los datos. Además, el FCA permite la inclusión de conocimiento previo en la construcción del retículo, y puede ampliarse fácilmente con nuevos contenidos, generando nuevos temas o incluyendo el contenido en temas existentes.

La propuesta basada en el FCA comparte similitudes con los enfoques basados en grafos al proporcionar una representación similar a un grafo, llamada red de conceptos o retículo, para la detección de temas. Ambas metodologías abordan de manera efectiva los problemas relacionados. Sin embargo, hay diferencias clave entre ellas. Mientras que las representaciones gráficas requieren definir previamente las relaciones entre los datos (por ejemplo, la co-ocurrencia de términos), el FCA es capaz de derivar automáticamente estas relaciones a partir de los datos de entrada. Esto permite que los datos se organicen en conceptos formales que agrupan información similar de manera automática (Ganter y Wille, 1998). Por otro lado, los enfoques basados en grafos requieren un procesamiento adicional de la estructura del

grafo para identificar grupos de datos, como la detección de comunidades. La detección de conceptos formales realizada por el FCA es un proceso determinista y exhaustivo, a diferencia de las representaciones basadas en grafos, que no garantizan esta exhaustividad. Además, la teoría del FCA define una relación de orden parcial entre los conceptos formales, lo que permite la construcción de una red de conceptos. Esta representación de datos es más rica que la proporcionada por los grafos, ya que explora mejor las correlaciones, similitudes, anomalías e incoherencias en las estructuras de datos (Carpineto y Romano, 2004a).

FCA ha sido utilizado previamente para la detección de temas en diversos contextos (Geng et al., 2008) y (Ren et al., 2011). Por ejemplo, en el trabajo de (Geng et al., 2008), se propone un sistema que aplica FCA para agrupar correos electrónicos en función de su contenido y aspectos temporales y sociales relacionados, como los participantes y los destinatarios. En este enfoque, se utilizan funciones de pertenencia difusas para ponderar las características y representar la probabilidad de que un objeto pertenezca a un concepto específico. En (Ren et al., 2011), se aplica una metodología similar para agrupar noticias recopiladas de Internet, utilizando los términos presentes en los títulos como descripción de los documentos. Además, una extensión de estos enfoques se propone en (Maio et al., 2012), donde se emplea *Fuzzy FCA* para incorporar aspectos temporales y modelar datos de Twitter en temas basados en la semántica, utilizando conceptos relacionados con los tweets extraídos de Wikipedia. Estos trabajos demuestran la versatilidad del FCA en la detección de temas en diferentes dominios y su capacidad para considerar diferentes aspectos de los datos.

### 2.3. Sistemas de recomendación

Los sistemas de recomendación (RS) surgieron en los años 90 (Balabanovic y Shoham, 1997) y (Resnick et al., 1994) como una solución al problema de la sobrecarga de información. Estos sistemas se basan en la extracción de información sobre las preferencias de los usuarios con respecto a un conjunto de elementos. Inicialmente, se aplicaron con éxito en la recomendación de productos en sitios de comercio electrónico, como noticias, páginas web, libros, películas y otros productos. Sin embargo, en la actualidad, su campo de aplicación se ha expandido considerablemente e incluye áreas como noticias,

chistes, películas, aplicaciones, sitios web y destinos de viaje (Bidart et al., 2014), (Huang y Bian, 2009) y (Yuan et al., 2013).

En los últimos años, se ha realizado una gran cantidad de trabajo en esta área, que va más allá de lo abordado en este estado del arte. Por ejemplo, la recomendación consciente del contexto se enfoca en considerar los diferentes contextos en los que un usuario puede interactuar con un sistema de recomendación, como la hora del día, la estación del año o el estado de ánimo. Sin embargo, en este trabajo nos centraremos únicamente en los aspectos especiales relacionados con la temática específica abordada.

Para obtener más detalles sobre el campo de los sistemas de recomendación, se pueden consultar diversas fuentes. En (Rao, 2008), así como en (Park et al., 2012) y (Lu et al., 2015), se enumeran una amplia variedad de sistemas de recomendación y se exploran sus campos de aplicación tanto en entornos académicos como empresariales. Para obtener una visión introductoria más completa, el libro de (Jannach et al., 2010) ofrece una perspectiva general sobre los sistemas de recomendación. Además, el libro de (Ricci, Rokach, y Shapira, 2011) y, más recientemente, el libro de (Aggarwal, 2016) abordan aspectos más avanzados en este campo.

La investigación en sistemas de recomendación es un área amplia que involucra diversos campos relacionados, como el aprendizaje automático, la recuperación de información, el procesamiento del lenguaje natural y la minería de datos. Dado el amplio espectro de campos de aplicación y las diferentes áreas de investigación involucradas, los sistemas de recomendación deben abordar varios problemas bien conocidos, como la escalabilidad, la proactividad, la privacidad, la diversidad, la adquisición de información, la integración de información y la integración entre dominios y sistemas. Algunos de estos problemas se exponen y se abordan en el libro de (Ricci, Rokach, y Shapira, 2011).

Tradicionalmente, la literatura en sistemas de recomendación ha estado organizada en torno a dos tipologías principales: recomendadores basados en filtrado colaborativo y recomendadores basados en contenido (Pazzani, 1999). Estas metodologías han sido los pilares fundamentales de la investigación en sistemas de recomendación. Sin embargo, con la aparición de nuevos campos de aplicación y la disponibilidad de diferentes tipos de información, se han diseñado métodos especializados que consideran características específicas. Por ejemplo, se han desarrollado enfoques que incorporan información

temporal, de localización, social o demográfica (Porcel et al., 2012). En las siguientes subsecciones se proporcionan más detalles sobre estas tipologías y los enfoques especializados correspondientes.

La organización de esta sección pretende organizar la literatura de recomendación de acuerdo con estos importantes temas. En particular, como se propone en (Aggarwal, 2016), la sección se organiza en dos categorías principales:

- **Algoritmos y Evaluación:** En esta sección se describen las diversas metodologías y algoritmos propuestos en la literatura de sistemas de recomendación. Se presentan dos enfoques principales ampliamente utilizados: el filtrado colaborativo en la sección 2.3.1 y la recomendación basada en contenido en la sección 2.3.1. Además, en la sección 2.3.1, se exploran propuestas para combinar estas dos metodologías y superar sus limitaciones individuales mediante enfoques híbridos. Por último, la sección 2.3.1 revisa los trabajos en la literatura que se centran en la evaluación formal de los sistemas de recomendación.
- **Temas relacionados:** En esta sección se abordan algunos temas relacionados con el ámbito de investigación de este trabajo. Se exploran los sistemas de recomendación basados en grafos, la descomposición matricial aplicada a sistemas de recomendación, otros enfoques que proponen espacios de representación comunes para la recomendación, y por último, la aplicación del FCA en la tarea de recomendación.

### 2.3.1. Algoritmos y evaluación

Existen varias aproximaciones para el desarrollo de sistemas de recomendación. Una de ellas es el filtrado colaborativo, que se basa en la colaboración entre usuarios para generar recomendaciones. Otra opción es el enfoque basado en el contenido, donde se utilizan características específicas de los elementos recomendados. También se puede optar por la hibridación de diferentes tipos de sistemas de recomendación para superar sus limitaciones individuales. Por último, es necesario contar con métodos de evaluación para medir el desempeño de los sistemas desarrollados, los cuales se abordarán más adelante.

### Recomendadores basados en el filtrado colaborativo

Los recomendadores colaborativos se basan en agrupar a usuarios similares según sus decisiones pasadas (Linden, Smith, y York, 2003) y (Kim et al., 2011). La premisa es que si dos usuarios han mostrado afinidad por un conjunto de artículos en el pasado, es probable que compartan gustos similares en el futuro. Siguiendo esta lógica, cuando dos usuarios son considerados similares, si uno de ellos consume un nuevo artículo, se le recomendará al otro.

Desde el punto de vista técnico, esta tarea está relacionada con el análisis de valores perdidos. El sistema utiliza una matriz de interacciones usuario-artículo que registra las interacciones pasadas, y debe inferir los valores faltantes en función de los valores observados. En el contexto de la recomendación, esta operación es particularmente desafiante debido al tamaño y dispersión típicos de la matriz usuario-artículo (Koren, 2008).

En la literatura se han definido dos tipos principales de métodos de FC. En primer lugar, los algoritmos de FC basados en la vecindad (también conocidos como basados en la memoria) se basan en la creación de vecindarios de usuarios similares -usuarios que valoran por igual los mismos artículos- (FC basada en el usuario) o de artículos similares -artículos valorados por igual por los mismos usuarios- (FC basada en el artículo). La tarea de recomendación se basa, por tanto, en la generación de tales vecindarios siguiendo un proceso que puede considerarse una generalización de los clasificadores de vecino más cercano o de la agrupación de *k-means*. Por tanto, esta metodología depende en gran medida de cómo se defina la similitud entre usuarios o artículos. Las métricas propuestas en la literatura van desde métricas básicas como la similitud del coseno (Devi y Venkatesh, 2009) hasta similitudes más sofisticadas como BM25 (Parra y Brusilovsky, 2010). Sin embargo, la mayoría de estas métricas no tienen en cuenta la longitud de los vectores de valoraciones de los usuarios (es decir, dos usuarios que comparten las mismas 10 valoraciones tienen más probabilidades de tener gustos similares que dos usuarios que sólo comparten un par de valoraciones) (Ma, King, y Lyu, 2007). Para superar esta limitación, estas medidas pueden aplicarse en combinación con la similitud de Jaccard (u otras medidas que tengan en cuenta el solapamiento de vectores) (Candillier, Meyer, y Fessant, 2008). Para más detalles, se incluye una extensa recopilación de medidas de similitud en el contexto de la recomendación por Filtrado Colaborativo en (Bobadilla et al.,

2013) y en (Pirasteh, Hwang, y Jung, 2015).

Otra aproximación en el Filtrado Colaborativo (FC) son los métodos basados en modelos, los cuales generan un modelo resumido de los datos durante el entrenamiento y lo utilizan para inferir nuevas recomendaciones. En la literatura se han estudiado diversas técnicas de modelización, muchas de ellas basadas en clasificadores de datos (Billsus y Pazzani, 1998). Algunos ejemplos son los sistemas basados en reglas (Shyu et al., 2005) y los clasificadores Naive Bayes (Miyahara y Pazzani, 2000). También se han propuesto enfoques basados en Redes Neuronales (RN), especialmente en los últimos años con el auge del Aprendizaje Profundo. Por ejemplo, (Salakhutdinov, Mnih, y Hinton, 2007) presenta un modelo de Máquinas de Boltzmann Restringidas (RBM), donde las unidades ocultas representan elementos y las valoraciones de los usuarios activan las unidades visibles. Las RBM han demostrado obtener resultados similares a otros modelos del estado del arte, como los modelos de factores latentes, en escenarios como el Premio Netflix (Bennett y Lanning, 2007).

Aunque los métodos basados en modelos son considerados más refinados y sofisticados, la complejidad asociada a la creación del modelo de recomendación puede limitar su aplicabilidad en escenarios de big data o en tiempo real. Por otro lado, los algoritmos basados en vecinos son más ligeros, ya que la recomendación es un proceso basado en instancias y no requiere la creación previa de un modelo.

Una línea de investigación más reciente se centra en el filtrado colaborativo con conciencia de grupo, que consiste en dividir la matriz usuario-elemento del FC en subgrupos más pequeños (submatrices). Luego, se realiza el proceso de recomendación de forma individual en cada subgrupo, utilizando algoritmos de FC específicos para cada submatriz (Ji y Shen, 2015). Sin embargo, esta metodología asume que cada usuario/artículo solo pertenece a un subgrupo, lo cual no refleja con precisión la realidad de los intereses múltiples de los usuarios. Para abordar este problema, se han propuesto enfoques con co-clusters solapados (Xu et al., 2012) y (Zhang et al., 2013), y más recientemente en (Wu et al., 2016b).

Los sistemas de filtrado colaborativo enfrentan el desafío de requerir una cantidad significativa de interacciones para identificar similitudes entre usuarios y realizar recomendaciones. En ausencia de estas interacciones, surge el problema del inicio frío, que ocurre cuando nuevos usuarios carecen de infor-



mación y el sistema no puede proporcionar recomendaciones. La literatura ha propuesto diversas soluciones para abordar este problema (Schein et al., 2002), (Pirasteh, Hwang, y Jung, 2015) y (Adomavicius y Tuzhilin, 2005). La complejidad de estos sistemas se menciona con frecuencia en la literatura (Koren y Bell, 2011) y (Su y Khoshgoftaar, 2009). Otro desafío se relaciona con la dimensión temporal, ya que las preferencias de los usuarios pueden cambiar con el tiempo. Por ejemplo, dos usuarios que estuvieron interesados en un evento pasado pueden tener gustos diferentes en el presente.

### Recomendadores basados contenido

Estos sistemas se basan en el contenido de los artículos que se van a recomendar. La idea fundamental detrás de esta metodología es que un usuario interesado en un determinado tipo de contenido, como películas de acción, es más probable que esté interesado en otras películas de acción que en películas de géneros diferentes, como comedias románticas. Por lo tanto, las recomendaciones que se ofrecen a los usuarios consisten en artículos similares a los que ya han consumido (Lops, Degemmis, y Semeraro, 2011).

A diferencia de los sistemas de Filtrado Colaborativo, estos sistemas no utilizan las valoraciones de otros usuarios, sino que se centran principalmente en las valoraciones y preferencias del propio usuario. Esto presenta tanto ventajas como desventajas. Al basarse en el contenido de los artículos, estas recomendaciones se sustentan en las relaciones ricas y contextuales descubiertas entre los artículos y los perfiles de los usuarios (Ignatov y Kuznetsov, 2009). Además, cuando se introducen nuevos artículos, es fácil recomendarlos a los usuarios a los que les gusta el tipo de contenido relacionado con esos artículos. Por otro lado, debido a que los sistemas basados en el contenido solo consideran las características del artículo en sí, tienden a generar recomendaciones sobre-especializadas, recomendando constantemente artículos con atributos similares (Zhang, Callan, y Minka, 2002).

Debido a las ventajas y desventajas mencionadas, los sistemas basados en contenido se han aplicado principalmente, aunque no exclusivamente, en dominios donde se pueden obtener representaciones informativas de los elementos, como la recomendación de noticias (Kompan y Bieliková, 2010). Como resultado, la clasificación de textos y la recuperación de información son las técnicas más utilizadas para desarrollar este tipo de sistemas (Pazzani y Billsus, 2007).

En resumen, los sistemas basados en contenido funcionan mediante la clasificación de elementos que tienen una representación "similar" de su contenido, como características o palabras clave descriptivas. Esta similitud se determina mediante una métrica que depende de la tarea y las representaciones de los elementos, siendo la similitud del coseno la más ampliamente aplicada. Para obtener más detalles sobre estas métricas, se puede consultar (Spertus, Sahami, y Buyukkokten, 2005). En este enfoque, las recomendaciones se realizan ofreciendo elementos clasificados junto con los elementos presentes en los perfiles de los usuarios. Este método se conoce como clasificación del vecino más cercano y es una de las metodologías más simples para la recomendación basada en contenido.

### Recomendadores híbridos

En las secciones anteriores se han explicado los dos principales tipos de sistemas de recomendación: los basados en contenido y los de filtrado colaborativo. Ambos tienen fortalezas y debilidades distintas cuando se utilizan de manera individual. En este contexto, los sistemas híbridos buscan combinar ambos tipos de sistemas para aprovechar todo el conocimiento disponible en diversas fuentes de datos, como las interacciones del usuario o el contenido de los artículos. El objetivo es aprovechar las ventajas complementarias de estos sistemas y evitar problemas asociados con los sistemas de recomendación, como el arranque en frío (Pereira y Hruschka, 2015), (Tejeda-Lorente et al., 2014) y (Wang y Wang, 2014).

El proceso de hibridación de sistemas de recomendación implica combinar de manera efectiva el funcionamiento de diferentes sistemas. En la Figura 2.1, propuesta por (Aggarwal, 2016), se presenta una clasificación de las metodologías de hibridación basada en siete mecanismos propuestos por (Burke, 2007):

- Ponderadas: Combina las puntuaciones de diferentes sistemas en una puntuación unificada (Mobasher, Jin, y Zhou, 2003).
- Mixtas: (Smyth y Cotter, 2000).
- De conmutación: Alterna entre diferentes sistemas según las necesidades del momento. Por ejemplo, puede utilizar un sistema basado en el conocimiento en las primeras etapas para evitar problemas de arran-

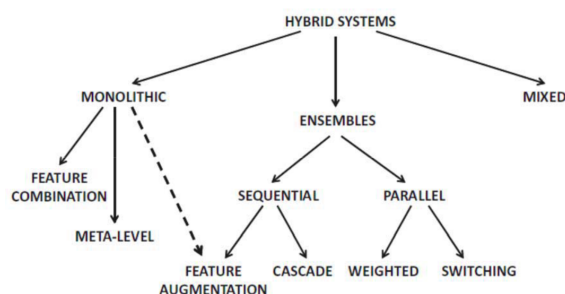


Figura 2.1: Taxonomía de sistemas híbridos (extraído de (Aggarwal, 2016))

que en frío y luego cambiar a un sistema basado en contenido o filtrado colaborativo (Billsus y Pazzani, 2000).

- **Combinación de características:** Combina características de diferentes fuentes en un solo sistema.
- **Aumento de características:** Utiliza la salida de un sistema para crear características adicionales para el siguiente sistema (O’Sullivan, Smyth, y Wilson, 2004).
- **Cascada:** Un sistema de recomendación refina las recomendaciones generadas por otro sistema (Burke, 2002).
- **Metanivel:** Comparte el modelo utilizado por un sistema de recomendación con otro sistema (Pazzani y Billsus, 1997).

Estos mecanismos proporcionan diferentes enfoques para combinar los sistemas y lograr una hibridación eficiente en sistemas de recomendación.

En cuanto a las categorizaciones de mayor nivel, se han propuesto clasificaciones adicionales, como la presentada en la figura por (Aggarwal, 2016), así como la distinción entre sistemas canalizados (donde la salida de un sistema se concatena con otro sistema) y sistemas paralelos (donde múltiples sistemas funcionan en paralelo) propuesta por (Jannach et al., 2010).

En resumen, los sistemas híbridos permiten la integración de diferentes tipos de sistemas de recomendación y múltiples fuentes de datos, lo que mejora el rendimiento de los sistemas individuales.

## Evaluación

Existen tres paradigmas principales para evaluar los sistemas de recomendación: estudios de usuarios, evaluaciones offline y evaluaciones online.

En los estudios de usuarios, se solicita a los participantes que interactúen con los sistemas de recomendación y realicen tareas específicas, a la vez que proporcionan su opinión sobre la calidad de las recomendaciones.

Las evaluaciones online son similares a los estudios de usuarios, pero en este caso, los usuarios reales de sistemas de recomendación completamente desarrollados son quienes participan y se recopila información sobre su interacción con el sistema.

Por último, las evaluaciones offline se basan en probar el rendimiento de los algoritmos de recomendación utilizando datos históricos. Se comparan los resultados del sistema con las interacciones reales de los usuarios, registradas en los datos históricos.

Los estudios de usuarios son considerados la metodología de evaluación más deseable, ya que proporcionan información sobre las interacciones reales de los usuarios con el sistema en un entorno controlado. Sin embargo, su realización puede ser difícil y costosa, ya que requiere la contratación de un gran número de usuarios. Además, los resultados no son directamente comparables con estudios externos y la replicación de los resultados puede ser complicada. Esto limita la extrapolación de las conclusiones obtenidas de estos estudios. Las evaluaciones en línea presentan desafíos similares, aunque permiten una comparación más fácil con otros algoritmos.

Debido a estos desafíos técnicos, el paradigma de evaluación offline es el más comúnmente utilizado. Aunque estos métodos no evalúan el rendimiento futuro de los sistemas de recomendación, ya que se basan en interacciones pasadas de los usuarios, han sido ampliamente adoptados en el estado del arte. Existen numerosos marcos y medidas de evaluación estandarizados que han facilitado la comparación de diferentes metodologías y sistemas de recomendación.

La aplicación de estas metodologías de evaluación ha tenido que adaptarse a nuevos escenarios, como Twitter, donde los enfoques tradicionales pueden no ser adecuados. Evaluar la relevancia de un tweet para un usuario en particular es un proceso complejo. En este contexto, es importante tener en cuenta que, a diferencia de los sistemas de recomendación tradicionales con valoraciones explícitas, en Twitter las valoraciones son implícitas

y unarias. Esto significa que los usuarios no evalúan explícitamente un elemento, sino que su interés por un tweet se deduce de su comportamiento, como publicar un contenido, compartir una URL o retuitear otros tweets. Además, estas "valoraciones" deben considerarse como unarias, ya que la falta de interacción de un usuario con un tweet puede indicar tanto falta de interés como falta de conocimiento. Por lo tanto, solo se pueden considerar las interacciones positivas de los usuarios. Formalmente, la valoración de un usuario  $u_i$  por un contenido  $c_j$  se define como:

$$r_{i,j} \in \{0,1\} = \begin{cases} 1 & u_i \text{ likes } c_j \\ 0 & \text{otherwise } (u_i \text{ does not like or not know } c_j) \end{cases} \quad (2.1)$$

El principal método para evaluar este interés es mediante un estudio de usuarios, como los de (Ramage, Dumais, y Liebling, 2010b) o (Phelan et al., 2011). Sin embargo, el gran tiempo, recursos y esfuerzo que requiere, hace inviable su aplicación para grandes cantidades de datos. Otra solución más sencilla propuesta en el estado del arte es inferir las evaluaciones a partir de los comentarios implícitos de los usuarios. En el contexto de Twitter, significa que si un usuario ha tuiteado o retuiteado algún contenido, éste puede considerarse interesante para él. Esta suposición se ha aplicado en muchos trabajos (Ramage, Dumais, y Liebling, 2010b), demostrando su idoneidad en comparación con los estudios de usuarios.

A continuación, se detallan algunas de las métricas de evaluación más destacadas que se han considerado tradicionalmente en la literatura. En este sentido, un aspecto importante a considerar es que la evaluación de los sistemas de recomendación no se basa en un único criterio (McNee, Riedl, y Konstan, 2006). Aunque las métricas basadas en la precisión son la medida más ampliamente aplicada (de hecho, los algoritmos de última generación se definen comúnmente de acuerdo con este criterio), algunos otros aspectos también han sido considerados en la literatura: cobertura, novedad (Konstan et al., 2006), confiabilidad (Cramer et al., 2008), serendipia (es decir, descubrimiento afortunado) (Ge, Delgado-Battenfeld, y Jannach, 2010), diversidad (Castells et al., 2014), robustez (Mobasher et al., 2007) o escalabilidad (Takács et al., 2009). Para una discusión más detallada sobre la evaluación de los sistemas de recomendación, así como las limitaciones que presentan, consulte el capítulo correspondiente en (Aggarwal, 2016), en el libro de (Shani

y Gunawardana, 2011) o la encuesta presentada por (Luo et al., 2014).

### Métricas de evaluación

Una cuestión importante en el proceso de evaluación es determinar la métrica más adecuada para el escenario de recomendación. Esto está relacionado con los criterios a los que se debe prestar atención en el proceso de evaluación. En la literatura, se han propuesto diferentes tipos de métricas, que se detallan a continuación:

#### *Métricas basadas en precisión*

Este tipo de métricas, así como el escenario de evaluación, se derivan del campo de la recuperación de información. Para ello, miden la calidad de un sistema comparando los ítems recomendados con un *ground-truth* que incluye los elementos que realmente son interesantes para los usuarios objetivo. Entre estas métricas, se incluyen:

- **Precisión (P):** Mide la probabilidad de que un elemento recomendado cumpla las preferencias del usuario.
- **Recuperación (R):** Mide la proporción de elementos recomendados que resultan relevantes respecto al número total de elementos relevantes.
- **Medida F:** Es la media armónica de su precisión y recuperación.
- **Área bajo la curva ROC (AUC):** Mide la calidad de un sistema comparando la tasa de verdaderos positivos y la de falsos positivos.

#### *Métricas basadas en error*

Las métricas de precisión mencionadas se centran en medir si un elemento recomendado es relevante (es decir, la relevancia binaria aplicada habitualmente en Recuperación de Información). En cambio, las métricas de error miden la calidad de los sistemas desde el punto de vista del error producido en la predicción de la calificación. En concreto, si  $r_{uj}$  es la valoración de un usuario  $u$  sobre el ítem  $j$  (contenido en el conjunto de prueba) y  $\hat{r}_{uj}$  es la valoración predicha por el sistema, el error viene dado por  $e_{uj} = \hat{r}_{uj} - r_{uj}$ . Entre este tipo de medidas, destacan dos métricas: El error cuadrático medio (RMSE) y el error absoluto medio (MAE).

- **RMSE:** Esta métrica representa la desviación típica muestral de las diferencias entre las valoraciones previstas y las reales. La formulación

de esta métrica es la siguiente:

$$RMSE = \sqrt{\frac{\sum_{(u,j) \in E} (e_{uj})^2}{|E|}} \quad (2.2)$$

Siendo  $E$  el conjunto de entradas sobre las que se realiza la evaluación y  $e_{uj}$  es el error de predicción sobre la entrada  $(u, j)$ . Una característica importante de esta métrica es que tiende a penalizar desproporcionadamente los errores grandes porque los errores se elevan al cuadrado en la formulación. En consecuencia, se ve más afectada por valores atípicos o grandes errores.

- **MAE:** El error absoluto medio es la media del error absoluto  $e_{uj}$  según la siguiente formulación:

$$MAE = \frac{\sum_{(u,j) \in E} |e_{uj}|}{|E|} \quad (2.3)$$

### *Métricas basadas en ranking*

Estas métricas tratan de medir no sólo la precisión de las recomendaciones, sino también lo bien que el sistema recomendador ordena las predicciones. Algunas de las métricas más destacadas de esta categoría son:

- **Success@K:** Esta métrica representa la probabilidad media de que un elemento relevante aparezca en la posición K de la clasificación.
- **Mean Reciprocal Rank (MRR):** El MRR, tal y como lo definen (Chakrabarti et al., 2008), se refiere a la posición inversa del primer elemento relevante en el ranking. Calculada como:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (2.4)$$

Donde  $rank_i$  se refiere a la posición del primer elemento relevante y  $Q$  es el ranking que se quiere evaluar.

- **Mean Average Precision (MAP):** MAP mide la media de la precisión media (AP) para cada lista de recomendaciones, donde la AP es igual a la media para la precisión en cada elemento relevante “visto”

en la lista de recomendaciones.

$$mAP = \frac{1}{|Q|} \sum_{i=1}^{|Q|} AP_i \quad (2.5)$$

Donde  $AP$  es la precisión media de cada clase y  $Q$  es el ranking que se quiere evaluar.

- **Normalized Discount Cumulative Gain (NDCG):** NDCG evalúa una lista de recomendaciones midiendo en qué medida mejora la calidad global de una clasificación dada la aparición de un documento con una relevancia determinada (la ofrecida por el algoritmo de recomendación) en una posición determinada de la clasificación.

### 2.3.2. Temas relacionados

Esta sección aborda temas relacionados con las cuestiones tratadas en este trabajo, en particular, nuestra propuesta basada en FCA. En resumen, se incluyen:

- Sistemas de recomendación que utilizan grafos para modelar las interacciones usuario-elemento (nuestra representación basada en FCA es un modelo gráfico de estas interacciones).
- Metodologías de descomposición matricial aplicadas al campo de la recomendación (FCA puede considerarse una metodología de descomposición matricial).
- Enfoques que proponen un espacio de representación común para modelar las interacciones usuario-elemento (como lo hacemos en este trabajo).
- Algunos trabajos que aplican directamente FCA al proceso de recomendación.

### Recomendación basada en grafos

Los grafos permiten representar usuarios e ítems, así como las relaciones entre ellos, lo que revela la estructura inherente a estas relaciones. Una representación basada en grafos puede identificar las relaciones entre ítems, agrupando los similares (Gori y Pucci, 2007), y también las relaciones entre



usuarios, permitiendo la detección de comunidades de usuarios con gustos o preferencias similares (Bidart et al., 2014). Esta relación usuario-usuario es especialmente importante, ya que ha demostrado ofrecer el mejor rendimiento para la recomendación por Filtrado Colaborativo (Hernando et al., 2014). Además, los métodos basados en grafos superan el problema de la dispersión usuario-artículo, ya que no es necesario que los usuarios o los artículos compartan muchas valoraciones para considerarse relacionados, siempre que haya algún camino entre ellos.

Esta representación basada en grafos también es útil para predecir valoraciones de artículos o crear rankings de recomendación mediante la propagación de información a través del grafo. Por ejemplo, (Adomavicius y Kwon, 2011) presentan un enfoque basado en grafos para mejorar la diversidad en las recomendaciones, mientras que (Fouss, Pirotte, y Saerens, 2005) y (Brand, 2005) utilizan la estructura del grafo para ofrecer recomendaciones más precisas. (Shi, 2013) proponen un enfoque basado en grafos llamado recomendación de coste-flujo, que aborda varios aspectos de la recomendación, como la precisión, la diversidad, la similitud y la larga cola.

Además, la representación gráfica permite visualizar el conjunto de información y las relaciones que la vinculan, lo cual es beneficioso para la recomendación. Sin embargo, la visualización basada en grafos para sistemas de recomendación ha sido abordada en pocos trabajos. Algunos ejemplos notables son los trabajos presentados en (Hernando et al., 2014) y (Hernando et al., 2013), que proponen la visualización de grafos en forma de mapa de grafos jerárquico (Abello, 2004). Esta forma de representación facilita la visualización de grafos grandes, como los asociados a los sistemas de recomendación.

### Recomendación basada en matrices

Este tipo de técnicas busca reducir la dimensionalidad de la matriz usuario-artículo sin perder la representatividad de los datos, a través de la identificación de factores subyacentes. Estas metodologías se basan en la premisa de que muchas porciones de la matriz usuario-artículo están altamente correlacionadas, lo que implica la existencia de grupos de usuarios y artículos altamente correlacionados en función de factores latentes. En consecuencia, los datos pueden ser representados en una matriz de bajo rango basada en estos factores latentes.

En el contexto de la recomendación, estas técnicas permiten descomponer la matriz usuario-artículo original en función de las interacciones entre usuarios y artículos, identificando estos factores latentes. Estos factores incluyen grupos de usuarios y artículos estrechamente relacionados, lo cual se espera que conduzca a recomendaciones precisas. De hecho, estas metodologías son especialmente efectivas en el ámbito de la recomendación debido a las siguientes razones:

1. Las matrices usuario-elemento son muy dispersas, por lo que la reducción esperada es mayor que en otros contextos.
2. Existe un conjunto claro de factores latentes que han dado lugar a los datos de la matriz usuario-elemento (las preferencias del usuario), expresados a través de las relaciones entre usuarios y elementos.

### Método de descomposición

Se han propuesto muchos métodos para la descomposición de matrices; a continuación se presentan algunos de los más comunes, prestando especial atención a los aplicados al ámbito de la recomendación.

#### *SVD*

SVD (Singular Value Decomposition) (Eldén y Berr, 2008) es un método para descomponer una matriz rectangular  $A \in R^{m \times n}$  ( $m > n$ ) en el producto de tres matrices:

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V \quad (2.6)$$

En la recomendación, las filas de las matrices  $U$  y  $V$  pueden interpretarse como vectores que representan la actitud o lealtad del usuario y del artículo hacia un tema o factor específico. Los valores singulares de la matriz  $\Sigma$ , por otro lado, representan la importancia relativa de cada tema o factor. En otras palabras, para un usuario  $u_j$  dado, la primera fila de  $U$  captura su interés en el tema o factor correspondiente, mientras que la primera fila de  $V$  contiene los elementos relacionados con ese factor. Los valores de  $\Sigma$  reflejan la relevancia o peso del interés del usuario en el factor correspondiente. La Figura 2.2 extraída del libro de (Aggarwal, 2016) ilustra este concepto y muestra una matriz de valoraciones con 7 usuarios y 6 elementos. Se observa una clara tendencia en las valoraciones de los usuarios, relacionada con los géneros de las películas (histórico y romántico). Esta matriz puede ser factorizada en

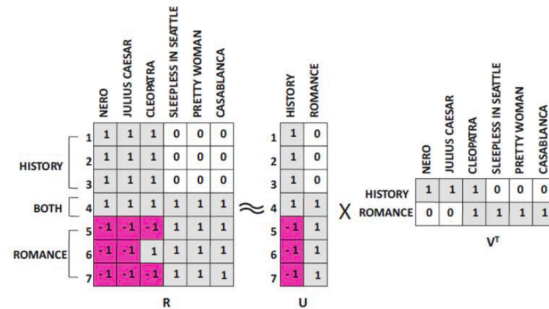


Figura 2.2: Ejemplo de factorización de matrices de rango 2

factores de rango 2, donde la matriz  $U$  representa el interés de los usuarios en los géneros y la matriz  $V$  muestra la relación entre esos géneros y las películas.

Basándose en la idea del SVD, se han propuesto metodologías más refinadas, como el conocido SVD++ (Koren, 2008), que ha obtenido resultados destacados en el Reto Netflix. El SVD++ es una mejora del SVD que incorpora información implícita de retroalimentación, como los elementos que el usuario ha valorado previamente. También se ha propuesto una extensión temporal del SVD, denominada time-SVD++ (Koren, 2009b). La idea principal detrás de este algoritmo es modelar los parámetros del SVD como funciones del tiempo. En particular, el time-SVD++ considera que los sesgos del usuario y del ítem, así como los factores del usuario, varían en función del tiempo.

### **BMF**

Al igual que la SVD, la BMF (Factorización de Matrices Binarias) se enfoca en reducir la dimensionalidad de una matriz, pero se aplica a una matriz binaria que representa el interés de un usuario con valores binarios (0 para desinteresado/no visto y 1 para interesado/visto).

La BMF es una descomposición de la matriz binaria original  $I \in \{0, 1\}^{n \times m}$  en el producto de dos matrices booleanas  $P \cdot Q$  donde  $P \in \{0, 1\}^{n \times k}$  y  $Q \in \{0, 1\}^{k \times m}$ , con el valor más pequeño posible para  $k$ :

$$(P \cdot Q)_{ij} = \sum_{l=1}^k P_{il} \cdot Q_{lj} \quad (2.7)$$

### **NNF**

La factorización de matrices no negativas (NNF, por sus siglas en inglés) se refiere a una metodología específica de descomposición de matrices

aplicada a matrices no negativas. La NNF descompone la matriz original  $V \in \mathbb{N}^{n \times m}$  en dos matrices  $W \in \mathbb{N}^{n \times r}$  y  $H \in \mathbb{N}^{r \times m}$ , donde  $r$  es el número de factores latentes, también conocidos como aspectos, de tal manera que:

$$W * H = V \quad (2.8)$$

Esta técnica, como afirma (Aggarwal, 2016), facilita la comprensión de las interacciones usuario-elemento, especialmente en los casos en los que los usuarios no disponen de ningún mecanismo para especificar una aversión (es decir, valoraciones unarias).

### **FCA**

Como ya se ha mencionado, FCA puede verse como una metodología de descomposición matricial para datos binarios, que factoriza la matriz de entrada en un conjunto de factores latentes (es decir, conceptos formales) que agrupan objetos cercanos según sus atributos compartidos. FCA garantiza que esta factorización es única (es decir, los mismos datos de entrada siempre dan como resultado el mismo conjunto de conceptos formales) porque, como demostraron (Nenova, Ignatov, y Konstantinov, 2013), la factorización proporcionada por FCA es la factorización óptima de la matriz de entrada.

### **Enfoques de recomendación**

Las técnicas de recomendación basadas en esta metodología se centran principalmente en el enfoque de Filtrado Colaborativo, es decir, encontrar los usuarios más similares a un usuario objetivo. Teniendo esto en cuenta, dadas las matrices reducidas obtenidas mediante la aplicación de las técnicas mencionadas, la recomendación se basará en encontrar los usuarios más similares (vecindario) según los factores latentes. A continuación, se utilizarán otros factores relacionados con los usuarios del vecindario para recomendar nuevos artículos, asociados a estos factores. El fundamento es que el espacio latente (obtenido mediante la descomposición de la matriz usuario-artículo) identificará mejor las relaciones implícitas usuario-artículo mediante los factores latentes detectados (Koren, 2008).

Algunos de los primeros usos de los modelos de factores latentes se propusieron como métodos independientes para la recomendación (Aggarwal y Parthasarathy, 2001) y (Sarwar et al., 2000), lo que dio lugar a resultados de vanguardia. A raíz de estas primeras propuestas, se han propuesto diferentes formas de factorización matricial, como el análisis de fábrica (Canny, 2002), los modelos semánticos latentes (Hofmann, 2004), NNF (Zhang et

al., 2006). Tras la popularización de estos modelos por el concurso Netflix Prize (Bell y Koren, 2007), trabajos posteriores presentaron modelos más avanzados. En (Paterek, 2007) se discuten los aspectos relacionados con los modelos de factores latentes (por ejemplo, un modelo factorial asimétrico) y se propusieron algunas de las innovaciones básicas que posteriormente se combinaron para crear metodologías punteras, como SVD++ (Koren, 2009a) y (Koren, 2008). Otras propuestas novedosas en este sentido se presentan en (Devooght, Kourtellis, y Mantrach, 2015) y en (Jain y Dhillon, 2013).

Los escenarios de aplicación de los modelos de factores latentes incluyen campos tan diversos como el comercio electrónico (Schafer, Konstan, y Riedl, 2004), la recomendación de películas (Pirasteh, Hwang, y Jung, 2015), o la recomendación de viajes (Noulas et al., 2012) y (Wang, Terrovitis, y Mammoulis, 2013). También se han aplicado para tratar entornos Big-data como en el trabajo de (Yu et al., 2014).

El estudio de los enfoques que abordan la tarea de recomendación aplicando FCA para factorizar la matriz usuario-elemento se dejan para la siguiente sección centrada en las técnicas de recomendación basadas en FCA.

### **Espacio común de representación para la recomendación**

Los trabajos en este sentido se centran en fusionar los espacios de representación de usuarios e ítems (es decir, las matrices de valoración y contenido). Al compartir una representación común, sólo sería necesario encontrar los ítems más cercanos en el espacio de representación de los perfiles de usuario.

Las representaciones basadas en el conocimiento se han propuesto como base para establecer una representación común en la recomendación. Por ejemplo, en el trabajo (Shoval, Maidel, y Shapira, 2008), se propone una ontología común para representar a los usuarios y los ítems. Uno de los desafíos asociados con esta representación ontológica es la necesidad de definir una medida de similitud para relacionar usuarios e ítems. Otros trabajos en la literatura, como (Cantador, 2008), también exploran la creación de una representación tripartita que conecta usuarios e ítems en una capa de representación común basada en conceptos que los relacionan.

Un enfoque propuesto en (Huang y Bian, 2015) utiliza FCA para relacionar usuarios e ítems en un espacio de representación común. En este trabajo, aplican FCA para relacionar dos ontologías: una que refleja las preferencias de los turistas y otra que describe los servicios ofrecidos por los proveedores

de información turística. En este enfoque, los conceptos de ambas ontologías se consideran objetos en el contexto formal, mientras que los atributos son conceptos extraídos manualmente de la literatura turística. El resultado final es un entramado conceptual que representa formalmente a los usuarios, descritos en la ontología turística, y a los ítems, descritos en la ontología de los proveedores turísticos, junto con las relaciones entre ellos.

También se han propuesto representaciones basadas en grafos con el fin de proporcionar un modelo común para usuarios e ítems. Por ejemplo, (Chen, Gan, y Song, 2013) proponen un grafo bipartito como modelo para integrar las representaciones de usuarios y elementos. En el grafo, los usuarios se relacionan según los enlaces entre ellos, mientras que los elementos se representan según su similitud de contenido.

En el contexto de los sistemas híbridos, se han propuesto algunas técnicas en este sentido. El fundamento es combinar los sistemas de Filtrado Colaborativo, que operan sobre la dimensión usuario-artículo, y los sistemas basados en el Contenido, que operan sobre la dimensión artículo-atributo, en un único sistema que integre la información de ambas dimensiones en una representación común. Uno de los primeros intentos en esta línea se presenta en (Basilico y Hofmann, 2004), donde los autores proponen un enfoque unificado para integrar toda la información (matriz usuario-elemento y atributos del ítem) en mapas de características. El trabajo propuesto en (Singh y Gordon, 2008) presenta un enfoque análogo que aplica un modelo colectivo de factorización matricial para factorizar simultáneamente las matrices usuario-elemento y atributos-elemento en un modelo común. (McAuley y Leskovec, 2013) también presentan un modelo de factorización para combinar el texto de la reseña y las valoraciones. También se han propuesto modelos basados en regresión para crear estos modelos de factores latentes (Agarwal, Chen, y Long, 2011) y (Ning y Karypis, 2012). Por último, (Moore et al., 2013), (Feng et al., 2015) o (Wu et al., 2013) proponen el uso de incrustaciones para representar usuarios y elementos en un espacio latente común.

En relación a esta última propuesta, el reciente auge de las Redes Neuronales Artificiales (RNA) y el Aprendizaje Profundo ha atraído el interés de los investigadores en el campo de los Sistemas de Recomendación. Las RNA han demostrado ser capaces de aprender representaciones latentes complejas de los datos de entrada. En este sentido, algunos trabajos han intentado aprovechar esta capacidad para implementar espacios de representación co-

munes para usuarios e ítems. El primer intento en este sentido se presenta en (Salakhutdinov, Mnih, y Hinton, 2007), donde se aplica una Máquina de Boltzmann Restringida para la Recomendación top-N. (Iyyer et al., 2014) proponen una red neuronal para responder preguntas sobre párrafos que modela tanto preguntas en forma de párrafos como respuestas en el mismo espacio vectorial. De este modo, esperan fomentar que las representaciones de las preguntas estén cerca de sus representaciones de respuestas correctas y lejos de las respuestas incorrectas en este espacio vectorial común. Una idea similar se presenta en (Socher et al., 2014). Sus autores presentan un modelo basado en un espacio de representación multimodal para mapear las salidas de una red convolucional aplicada para detectar objetos visuales en imágenes y representaciones vectoriales para frases generadas por una DT-RNN (*Dependency Tree - Recurrent Neural Network*). Se espera que permita vincular imágenes con frases que puedan describir la imagen. Enfoques similares para vincular imágenes y palabras en un espacio común se presentan en (Socher y Fei-Fei, 2010) y en (Srivastava y Salakhutdinov, 2012). Más recientemente, se han propuesto autocodificadores para predecir las valoraciones de los usuarios (Sedhain et al., 2015) y (Wang, Wang, y Yeung, 2014). Especialmente interesante es el enfoque presentado por (Wu et al., 2016a) que propone un nuevo modelo para la recomendación de CF. Los autores proponen un *Autoencoders Denoising* para construir una representación latente de usuarios y elementos, que es capaz de superar a otros recomendadores del estado del arte para la recomendación top-N.

### Recomendación basada en FCA

El contexto de un sistema de recomendación puede interpretarse como un grafo bipartito dividido en usuarios ( $U$ ) e ítems ( $I$ ). Las aristas de este grafo, de la forma  $\rho = r(u, i)$ , establecen la relación de interés del usuario  $u$  por y el ítem  $i$  ponderado con una valoración  $r$ . Siguiendo la teoría FCA, la tripleta  $(U, I, \rho)$  puede interpretarse como un *contexto formal* (o un *contexto de recomendación*), según la definición de la sección 3.1, que puede factorizarse en un conjunto de conceptos formales FCA que incluye el conjunto de usuarios que han valorado el mismo conjunto de ítems.

En este sentido, varios trabajos han abordado el problema de la recomendación desde el punto de vista del FCA. En (Simovici, Fomenky, y Kunz, 2012), los autores aplican la base FCA para obtener subconjuntos de usua-

rios que comparten las mismas compras. A continuación, calculan la entropía de cada subconjunto con el fin de encontrar los conjuntos de usuarios más adecuados para recomendar un artículo específico. En (du Boucher-Ryan y Bridge, 2005) los autores proponen un enfoque de Filtrado Colaborativo que pretende aprovechar la estructura de la red para encontrar similitudes entre usuarios en función de los artículos con los que interactúan. Para ello, se proponen dos métodos basados en el concepto de nivel de entrada: uno basado en el nivel de entrada de un atributo y otro en el nivel de entrada de un usuario. Este último es especialmente interesante ya que plantean una metodología para recorrer el entramado con el fin de encontrar los usuarios en la vecindad del usuario objetivo.

Las Reglas de Asociación pueden ser también una valiosa técnica para generar recomendaciones. En este sentido, en (Zhou, Hui, y Chang, 2004) se aplican reglas de asociación a la minería de uso de la web para detectar patrones de navegación. Dada una secuencia de acceso a la web, realizada por un usuario en una sesión, las reglas identificadas se utilizarán para recomendar nuevos contenidos a los que acceder por el usuario. También se han propuesto reglas de asociación para ampliar perfiles de usuario. Algunos ejemplos de esto último se presentan en (Shaw, Xu, y Geva, 2010) y en (Sobhanam y Mariappan, 2013). En (Ignatov y Kuznetsov, 2009), y ampliado en (Ignatov, Kuznetsov, y Poelmans, 2012), los autores aplican FCA a un enfoque de Filtrado Colaborativo pero en lugar de recomendar artículos (como en los ejemplos anteriores), su sistema recomienda terminología para Publicidad en Internet: dada una empresa que ha utilizado algunos términos en el pasado para campañas de marketing; el sistema recomienda nuevos términos relacionados. La recomendación se realiza mediante reglas de asociación: cuanto mayor es la confianza de la regla de asociación, más probable es que se recomiende el consecuente de la regla.

(Senatore y Pasi, 2013) presenta otro ejemplo de enfoque de filtrado colaborativo basado en FCA. La particularidad de este enfoque es la aplicación del FCA en datos difusos (es decir, en lugar de utilizar valores binarios, los valores son continuos en un intervalo  $[0-1]$ ). Para llevar a cabo el proceso de recomendación, los autores proponen un algoritmo básico de filtrado colaborativo que recomienda los artículos ya vistos por los usuarios que comparten algún artículo con el usuario objetivo. Las recomendaciones se clasifican en función de los valores difusos.



Sin embargo, la recomendación basada en el filtrado colaborativo no es la única que puede abordarse mediante FCA, sino que también puede aplicarse a la recomendación basada en el contenido. En este contexto, en lugar de considerar los elementos a recomendar como atributos, se consideran los contenidos de dichos elementos. (Ignatov et al., 2013) aplica FCA a una plataforma de crowdsourcing para representar a los usuarios según el contenido (principalmente palabras clave) de los proyectos de esta plataforma con los que los usuarios ya han interactuado. Su propuesta de modelado tiene en cuenta que los atributos pueden ser multivaluados mediante el uso de conceptos formales multivaluados (más concretamente conceptos triádicos (Wille, 1995)). Basándose en esta modelización, proponen dos metodologías de recomendación diferentes:

- Recomendar usuarios similares a un usuario objetivo: El sistema busca usuarios que hayan interactuado con contenidos similares a los relacionados con el usuario objetivo. Para ello, se utilizan los intentos del concepto formal en el que aparece el usuario objetivo para encontrar otros usuarios que pertenezcan a alguno de estos ítems.
- Recomendar usuarios antagonistas (es decir, usuarios que han interactuado con el mismo conjunto de contenidos pero cuya opinión sobre ellos es completamente diferente). El sistema busca usuarios que compartan el mismo conjunto de atributos (los del mismo concepto formal) y calcula la distancia entre ellos, según sus interacciones con estos atributos (es decir, su opinión sobre los proyectos de la plataforma que se identifican por estos atributos).

En (Li y Murata, 2010) se utiliza FCA para modelar los perfiles de los artículos y construir el conjunto de recomendaciones candidatas. Para crear los perfiles de los ítems, se crea un contexto formal teniendo en cuenta los ítems y sus metadatos. Este contexto formal, representado en un entramado, se utiliza para inferir relaciones entre el usuario y los ítems (descripciones basadas en FCA, reflejadas en los conceptos formales del contexto formal); recomendando los ítems relacionados con estas descripciones (es decir, aquellos que pertenecen al contexto formal). El trabajo de (Maio et al., 2012) propone un enfoque de recomendación basado en FCA para su aplicación en un entorno de e-learning. Más concretamente, los autores aplican *Fuzzy*

*Formal Concept Analysis* (FFCA) para modelar el contenido de los RSS-feeds. En este escenario, dado el contexto de aprendizaje de un usuario, se le recomiendan los conceptos más similares de la red (según la similitud de Wu y Palmer). En (González et al., 2016), se propone un modelado conceptual para relacionar usuarios e elementos para luego extender esa representación con la integración de rasgos semánticos relacionados con el contenido de los elementos, esta idea se ha usado de inspiración en el presente trabajo. Otros enfoques interesantes de la recomendación basada en FCA en escenarios de *e-learning* se detallan en (Fang y Zheng, 2009) y en (Lau et al., 2007).

El FCA y la Recomendación también se han aplicado conjuntamente en otros escenarios. Una aplicación interesante se describe en (Kashnitsky y Ignatov, 2015). Aquí, FCA se propone para modelar un conjunto de clasificadores en un Sistema de Clasificadores Múltiples de acuerdo con sus predicciones y, dado un nuevo contenido a clasificar; la estructura de retículo se utiliza para seleccionar el clasificador adecuado. En (Manuel y Recuero, 2008) se usa FCA como base para generar un modelo de clusterinm y así permitir la herencia múltiple y su descripción de forma automática de los clusters. En (Asmus et al., 2014) se ha aplicado FCA para desarrollar un sistema de recomendación de algoritmos de optimización de caja negra. Por otro lado en (Watson y Weber, 2020) se defiende la necesidad de recomendaciones explicativas para que el usuario confíe en los resultados, para ello se propone un sistema de recomendación basado en contexto enriquecido con FCA para realizar las explicaciones. En (Cornejo et al., 2019) tratan de describir los hábitos de los usuario de una red social basada en ubicaciones usando FCA para relacionar las horas y los atributos del los lugares que el usuario ha visitado. FCA se usa también para entornos de computacion en la nube, en (Indira et al., 2021) proponen el uso conjunto del algoritmo CLARANS, usado para resolver problemas de dimensionalidad, conjuntamente con FCA para mejorar la precisión.

En (Valverde-Albacete y Peláez-Moreno, 2013) se detallan algunas de las principales posibilidades y retos, así como las líneas de futuro de la aplicación del FCA al campo de los Sistemas de Recomendación. Aunque este trabajo se centra en la relación del FCA y la Recuperación de Información, la mayoría de sus conclusiones son también aplicables al campo de los Sistemas de Recomendación.

## Capítulo 3

# Solución propuesta

Como se ha mencionado en las secciones anteriores, FCA es un método eficaz para representar datos interrelacionados de una forma intuitiva y así poder descubrir nuevas conexiones entre ellos que no conocíamos. En este trabajo, queremos evaluar si FCA es capaz de abordar la tarea de recomendación propuesta por (RecSys, 2019) de manera efectiva, con la ventaja de ofrecer una representación de su funcionamiento más comprensible para el ser humano que otras alternativas como el aprendizaje profundo.

La tarea propuesta consiste en ordenar por relevancia un conjunto de hoteles que se han mostrado al usuario teniendo un amplio histórico de distintos usuarios. Para abordar la tarea se ha dividido el proceso en dos fases. La primera es la creación del retículo de conceptos formales con los datos proporcionados. En esta primera fase se han encontrado distintos problemas de rendimiento que se han debido de solucionar y que impiden que sea posible comparar nuestros resultados con los demás participantes, con esto se ha decidido crear una línea base para poder verificar el funcionamiento del experimento. La segunda fase es la ordenación de los hoteles mostrados al usuario en función de la probabilidad de que el usuario seleccione cada hotel. Para esta segunda fase se ha necesitado de un sistema que transforme la información del retículo en un ranking, este sistema está formado por distintas ponderaciones que se explicarán posteriormente, las cuales están basadas en popularidad, en la frecuencia de visita del usuario y en la distancia entre el perfil de usuario y el hotel en el retículo.

Para comprender el trabajo de recomendación, primero necesitamos conocer FCA y su funcionamiento. Una vez conocido FCA y su funcionamiento, se pasará a explicar en detalle la tarea propuesta por (RecSys, 2019),

	Atributo 1	Atributo 2	Atributo 3	Atributo 4	Atributo 5
Objeto 1	X				X
Objeto 2	X	X	X		
Objeto 3	X	X		X	
Objeto 4					X

Tabla 3.1: Ejemplo de Contexto Formal

la colección de datos propuesta y las métricas propuestas. Posteriormente entraremos en el proceso de creación de los experimentos, los problemas encontrados, las decisiones tomadas y el porqué de ellas. A continuación, veremos en detalle los distintos experimentos y hablaremos sobre los resultados obtenidos individualmente. Y finalmente se realizara una discusión general comparando todos los resultados.

### 3.1. Teoría de análisis de conceptos formales

El Análisis de Conceptos Formales (FCA) es una rama de la matemática que se ocupa de la formación y clasificación de conceptos a partir de datos (Belohlavek, 2008), (Ganter y Wille, 1998), (Wille, 1992) y (Wille, 2009). Se basa en las nociones de retículos y conjuntos ordenados para representar las relaciones entre los objetos y los atributos que los caracterizan de una forma jerárquica.

FCA utiliza un modelo llamado **contexto formal** para organizar la información. Un contexto formal es una estructura de conjunto  $\mathbb{K} := (G, M, I)$ , donde  $G$  es un **conjunto de objetos** (formales),  $M$  es un **conjunto de atributos** (formales) y  $I$ , una relación binaria del tipo  $G$  "tiene"  $M$  del tipo  $I \subseteq G \times M$ , denotada como  $glm$ , que significa que el objeto  $g$  tiene el atributo  $m$ . Un ejemplo de contexto formal se muestra en el Cuadro 3.1, donde se tiene un conjunto de objetos ( $G$ ), un conjunto de atributos ( $M$ ) y una relación entre ellos ( $I$ ), indicada por las cruces en la tabla, que indican que el objeto  $g$  tiene el atributo  $m$ .

A partir de la información del contexto formal, se puede generar un **conjunto de conceptos formales**. Para definirlos se necesita la siguiente operación de derivación:

$$\begin{aligned} A \rightarrow A^I &:= \{m \in M \mid glm \text{ for all } g \in A\} \\ B \rightarrow B^I &:= \{g \in G \mid glm \text{ for all } m \in B\} \end{aligned} \quad (3.1)$$

Donde  $A$  es un conjunto de objetos ( $A \subseteq M$ ),  $B$  es un conjunto de atributos ( $B \subseteq M$ ) e  $I$  es el *operador primo*. Aplicando el operador primo a  $A$  obtenemos el conjunto o aquellos atributos que están presentes en todos los objetos pertenecientes a  $A$ , denotado como  $A^I$ . Por otro lado, aplicando el operador primo a  $B$  obtenemos el conjunto de objetos que tienen al menos los atributos dados en  $B$ , denotado como  $B^I$ . Esta operación satisface las siguientes propiedades:

$$\begin{aligned} \mathbb{Z}_1 \subseteq \mathbb{Z}_2 &\implies \mathbb{Z}_1^I \supseteq \mathbb{Z}_2^I \\ \mathbb{Z} &\subseteq \mathbb{Z}^{II} \\ \mathbb{Z}^{III} &\subseteq \mathbb{Z}^I \end{aligned} \quad (3.2)$$

Así, un concepto formal es una tupla  $(A, B)$  si y solo si  $A \subseteq G$  es un conjunto de objetos y  $B \subseteq M$  es un conjunto de atributos que describen estos objetos; siendo  $B$  la **extensión** de  $A$  ( $A = B^I$ ) y, a la inversa,  $A$  es la **intensión** de  $B$  ( $B = A^I$ ) (Ganter y Obiedkov, 2016). Un concepto formal tiene las siguientes propiedades:

- Si un objeto  $a \in A$  está etiquetado con un atributo  $b$ , entonces  $b$  debe estar incluido en  $B$ , es decir, **la intención del concepto formal incluye todos los atributos compartidos por los objetos de la extensión.**
- A la inversa, si un objeto  $a$  está etiquetado con todos los atributos en  $B$ , entonces  $a$  debe estar incluido en  $A$ , es decir, **la extensión del concepto formal incluye todos aquellos objetos filtrados por la intención.**

Para ejemplificar la generación de conceptos formales, a partir del contexto formal del cuadro 3.1, se generan los conceptos formales del Cuadro 3.2. Si nos fijamos por ejemplo en el concepto 4 ( $C_4$ ) nos indica que los objetos 2 y 3 tienen los atributos 1 y 2, si nos fijamos en la tabla vemos cruces en las intersecciones de ambos objetos con ambos atributos, esto no quiere decir que los objetos no pueden tener más atributos, solo nos indican los atributos que tienen en común ambos objetos. Para obtener los atributos

	Extensión	intensión
$C_1$	{O1,O2,O3,O4}	$\emptyset$
$C_2$	{O1,O2,O3}	{Attr1}
$C_3$	{O1,O4}	{Attr5}
$C_4$	{O2,O3}	{Attr1, Attr2}
$C_5$	{O1}	{Attr1, Attr5}
$C_6$	{O2}	{Attr1, Attr2, Attr3}
$C_7$	{O3}	{Attr1, Attr2, Attr4}
$C_8$	$\emptyset$	{Attr1, Attr2, Attr3, Attr4, Attr5}

Tabla 3.2: Ejemplo de Conceptos Formales generados

de un objeto en concreto tenemos que ir a un concepto más específico (más adelante veremos que habría que recorrer el retículo hacia abajo) para llegar a por ejemplo  $C_6$  (como veremos más adelante es el **concepto objeto** del objeto 2) y vemos que todos sus atributos son el 1, 2 y 3 dado que no hay ningún concepto más específico del objeto.

Los conceptos formales pueden ordenarse formalmente (de manera parcial) en una relación subconcepto-superconcepto en función de sus extensiones. Esta basado en la forma natural en que los humanos solemos ordenar los conceptos: un coche es un subconcepto de vehículo porque todo coche es un vehículo. Para ello, es posible definir una relación de orden  $\leq$  sobre los conceptos formales que **los ordena del más genérico al más específico** de la siguiente manera:

$$(A, B) \leq (C, D) \iff A \subseteq C \quad (3.3)$$

Donde  $(C, D)$  es llamado superconcepto de  $(A, B)$  e, inversamente,  $(A, B)$  es un subconcepto de  $(C, D)$ . Es decir,  $(A, B)$  es más específico que  $(C, D)$ .

Se puede demostrar que el orden resultante es un retículo, que se denomina retículo de conceptos, denotado como  $\mathfrak{B}(G, M, I)$ , asociado al contexto formal. En este retículo cuanto más abajo nos coloquemos, más específicos serán los nodos (más atributos y menos objetos) y, en cambio, cuanto más arriba, más genéricos (menos atributos y más objetos).

Dado que los retículos de conceptos son conjuntos ordenados, se pueden representar de forma natural en términos de diagramas de Hasse ([Ganter y Wille, 1998](#)). En un diagrama de Hasse:

- Existe exactamente un nodo para concepto formal.

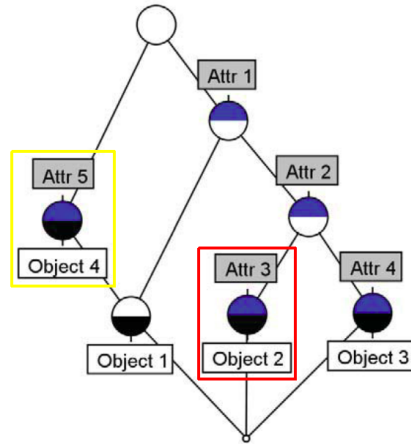


Figura 3.1: Ejemplo de diagrama de Hasse

- Si  $C \subseteq C'$ , entonces  $C'$  se coloca encima de  $C$ . Es decir,  $C$  es un sub-concepto de  $C'$  o  $C'$  es un super-concepto de  $C$ .
- Si  $C \subseteq C'$  pero no hay ningún concepto  $C''$  intermedio tal que  $C \subseteq C'' \subseteq C'$ , entonces habrá una línea uniendo  $C$  y  $C'$ .

En la Figura 3.1 puede verse un ejemplo de la representación reticular de conceptos correspondiente al Contexto Formal de la Tabla 3.1. En esta figura, las etiquetas blancas se refieren a las entradas (objetos) y las grises a las características (atributos). Si un nodo tiene adjuntado una etiqueta de objeto quiere decir que representa su concepto de objeto ( $\gamma g$ ) y se denota mediante un semicírculo negro; a la inversa, el nodo que tiene adjuntado una etiqueta de atributo representa su concepto de atributo ( $\gamma m$ ) y se denota mediante un semicírculo azul. Para evitar una representación sobrecargada, cada concepto formal se representa con un conjunto mínimo de etiquetas de objeto y atributo.

A partir de este diagrama, cada concepto formal puede reconstruirse fácilmente como sigue:

- La extensión incluye todos los objetos representados en los nodos de todos los posibles caminos que van desde el concepto formal objetivo hasta el nodo hoja del retículo. Por ejemplo, la extensión del concepto formal rodeado con un cuadrado amarillo en la Figura 3.1 es {Object

1, Object 4}, en cambio el concepto formal marcado con un cuadrado rojo solo tiene en su extensión {Object 2}.

- La intensión incluye todos los atributos representados en los nodos de todos los posibles caminos que van desde el concepto formal objetivo hasta el nodo raíz del retículo. Por ejemplo, en la Figura 3.1 la intensión del concepto marcado con un cuadrado rojo es {Attr 1, Attr 2, Attr 3}, en cambio la del nodo marcado con un cuadrado amarillo es únicamente {Attr 5}.

Cabe recordar que como esta representado el diagrama de Hasse cuanto más abajo nos situamos en el retículo, más específico es el concepto formal, es decir, hay una mayor cantidad de atributos en la intensión y un menor número de objetos en la extensión. Y, lo mismo ocurre a la inversa, cuanto más arriba nos situamos en el retículo más generalizamos, es decir, hay una menor cantidad de atributos en la intensión y una mayor cantidad de objetos en la extensión. Para más detalles sobre cómo leer retículos de conceptos mostrados como diagramas de Hasse, consulte el libro de (Ganter y Obiedkov, 2016), en particular la sección 1.3.

En los retículos de conceptos, dos tipos importantes de conceptos formales, necesarios para nuestro posterior proceso de recomendación, son los **conceptos objeto** y los **conceptos atributo**:

- **Concepto objeto:** Denotado como  $\gamma g$ , asociado a un objeto  $g$  es el concepto más específico (el concepto más pequeño) que incluye a  $g$  en su extensión. Para construirlo, es necesario incluir en su intensión todos los atributos de  $g$ , e incluir en su extensión, además de  $g$ , todos aquellos objetos etiquetados exactamente con los mismos atributos que  $g$  ( $\gamma g := (\{g\}^{II}, \{g\}^I)$ ).

Si observamos un diagrama de Hasse como el de la Figura 3.1, **el concepto objeto es el concepto formal (nodo) más específico que contenga el objeto**. Por ejemplo, el nodo rodeado con un cuadrado rojo es el concepto objeto del Objeto 2.

**Que sea un concepto objeto no quiere decir que dicho objeto sea el único de su extensión**, solo quiere decir que el objeto no se encuentra en nodos más específicos, por ejemplo, el nodo marcado con un cuadrado amarillo es el concepto objeto del Objeto 4 pero contiene en su extensión además al Objeto 1.



- **Concepto atributo:** Denotado como  $\mu m$ , asociado con el atributo  $m$  es el concepto más genérico que incluye a  $m$  en su intensión ( $\mu m := (\{m\}^I, \{m\}^{II})$ ). Se puede construir de forma dual a un concepto de objeto:

1. Añadir todos los objetos etiquetados con la característica  $m$  a la extensión.
2. Además del atributo  $m$ , añadir los atributos compartidos por esos objetos a la intensión.

Si observamos un diagrama de Hasse como el de la Figura 3.1, **el concepto atributo es el concepto formal (nodo) situado más arriba que contenga el atributo**, es decir, no existirá ningún concepto formal más arriba que contenga el atributo. Por ejemplo, el nodo rodeado con un cuadrado amarillo es el concepto atributo del Attr 5.

**Que sea un concepto atributo no quiere decir que dicho atributo sea el único de su intensión**, solo quiere decir que el atributo no se encuentra en nodos superiores, por ejemplo, el nodo marcado con un cuadrado rojo es el concepto atributo del Attr 3 pero contiene en su extensión además a los Atributos 1 y 2.

## 3.2. Descripción de la tarea

El objetivo de este trabajo es evaluar la aplicabilidad de FCA (Formal Concept Analysis) en sistemas de recomendación, como el propuesto en el congreso (RecSys, 2019). Según su página web<sup>1</sup>, "El reto de este año consiste en desarrollar un sistema de recomendación basado en sesiones y sensible al contexto para adaptar una lista de hoteles en función de las necesidades del usuario.". La idea es predecir qué hotel elegirá el usuario entre los resultados de una búsqueda realizada en Trivago.

Para ello, se propone utilizar FCA para generar un retículo de conceptos formales que represente las relaciones entre los usuarios y los hoteles de una forma comprensible para el ser humano y que permita agrupar a los usuarios en perfiles similares y, posteriormente, se utilizará la información del retículo para ordenar los hoteles según la probabilidad de que el usuario los seleccione.

---

<sup>1</sup><https://recsys.trivago.cloud/challenge/>

Para realizar la tarea, (RecSys, 2019), proporciona un conjunto de datos<sup>2</sup>, que contiene información de **250852** usuarios distintos, **927142** hoteles y **157** características de dichos hoteles. Desgraciadamente, como veremos en futuras secciones, la complejidad del algoritmo de creación de FCA sumado al conjunto de datos tan grande que se proporciona va a provocar ciertas decisiones para poder realizar el trabajo. El conjunto de datos para ser más manejable está separado en vario archivos:

- **train.csv**: Un conjunto de datos con las acciones que han realizado diferentes usuarios con los resultados esperados para poder entrenar un modelo supervisado.
- **test.csv**: Un conjunto de datos similar a **train.csv**. No obstante en este caso no se incluye la solución esperada. Este conjunto esta pensado para evaluar las soluciones desarrolladas.
- **test\_ground\_truth.csv**: Este conjunto de datos contiene las soluciones del conjunto de test para poder comparar la salida del modelo y asi poder puntuarlo.
- **item\_metadata.csv**: Este conjunto de datos contiene distintas características (wifi, garaje, valoracion en estrellas...) de cada hotel y contiene información de los hoteles presentes en el conjunto de entrenamiento y de prueba.

Los conjuntos de datos de entrenamiento y prueba contienen los siguientes campos:

<b>Campo</b>	<b>Descripción</b>
user_id	Identificador único del usuario.
session_id	Identificador único de la sesión del usuario.
timestamp	Milisegundos desde la creación de UNIX hasta el momento de la interacción del usuario.
step	Número de secuencia de la acción en una sesión.
action_type	Identificador del tipo de acción llevada a cabo por el usuario.

<sup>2</sup>Descargable desde: <https://recsys2019data.trivago.com/>

reference	Referencia el valor de la acción como está descrito en el punto de "action_type".
platform	Página web con dominio del país usado para la búsqueda.
city	Nombre de la ciudad actual dentro del contexto de la búsqueda.
device	Dispositivo usado para realizar la búsqueda.
current_filters	Lista de filtro separados por " " activados en un momento dado.
impressions	Lista de elementos separados por " " mostrados al usuario en el momento en el que ha hecho clic (ver action_type = clickout_item).
prices	Lista de precios de los elementos mostrados al usuario separados por " " (ver action_type = clickout_item).

Dentro del campo action\_type podemos encontrar las siguientes acciones con sus respectivos identificadores:

Action_type	Descripción
clickout item	Indica que el usuario hace clic en un elemento de los listados en "impressions" y es redirigido a alguna página. El valor del campo "reference" hace referencia al campo "item_id".
interaction item rating	El usuario interactúa con la puntuación o un comentario de un elemento. El valor del campo "reference" hace referencia al campo "item_id".
interaction item info	El usuario interactúa con la información de un elemento. El valor del campo "reference" hace referencia al campo "item_id".
interaction item image	El usuario interactúa con la imagen de un elemento. El valor del campo "reference" hace referencia al campo "item_id".
interaction item deals	El usuario clica en el botón de ver más detalles del elemento. El valor del campo "reference" hace referencia al campo "item_id".

change of sort order	El usuario cambia el orden de los elementos. El valor del campo "reference" es la descripción del orden.
filter selection	El usuario selecciona un filtro. El valor del campo "reference" es la descripción del filtro.
search for item	El usuario realiza una búsqueda. El valor del campo "reference" hace referencia al campo "item_id".
search for destination	el usuario busca un destino. El valor del campo "reference" es el nombre del destino.
search for poi	El usuario ha buscado por algún punto de interés. El valor del campo "reference" es el nombre del punto de interés.

En la Figura 3.2 podemos observar la representación de la sesión de un usuario completa. En esta sesión, un usuario de la plataforma de EE.UU. ha utilizado trivago en un dispositivo de escritorio. Las acciones de esta sesión son las siguientes:

- (tipo de acción: *search for destination*, referencia: Barcelona, España): El usuario busca Barcelona, España.
- (tipo de acción: *filter selection*, referencia: "Focus on Distance"): Se activa el filtro "centrarse en la distancia". En este momento la columna `current_filters` indica que es el único filtro que está activo.
- (tipo de acción: *search for poi*, referencia: Port de Barcelona): El usuario busca un punto de interés (POI), el Puerto de Barcelona.
- (tipo de acción: *interaction item deals*, referencia: 40255): El usuario consulta el botón "Más ofertas" del artículo 40255. El filtro "centrarse en la distancia" ya no está activado.
- (tipo de acción: *clickout item*, referencia: 40225): El usuario hace clic en el elemento 40225. La lista completa de artículos mostrados y sus precios asociados se puede ver en las columnas "impresiones" y "precio".
- (tipo de acción: *search for item*, referencia: 81770): El usuario busca el artículo 81770.

- (tipo de acción: *interaction item info*, referencia: 81770): El usuario interactúa con la información del artículo 81770.
- (tipo de acción: *clickout item*, referencia: 81770): El usuario hace clic en el elemento 81770. La lista completa de artículos y sus precios asociados puede verse en las columnas "impresiones" y "precio".

Por último, existe una información extra en el archivo `item_metadata.csv` que contiene metadatos de los distintos hoteles. Este archivo contiene dos campos:

Campo	Descripción
item_id	Identificador único de un elemento al que hacen referencia algunos campos del conjunto de datos.
properties	Características del elemento separadas por " "

A modo de ejemplo del fichero `item_metadata.csv`, podemos ver un extracto en la Figura 3.3. Este fichero nos indica que el hotel con id 5101 tiene televisión por satélite, campo de golf, transporte al aeropuerto, etc.

	item_id	properties
0	5101	Satellite TV Golf Course Airport Shuttle Cosme...
1	5416	Satellite TV Cosmetic Mirror Safe (Hotel) Tele...
2	5834	Satellite TV Cosmetic Mirror Safe (Hotel) Tele...
3	5910	Satellite TV Sailing Cosmetic Mirror Telephone...
4	6066	Satellite TV Sailing Diving Cosmetic Mirror Sa...
5	6094	Satellite TV Sailing Safe (Hotel) Telephone Ho...
6	6288	Satellite TV Airport Shuttle Cosmetic Mirror S...
7	6358	Cosmetic Mirror Safe (Hotel) Telephone Hotel C...
8	6456	Satellite TV Cosmetic Mirror Safe (Hotel) Tele...
9	6561	Satellite TV Golf Course Cosmetic Mirror Safe ...

Figura 3.3: Muestra del fichero `item_metadata.csv`

Una vez que se ha adquirido conocimiento acerca del conjunto de datos, es posible profundizar en la descripción de la tarea propuesta por ([RecSys](#),

user_id	session_id	timestamp	step	action_type	reference	platform	city	device	current_filters	impressions	prices
93F7WGHBP03A	569f5ea70df51	1541543231	1	search for destination	Barcelona, Spain	US	Barcelona, Spain	desktop			
93F7WGHBP03A	569f5ea70df51	1541543269	2	filter selection	Focus on Distance	US	Barcelona, Spain	desktop	Focus on Distance		
93F7WGHBP03A	569f5ea70df51	1541543269	3	search for poi	Port de Barcelona	US	Barcelona, Spain	desktop	Focus on Distance		
93F7WGHBP03A	569f5ea70df51	1541543371	4	interaction item deals	40255	US	Barcelona, Spain	desktop		6744 40181 40630 84610 2282416	162 91 218 190 176
93F7WGHBP03A	569f5ea70df51	1541543425	5	clickout item	40255	US	Barcelona, Spain	desktop		1258693 974937 147509 128238 7998246	365 272 159 139 240
93F7WGHBP03A	569f5ea70df51	1541543741	6	search for item	81770	US	Barcelona, Spain	desktop		40255 3058538 1637385 40285 147502	136 5099 164 116 90
93F7WGHBP03A	569f5ea70df51	1541543770	7	interaction item info	81770	US	Barcelona, Spain	desktop		921707 40849 6757 12770 893733	192 191 213 109 178
93F7WGHBP03A	569f5ea70df51	1541543813	8	clickout item	81770	US	Barcelona, Spain	desktop		685091 147522 40708 860451 6819	131 128 168 101 331
93F7WGHBP03A	569f5ea70df51	1541543741	6	search for item	81770	US	Barcelona, Spain	desktop		6832 40396 6621784 40197 6743	347 245 199 65 359
93F7WGHBP03A	569f5ea70df51	1541543770	7	interaction item info	81770	US	Barcelona, Spain	desktop		147488 40635 6177052 6742 1319782	233 227 270 294 625
93F7WGHBP03A	569f5ea70df51	1541543813	8	clickout item	81770	US	Barcelona, Spain	desktop		40763 945255 83855 39937 1870125	208 174 121 217 226
										1354432 6812 82400 40181 6834	616 293 166 91 198
										81770 5056102 40797 923935 40284	274 272 123 130 131

Figura 3.2: Muestra de la sesión de un usuario

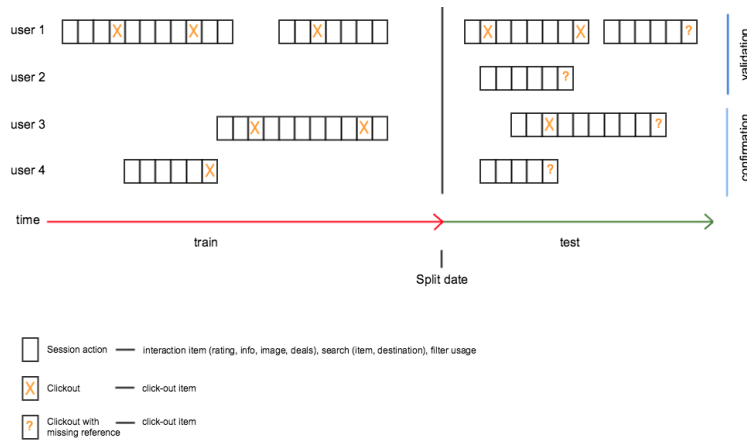


Figura 3.4: Esquema de la tarea de RecSys

2019). En la figura 3.4, se presenta un esquema que ilustra las diferentes sesiones de varios usuarios. Si se examinan las acciones del usuario 1, se pueden observar cuatro sesiones distintas representadas como cuadrados colindantes. Durante las primeras tres sesiones, el usuario lleva a cabo diversas acciones indeterminadas (cuadrados blancos) y realiza dos clics en algún elemento (cuadrados con una "x" amarilla). En la última sesión, se muestra una acción de clickout item que se representa con un cuadrado con un signo de interrogación, lo que indica que no se conoce la referencia (es decir, el elemento que ha seleccionado el usuario). En esta última acción, es posible obtener la lista de elementos que se presentaron al usuario y en los que pudo hacer clic en el campo impressions. La tarea consiste en predecir el elemento que ha seleccionado el usuario, ordenando esta lista presente en impressions según la probabilidad de que el elemento aparezca en el campo reference. Este valor del campo reference se encuentra en el conjunto de datos ground\_truth, lo que permite comprobar la respuesta y evaluar el experimento de manera objetiva.

La tarea propuesta evalúa las distintas propuestas utilizando la métrica denominada *Mean Reciprocal Rank (MRR)*, que es una medida basada en el ordenamiento de los elementos. Esta métrica considera la posición del hotel seleccionado por el usuario en la lista generada, penalizando aquellas propuestas que se ubiquen en posiciones más bajas. Para ello, se utiliza el inverso de la posición ocupada por el hotel en la lista.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \quad (3.4)$$

### 3.3. Configuración experimental

Teniendo el conocimiento de la teoría de Análisis de Conceptos Formales (FCA, por sus siglas en inglés) y la propuesta presentada por (RecSys, 2019), se procederá a profundizar en la configuración y la toma de decisiones que se llevaron a cabo en el proyecto. En esta sección, se presentará en primer lugar la descripción de la infraestructura utilizada en el desarrollo. A continuación, se abordarán las dificultades computacionales que surgieron durante la generación del retículo de FCA. Finalmente, se expondrá la evolución del proyecto, los problemas que se encontraron y las decisiones que se tomaron para su resolución.

El objetivo principal del proyecto es comprobar la viabilidad del uso de FCA en la tarea propuesta. En lugar de centrarse en el desarrollo de un algoritmo de generación del retículo eficiente, se ha tomado la decisión de utilizar una librería existente en Java (Castellanos, Cigarrán, y Garcia-Serrano, 2021). La librería realiza la creación del retículo utilizando el Algoritmo de Próximos Vecinos, cuyo funcionamiento se detalla en la Figura 3.5. Es importante destacar que este algoritmo presenta una complejidad de  $\mathbf{O}(|C||G||M|^2)$ , donde  $|C|$  es la cantidad de conceptos formales,  $|G|$  es la cantidad de objetos y  $|M|$  es la cantidad de atributos. Si se desea obtener información adicional sobre este algoritmo, sus problemas de complejidad y otros algoritmos relacionados, se recomienda consultar (Carpineto y Romano, 2004b). Es importante destacar que, si bien la librería en Java existente resulta útil para la tarea propuesta y la generación del retículo de FCA, presenta algunos problemas computacionales que provocan largos tiempos de procesamiento y que podrían mejorarse en trabajos futuros.

- Java no es un lenguaje específicamente diseñado para el procesamiento matemático y, por lo tanto, puede presentar cierta ineficiencia en cálculos de esta naturaleza.
- La librería no ha sido desarrollada con la paralelización en mente, lo que puede resultar en una baja velocidad de procesamiento del algoritmo.



---

**Algorithm 2** Next Neighbours

---

**Input:** Context  $(G, M, I)$   
**Output:** The Concept Lattice =  $(C, E)$  of  $(G, M, I)$

```

1:  $C := (G, G')$ 
2:  $E := \emptyset$ 
3:  $currentLevel := (G, G')$ 
4: while  $currentLevel := \emptyset$  do
5:    $nextLevel := \emptyset$ 
6:   for each  $(X, Y) \in currentLevel$  do
7:      $lowerNeighbours := FindLowerNeighbours((X, Y))$ 
8:     for each  $(X_1, Y_1) \in LowerNeighbours$  do
9:       if  $(X_1, Y_1) \notin C$  then
10:         $C := C \cup \{(X_1, Y_1)\}$ 
11:         $nextLevel := nextLevel \cup \{(X_1, Y_1)\}$ 
12:       end if
13:       Add edge  $(X_1, Y_1) \rightarrow (X, Y)$  to  $E$ 
14:     end for
15:   end for
16:    $currentLevel := nextLevel$ 
17: end while
18:
19: function  $FindLowerNeighbours((X, Y))$ 
20: /* Returns the lower neighbours of a concept */
21:  $candidates := \emptyset$ 
22: for each  $m \in M \setminus Y$  do
23:    $X_1 := (Y \cup \{m\})'$ 
24:    $Y_1 := X_1'$ 
25:   if  $(X_1, Y_1) \notin candidates$  then
26:     $candidates := candidates \cup \{(X_1, Y_1)\}$ 
27:   end if
28: end for
29: return maximally general  $candidates$ 

```

---

Figura 3.5: Algoritmo Próximos vecinos

- La librería no utiliza la GPU, lo que podría aumentar significativamente el rendimiento de los cálculos.
- Todos los cálculos se realizan en la memoria del dispositivo, lo que puede provocar errores de memoria si se manejan retículos muy grandes.

En el procesamiento del conjunto de datos se ha seleccionado el lenguaje Python debido a su amplia variedad de librerías especializadas en el tratamiento de conjuntos de datos, como sklearn y pandas, y por la facilidad de realizar pruebas mediante notebooks con Jupyter durante las fases iniciales de estudio del conjunto de datos. Además, las métricas de evaluación proporcionadas por (RecSys, 2019) están desarrolladas en Python, lo que justifica aún más esta elección. Por lo tanto, se decidió que el desarrollo del proyecto se realizaría en Python, y desde él se llamaría a un pequeño desarrollo que utiliza la librería de Java para generar el retículo de FCA. Sin embargo, esto plantea el problema de cómo compartir el retículo generado por la librería Java para su uso en Python. Para solucionar este problema, se ha utilizado una base de datos basada en grafos llamada Neo4j<sup>3</sup>.

Estas complicaciones han provocado distintos problemas y errores que han ido condicionando el desarrollo de la solución a la hora de generar el retículo. De hecho, se han llevado a cabo varias pruebas y se ha experimentado con distintas configuraciones de la infraestructura, lo que ha supuesto una importante carga de trabajo.

La propuesta inicial es generar un retículo que relacione los usuarios, considerando **250852 usuarios**, con las características de los hoteles encontrados en los metadatos proporcionados en el conjunto de datos de (RecSys, 2019). Estas características incluyen información como la existencia de wifi o la valoración en estrellas del hotel, y se usan como atributos del retículo. El objetivo es crear un retículo que refleje los intereses de cada usuario en relación a las características de los hoteles para poder generar un ranking de hoteles ordenados por la presencia de estas características. Sin embargo, debido a que el número de características es elevado, específicamente **157 características**, se genera el problema de memoria al intentar generar un retículo con todos los usuarios.

La siguiente iteración descarta las características encontradas en los metadatos para hacer un retículo únicamente con los **usuarios como objetos**

---

<sup>3</sup>Página web oficial: <https://neo4j.com/>

**y los hoteles como atributos**, pero esta solución no mejora el resultado dado que hay más hoteles (927142 hoteles) que características en los metadatos (157 características). Por ello, en la siguiente iteración se decide usar únicamente los hoteles encontrados en la lista de hoteles mostrados al usuario, el campo "impressions", para reducir el número de atributos a 25 pero tampoco se consigue un retículo manejable.

Dado que no se encuentra la manera de reducir la lista de atributos aún más, se decide descartar la idea de generar un retículo genérico que contenga toda la información presente en el conjunto de datos. En su lugar, se opta por generar retículos pequeños ad hoc para cada caso, con los datos de los usuarios estrictamente relevantes para el ranking que se va a generar. Estos usuarios seleccionados serán aquellos que hayan elegido en algún momento algún hotel presente en la lista de atributos, que en este momento se reduce a 25 hoteles propuestos al usuario. De esta forma se crea un retículo por cada usuario para poder realizar la predicción. Aunque el tiempo de creación de estos retículos es asequible, el problema en este caso es el número de usuarios, que asciende a **250852**. Esto significa que se tendrán que generar **250852 retículos distintos** para poder llevar a cabo el experimento completo y compararse con los demás participantes. Con todo esto, se ha llegado a la conclusión de que el tiempo necesario para generar la cantidad de retículos requeridos es inviable, por lo que se ha tomado la decisión de **reducir el número de usuarios a utilizar en el experimento a 1000**. Esta medida permitirá llevar a cabo el experimento en un tiempo razonable.

En esta última iteración se está creando un retículo con los 25 hoteles de la muestra como atributos y los usuarios que en algún momento han hecho clic en ellos como objeto. Se ha logrado hacerlo en un tiempo aceptable, pero se ha querido agregar más información para poder tener mejores perfiles de usuario. Por esa razón, se ha decidido incluir en la lista de atributos los hoteles en los que el usuario ha hecho clic en el pasado y que no estén presentes en la muestra actual. De esta manera, se pueden crear retículos con un número de atributos variable para cada usuario, que como mínimo puede ser de 25, y un número de objetos variable, ya que los objetos dependen de la lista de atributos. Esta última generación de retículos es factible y, por lo tanto, es la que se utilizará en el trabajo. Sin embargo, existe la posibilidad de que surjan problemas de memoria durante su ejecución, por lo que se han añadido sistemas de *checkpoint* cada vez que se crea un ranking

para un usuario, para que el experimento pueda ejecutarse después de un fallo y continuar desde el primer usuario sin procesar y descartar los ya procesados. Con estos cambios, se han obtenido retículos que, en promedio, están formados por **97 objetos, 106 atributos, 363 relaciones y 214 conceptos formales**. Entre todos los retículos que se han formado en el experimento, el más grande ha tenido **854 objetos, 767 atributos, 3333 relaciones y 2143 conceptos formales**.

Con las distintas iteraciones, se han tomado decisiones que nos han impedido compararnos con los demás participantes. Por lo tanto, se ha tomado la decisión de crear dos experimentos adicionales que no utilicen FCA y que actúen como una "línea base" para evaluar si FCA aporta valor:

- Aleatorio: Se realiza el ranking de los hoteles de manera aleatoria. Con esto queremos comprobar si la solución basada en FCA es mejor que una solución puramente aleatoria.
- Popularidad: Se realiza el ranking contando el número de veces que aparece el hotel en el campo "reference" en el conjunto de datos. Con esto se pretende organizar por "Más visitados", un sistema muy simple que pretendemos mejorar.

Con todo lo mencionado anteriormente, se ha tomado la decisión de realizar varios experimentos para ordenar el ranking con diferentes datos extraídos del retículo. Con el objetivo de comparar los distintos experimentos, se ha añadido una nueva métrica, además de la ya existente en la tarea propuesta por (RecSys, 2019). Esta nueva medida de evaluación es el **Mean Average Precision (MAP)** que evalúa la presencia del hotel seleccionado por el usuario en las primeras posiciones del ranking. El número de posiciones a considerar se indica en el nombre, por ejemplo,  $mAP5$  evalúa la presencia del hotel seleccionado en los primeros 5 puestos del ranking. En este trabajo se usarán  $mAP1$ ,  $mAP5$ ,  $mAP10$ ,  $mAP15$  y  $mAP20$ .

$$mAP = \frac{1}{|Q|} \sum_{i=1}^{|Q|} AP_i \quad (3.5)$$

Y siendo:

$$AP = \begin{cases} \frac{1}{N} & | N = 1 \text{ en } mAP1, N = 5 \text{ en } mAP5... & \text{Si se encuentra presente} \\ 0 & & \text{Otherwise} \end{cases} \quad (3.6)$$

En los distintos *mAP* los valores máximos que se pueden obtener son respectivamente: 1, 0.20, 0.1, 0.067, 0.05.

En resumen se han tomado las siguientes decisiones:

- Se ha decidido trabajar con un subconjunto aleatorio de usuarios en lugar del conjunto completo de datos. Para permitir la comparación entre los distintos experimentos, **se utilizará el mismo conjunto de 1000 usuarios para todos ellos.**
- Se va a centrar la formación de los retículos en relacionar los hoteles y los usuario, descartando la información recogida en el conjunto de metadatos.
- En lugar de crear un retículo que contenga toda la información de los conjuntos de datos, se ha optado por generar **retículos más pequeños y específicos para cada predicción.** De esta manera, se facilita la implementación de un sistema de *checkpoints* que permita continuar la ejecución del experimento en caso de que se produzca algún fallo, sin tener que reiniciar desde el principio.
- Al utilizar un subconjunto aleatorio, consideramos que la comparación entre los participantes y esta solución no es totalmente válida. Por lo tanto, se ha decidido llevar a cabo distintos experimentos que implican diversas formas de ordenar un ranking. Además, se ha agregado la evaluación *mAP* a la ya propuesta por (RecSys, 2019) (*MRR*) para comparar los distintos experimentos entre sí.

### 3.4. Experimentos y Resultados

En la presente sección se detallará el preprocesamiento realizado a los datos proporcionados por (RecSys, 2019), así como las diferentes fases de experimentación llevadas a cabo. En primer lugar, se explicará el proceso de preparación del retículo y posteriormente se presentará la fase de generación

del ranking. A continuación, se expondrán los resultados obtenidos de los distintos experimentos realizados. Finalmente, se llevará a cabo una discusión de los resultados generales obtenidos.

Es importante destacar que los experimentos toman como entrada el usuario y la lista de hoteles a ordenar para generar el ranking, además del conjunto de datos. Antes de iniciar el proceso de selección, se realiza un pequeño preprocesamiento para reducir el conjunto de datos.

### 3.4.1. Preprocesamiento y creación del retículo

El preprocesamiento esta formado por dos fases:

- Se eliminan del conjunto de datos los usuarios que **no contengan una entrada "clickout item" y con "reference" "null"**. Esta entrada es la que se utilizará para la recomendación y se espera que todos los usuarios la tengan. Sin embargo, es posible que algunos usuarios no tengan esta entrada por error o por la intención de aumentar aún más el conjunto de datos. En nuestros experimentos, estos usuarios no son necesarios y se eliminan.
- De los usuarios restantes, se seleccionan **1000 usuarios aleatoriamente**. Estos mismos 1000 usuarios se utilizarán en todos los experimentos para que puedan ser comparables entre sí.

Después de la preselección de usuarios, se procede a realizar los experimentos. El proceso de recomendación se divide en dos fases: en primer lugar, se lleva a cabo la **preparación del retículo**; en segundo lugar, se **genera el ranking final** ponderando los elementos, preparando la salida y evaluando los resultados. El proceso de creación del retículo es el mismo para todos los experimentos y consta de los siguientes pasos para cada usuario:

1. Se extrae la lista de "impressions" de la entrada del usuario que tiene como valor "reference" a "null". Esta lista de hoteles se añade a la lista de atributos  $M$  utilizada para formar el retículo.
2. Se busca en el histórico del usuario todos los elementos con los que ha interactuado y se añaden, si no estaban, a la lista de atributos  $M$ .
3. Se buscan los usuarios que en algún momento han interactuado con alguno de los elementos encontrados en la lista de atributos  $M$  y con

ellos y el usuario se forma el conjunto de objetos  $G$  y se prepara la tabla de relaciones entre ambos conjuntos  $I$ .

4. Se llama a la librería de Java para formar el retículo de conceptos formales con los datos generados previamente. Esta librería se encarga de guardar el retículo en Neo4j para que pueda ser accedido desde Python.
5. En este punto, se tiene el retículo del usuario y la lista de hoteles extraída del campo "impressions". Se intenta reducir esta lista basándose en el retículo formado, dado que si se eliminan hoteles del ranking y el usuario no ha clicado en ellos, la puntuación otorgada por las métricas es mayor ya que el tamaño del ranking penaliza. Para ello se siguen los siguientes pasos:
  - a) Se busca el **concepto objeto del usuario**, que es la representación más específica del usuario y puede representar su perfil.
  - b) Se extrae la intensidad del nodo, obteniendo la **lista de elementos con los que ha interactuado el usuario**. Es posible que en este punto haya elementos que no pertenezcan a las "impressions".
  - c) Si no se han obtenido 25 hoteles, se procede a recorrer el retículo buscando nodos **cada vez más específicos** y con más hoteles en su intensidad, con el objetivo de obtener hoteles de conceptos formales cercanos y, por lo tanto, similares, y añadirlos a la lista del ranking hasta llegar o superar los 25 hoteles.
  - d) Se eliminan los hoteles que no pertenecen a las "impressions". Debido a cómo se ha formado la lista de hoteles para el ranking, **es posible que algún hotel perteneciente al campo "impressions" no haya sido incluido porque pertenece a un nodo demasiado lejano respecto al concepto objeto del usuario**. Esto puede dar lugar a un ranking con menos de 25 elementos y puede haber eliminado hoteles pertenecientes al campo "impressions" que son menos afines al usuario.

### 3.4.2. Generación del ranking y evaluación

Para la segunda fase, la generación del ranking final, se han desarrollado cuatro enfoques diferentes que utilizan información extraída del retículo pa-

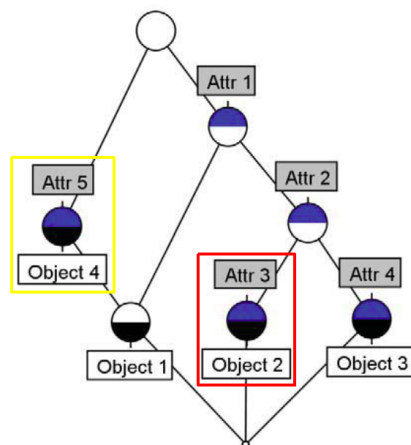


Figura 3.6: Recuerdo del ejemplo de diagrama de Hasse

ra puntuar los distintos hoteles. Todos los enfoques recorren uno por uno todos los hoteles de la lista de hoteles a evaluar y para cada uno calculan lo siguiente:

- **Popularidad:** Primero se busca el nodo correspondiente al **concepto atributo del hotel** en el retículo. Este nodo contiene en su extensión todos los usuarios que han accedido al menos una vez al hotel. La extensión no indica cuántas veces cada usuario ha accedido al hotel. Este método cuenta el **número de elementos que contiene la extensión** y lo asigna como valor de ponderación al hotel.

Por ejemplo, si consideramos el diagrama de Hasse que hemos visto en la explicación de FCA y ahora representado en la Figura 3.6, si el hotel a ponderar fuera el correspondiente al nodo *Attr5*, primero se buscaría el concepto atributo correspondiente, que es el nodo rodeado por el cuadrado amarillo. Su extensión contiene los objetos que se encuentran en el nodo y en nodos inferiores, es decir, la extensión contiene *Objeto1*, *Objeto4*. Como la extensión contiene dos objetos, este método asignaría un valor de ponderación igual a 2 al hotel.

- **Penalización por frecuencia de visita:** Esta ponderación utiliza un método que **penaliza la visita previa de los usuarios a un hotel y potencia la recomendación de hoteles nuevos**. La idea detrás de esta ponderación es que si un usuario está buscando hoteles para unas



vacaciones y ya ha visitado ciertos hoteles, es porque está interesado en explorar otros hoteles. Por lo tanto, parece razonable recomendar hoteles nuevos en lugar de aquellos que ya han sido visitados. Sin embargo, como se detallará en la sección de resultados, **este método produce resultados negativos**, lo que sugiere que es más probable que los usuarios vuelvan a visitar hoteles que ya han visitado antes.

Para determinar los hoteles que el usuario ha visitado, se busca su **concepto objeto** y se obtiene su intensidad. Sin embargo, el retículo no almacena la cantidad de veces que un usuario ha visitado un hotel. Para obtener esta información, se recupera el número de veces que el hotel aparece en las entradas de tipo "clickout item" del usuario desde el conjunto de datos directamente. Con ese número y usando como valor base la popularidad del hotel y calcula un nuevo valor mediante la siguiente operación:

$$ranking = \begin{cases} popularidad & num\ clicked = 0 \\ \frac{popularidad}{num\ clicked} & otherwise \end{cases} \quad (3.7)$$

A modo de ejemplo, supongamos que queremos evaluar el hotel *Attr2* y el usuario es *Object2*, utilizando el diagrama de Hasse que se muestra en la Figura 3.6. Primero, obtenemos el **concepto objeto** del usuario, que está representado por el nodo rodeado con un cuadrado rojo. Al examinar su intensidad, vemos que contiene *Attr1*, *Attr2*, *Attr3*, lo que significa que el usuario **ya ha visitado este hotel anteriormente, aunque no sabemos cuántas veces**. Para obtener esta información, consultamos el conjunto de datos y buscamos las entradas en las que el usuario hizo clic en el hotel *Attr2*. Supongamos que esto ocurrió 3 veces. Esto indica que el usuario ha visitado este hotel en 3 ocasiones anteriores. Además, necesitamos conocer la popularidad del hotel *Attr2*, que se obtiene yendo al **concepto atributo** inmediatamente más genérico al concepto objeto. En este caso, vemos que su extensión contiene 2 elementos, *Object2* y *Object3*. Por lo tanto, la ponderación final es  $\frac{2}{3} = 0,67$ .

Otro ejemplo interesante podría ser evaluar el hotel *Attr2* con el usuario *Object4*. En este caso, al examinar el **concepto objeto**, representado por el nodo rodeado con un cuadrado amarillo, vemos que su intensidad

solo contiene *Attr5*, lo que indica que el usuario nunca ha visitado este hotel antes. Por lo tanto, la **ponderación se basa en la popularidad del hotel**, que como vimos anteriormente, es de 2.

- **Bonificación por frecuencia de visita:** Este método tiene un enfoque opuesto al método anterior, ya que en lugar de penalizar, premia los hoteles que el usuario ha visitado previamente, y disminuye la recomendación de hoteles nuevos. Se creó debido a los resultados negativos obtenidos al utilizar la ponderación opuesta. Al igual que el método anterior, utiliza la popularidad como valor base a modificar, utilizando la siguiente función:

$$ranking = popularidad + num\ clicked \quad (3.8)$$

Para ilustrar el método descrito, consideremos el diagrama de Hasse presentado en la Figura 3.6, y supongamos que deseamos puntuar el hotel *Attr2* para el usuario *Object2*. En primer lugar, obtenemos el **concepto objeto** correspondiente al usuario, el cual está representado por el nodo rodeado por un cuadrado rojo. Al examinar su extensión, observamos que contiene los atributos *Attr1*, *Attr2*, y *Attr3*, lo que nos indica que el usuario ya ha visitado este hotel en ocasiones previas, aunque desconocemos la cantidad de veces. Para determinar la cantidad de visitas, debemos inspeccionar el conjunto de datos y buscar las entradas correspondientes al usuario que contengan *Attr2* en el campo "clickout item"; supongamos que encontramos tres de estas entradas, lo que significa que el usuario ha visitado el hotel en cuestión en tres ocasiones previas. Además, necesitamos conocer la popularidad de *Attr2*, lo cual podemos obtener yendo al **concepto atributo** inmediatamente superior al concepto objeto, donde observamos que su extensión contiene los elementos *Object2* y *Object3*. Por consiguiente, la ponderación final es de  $2 + 3 = 5$ .

Otro ejemplo que puede resultar interesante es si deseamos ponderar el mismo hotel *Attr2*, pero para el usuario *Object4*. En este caso, al examinar el **concepto objeto** correspondiente al usuario, representado por el nodo rodeado por un cuadrado amarillo, observamos que su extensión solo contiene el atributo *Attr5*, lo que nos indica que el usuario nunca ha visitado este hotel con anterioridad. Por lo tanto, la

ponderación correspondiente es simplemente la popularidad del hotel, que, como vimos anteriormente, es de 2.

- **Distancia con el concepto objeto:** En este experimento, se emplea la distancia entre el **concepto objeto del usuario** y el nodo más cercano que contiene el hotel. Se penalizan los hoteles que están más lejos, ya que se entiende que cuanto mayor sea la distancia, menos probable es que sean afines al perfil del usuario representado por el concepto objeto. Como valor base para la modificación, se utiliza la popularidad del hotel, y se aplica la siguiente función:

$$ranking = \begin{cases} popularidad & distancia = 0 \\ popularidad \times \frac{\alpha}{distancia} & otherwise \end{cases} \quad (3.9)$$

Siendo  $\alpha$  un valor de penalización cuyo valor en el experimento realizado es de 0,5 y la distancia el número de pasos necesarios para llegar desde el concepto objeto del usuario al nodo más cercano que contenga el hotel en su intensidad.

Por ejemplo, si se quiere calcular la ponderación del hotel *Attr2* para el usuario *Object4* en el diagrama de Hasse de la Figura 3.6, primero se obtiene el **concepto objeto** del usuario, que en este caso está representado por el nodo rodeado por un cuadrado amarillo. Como el hotel *Attr2* no aparece en la intensidad de este nodo, se recorre el retículo en busca de nodos más específicos y con nuevos hoteles. En la intensidad del siguiente nodo inferior se encuentran los atributos *Attr1* y *Attr5*, pero no el hotel buscado, por lo que se avanza otro nodo más. En la intensidad de este último nodo se encuentra el hotel *Attr2*, lo que significa que se ha necesitado de dos movimientos para llegar a él, por lo que la distancia es de 2. Además, se obtiene la popularidad del hotel mediante el **concepto atributo**, que es el nodo inmediatamente superior al nodo representado por un cuadrado rojo y que contiene los objetos *Object2* y *Object3*. La ponderación final se calcula como  $2 * \frac{0,5}{2} = 0,5$ .

Otro ejemplo interesante es si se quiere calcular la ponderación del mismo hotel *Attr2*, pero para el usuario *Object2*. En este caso, el **concepto objeto** del usuario, representado por un nodo rodeado por un cuadrado rojo, contiene el hotel en su intensidad, por lo que la ponderación es simplemente la popularidad del hotel, que es 2.

Posteriormente, se genera una salida a partir de un ranking formado por distintas combinaciones de las ponderaciones expuestas anteriormente. Esta salida se añade al final de un archivo csv que sigue la definición establecida por (RecSys, 2019), el cual contiene los siguientes campos:

- `user_id`: identificador del usuario que realizó el clic.
- `session_id`: identificador de la sesión en la que se realizó el clic.
- `timestamp`: marca de tiempo UNIX del clic.
- `step`: paso en la secuencia de acciones que indica el clic.
- `item_recommendations`: lista de artículos separados por espacios que es probable que el usuario haya hecho clic.

Una vez finalizado el proceso de generación de recomendaciones y agregados los resultados al archivo del usuario, se procede a eliminar el retículo de Neo4j para liberar recursos de disco. Sin embargo, en un entorno real, este retículo se mantendría almacenado para su reutilización posterior.

Cuando se obtiene un archivo con los rankings de todos los usuarios, se ejecutan los métodos de evaluación para obtener las puntuaciones. Los distintos experimentos buscan modificar esta segunda fase mediante diferentes combinaciones de métodos de ponderación. Es importante recordar que los experimentos se comparan con una línea base para evaluar los beneficios de FCA.

### 3.4.3. Línea Base

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473

Tabla 3.5: Resultados de la línea base

La línea base denominada **Aleatorio** consiste en generar un ranking aleatorio de los hoteles encontrados en el campo "impressions". Este experimento se utiliza como línea base para comparar los resultados de los demás experimentos y comprobar si aportan algún valor por encima de la aleatoriedad.

Los resultados obtenidos son los peores de todos los experimentos, con un valor de  $MRR$  de 0.1746. Es importante tener en cuenta que el valor máximo de  $MRR$  es 1, que se obtiene cuando el hotel seleccionado siempre está en primer lugar.

En cuanto a los resultados de  $mAP$ , se observa que el resultado de  $mAP1$  es muy alejado del máximo (que es 1), pero va mejorando gradualmente hasta llegar a un valor de  $mAP20$  de 0.043, que es cercano al valor máximo de 0.05. Esta mejora constante en los resultados puede deberse a la distribución aleatoria del ranking. Se pueden encontrar más detalles de los resultados en el Cuadro 3.5.

En la otra línea base llamada **Popularidad** se ha utilizado la popularidad de los hoteles como base para el ranking. Se ha ordenado el ranking según el número de veces que cada hotel aparece en el campo "reference" con la acción "clickout item" en el conjunto de datos, lo que indica la cantidad de veces que un usuario ha seleccionado ese hotel.

Los resultados de esta línea base se detallan en el Cuadro 3.5 y son superiores a los de la anterior, obteniendo un valor de  $MRR$  de 0.2937. Sin embargo, sigue estando alejado del resultado máximo posible.

Por otro lado, al observar los resultados de  $mAP$ , se puede apreciar que el valor obtenido en  $mAP1$  de 0.147 está todavía muy lejos del valor máximo de 1. A medida que se incrementa el número de elementos en el ranking, el valor de  $mAP$  aumenta gradualmente, llegando a ser bastante aproximado al valor máximo en  $mAP15$ , con un resultado de 0.0548 frente al valor máximo de 0.067. Esto podría indicar que esta línea base es capaz de colocar el hotel seleccionado por el usuario dentro de los primeros 15 elementos en la mayoría de los casos.

#### 3.4.4. Experimento 1: Popularidad FCA

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20	Mejora Aleatorio	Mejora popular
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043	N/A	N/A
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473	N/A	N/A
Experimento 1	0.2909	0.153	0.086	0.0625	0.05	0.0418	66.61 %	-0.95 %

Tabla 3.6: Resultados de experimento 1

La finalidad de este enfoque consiste en emplear exclusivamente el retículo para obtener el valor de popularidad del hotel, prescindiendo de cualquier información directa proveniente del conjunto de datos. Se espera que los resultados obtenidos sean **similares a los de la línea base de popularidad**.

Al examinar detalladamente los resultados del experimento presentados en el Cuadro 3.6, como era de esperar, se puede observar que el valor de *MRR* es similar, aunque algo menor, al de la línea base basada en popularidad, concretamente presenta una **disminución de un 0.95 %**. El *MRR* del experimento 1 es de 0.2909, mientras que el de la línea base es de 0.2937. Sin embargo, al analizar los distintos valores de *mAP*, se observan valores superiores en *mAP1* (0.153 en el experimento 1 frente a 0.157 en la línea base) y *mAP5* (0.086 en el experimento 1 frente a 0.0856 en la línea base), lo que indica una mayor precisión al posicionar el hotel seleccionado en las primeras posiciones.

Sin embargo, se puede apreciar una reducción en los valores a partir de *mAP10* (0.0625 en el experimento 1 frente a 0.0655 en la línea base), lo que sugiere que si este modelo no logra ubicar el hotel en una posición favorable, es más probable que lo coloque en posiciones peores. A pesar de estas diferencias marginales, se puede afirmar que **su comportamiento es equivalente al del experimento 1, tal como se esperaba**.

### 3.4.5. Experimento 2: Distancia

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20	Mejora Aleatorio	Mejora popular
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043	N/A	N/A
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473	N/A	N/A
Experimento 2	0.3465	0.204	0.0992	0.0656	0.0512	0.042	98.45 %	17.98 %

Tabla 3.7: Resultados de experimento 2

En este experimento, se utiliza como base **los resultados obtenidos en el experimento 1** y se aplica una ponderación adicional basada en la distancia. Esta ponderación penaliza la distancia entre el concepto objeto y el nodo más cercano donde aparece el hotel por primera vez, considerando que a mayor distancia, menor afinidad existe entre el hotel y el usuario.

Al analizar los resultados obtenidos, detallados en el Cuadro 3.7, se obser-

va un aumento significativo en el valor de  $MRR$ , **aumentando un 98.45 %** respecto a la línea base aleatoria y casi un **18 %** en la línea base de popularidad, incrementando en este último caso de 0.2909 a 0.3465.

Al examinar los valores de  $mAP$ , se constata una mejora notable en la precisión al asignar el hotel en posiciones superiores a la posición 5. El  $mAP1$  en el experimento 2 alcanza un valor de 0.204, en comparación con 0.147 en la línea base popular, lo que demuestra una mayor eficacia en colocar el hotel seleccionado en la posición más relevante. Asimismo, el  $mAP5$  muestra una mejora significativa, siendo de 0.0992 en el experimento 2 frente a 0.0856 en la línea base popular.

Sin embargo, los resultados en los siguientes valores de  $mAP$  son similares entre el experimento 2 y la línea base popular. Por ejemplo, el valor de  $mAP20$  en el experimento 2 es de 0.042, ligeramente inferior al valor de 0.0473 en la línea base. Esto sugiere que el modelo tiende a posicionar el hotel en posiciones superiores o inferiores, dejando las posiciones intermedias más libres.

En conclusión, el experimento 2 muestra una mejora significativa en el valor de  $MRR$  y una mayor precisión en la asignación del hotel seleccionado en las primeras posiciones, particularmente en  $mAP1$  y  $mAP5$ . No obstante, se observa una tendencia a evitar las posiciones intermedias en los resultados posteriores de  $mAP$ .

### 3.4.6. Experimento 3: Penalización por visita

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20	Mejora Aleatorio	Mejora popular
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043	N/A	N/A
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473	N/A	N/A
Experimento 3	0.2867	0.152	0.0838	0.0611	0.0497	0.0418	64.20 %	-2.38 %

Tabla 3.8: Resultados de experimento 3

En este experimento, se utilizan los resultados obtenidos en el experimento 1 y se aplica una ponderación adicional que penaliza las múltiples visitas del usuario. Esta ponderación se basa en el número de veces que el usuario ha visitado previamente el hotel, con el objetivo de reforzar la recomendación de hoteles nuevos.

Al analizar los resultados, detallados en el Cuadro 3.8, se observa una mejoría de un 64% respecto a la línea base aleatoria en cuanto a los valores de  $MRR$ . Sin embargo, sorprendentemente, en caso de la línea base popular, se observa un deterioro, **disminuyendo un 2.38%**, de 0.2937 obtenido en la línea base popular a 0.2867 en el experimento 3. Al examinar los distintos valores de  $mAP$ , se observa que son ligeramente mejores en las primeras posiciones. Por ejemplo, el  $mAP1$  en el experimento 3 es de 0.152 en comparación con 0.147 en la línea base popular. No obstante, esta mejora disminuye en el resto de posiciones, por ejemplo, en el  $mAP20$  baja de 0.0473 de la línea base popular a 0.0418.

En resumen, el experimento 3 muestra un deterioro en el  $MRR$  y una ligera mejora en la precisión de las primeras posiciones en comparación con la línea base popular. Sin embargo, en el resto de casos, este modelo parece empeorar en los resultados.

### 3.4.7. Experimento 4: Bonificación por visita

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20	Mejora Aleatorio	Mejora popular
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043	N/A	N/A
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473	N/A	N/A
Experimento 4	0.2965	0.155	0.088	0.0628	0.0503	0.0418	69.82%	0.95%

Tabla 3.9: Resultados de experimento 4

Este experimento surge como respuesta a los resultados insatisfactorios del experimento 3. Al igual que en ese experimento, en este caso se utilizan los resultados obtenidos en el experimento 1 y se aplica una ponderación adicional que bonifica las múltiples visitas del usuario. Recordemos que esta ponderación busca reforzar la recomendación de hoteles que han sido visitados previamente por el usuario, otorgándoles un mayor peso en los resultados.

Al analizar los resultados, detallados en el Cuadro 3.9, se confirma la expectativa de una mejora en los valores en comparación con el experimento 3. El  $MRR$  aumenta de 0.2937 obtenido en la línea base popular a 0.2965 en el experimento 4, teniendo una **mejora en el  $MRR$  del 0.95%**. Por otro lado, comparando con la línea base aleatoria se ha obtenido una mejora de un 69.82%. Al examinar los distintos valores de  $mAP$ , se observa una



ligera mejora en las primeras posiciones, siendo el  $mAP1$  de 0.155 en el experimento 4 frente a 0.147 en la línea base popular. Sin embargo, estos valores se invierten a medida que avanzamos en el ranking, llegando a ser peor el  $mAP20$ , con un valor de 0.0418 en el experimento 4 frente a 0.0473 en la línea base popular.

En resumen, el experimento 4 muestra una mejora en el  $MRR$  y una ligera mejora en las primeras posiciones en comparación con la línea base popular. Sin embargo, a medida que avanzamos en el ranking, los resultados se invierten, lo que sugiere que este modelo posiciona el hotel seleccionado en las primeras posiciones o en peores posiciones, dejando de lado las posiciones intermedias.

### 3.4.8. Experimento 5: Mixto distancia - Penalización por visita

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20	Mejora Aleatorio	Mejora popular
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043	N/A	N/A
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473	N/A	N/A
Experimento 5	0.339	0.201	0.096	0.0645	0.0509	0.0419	94.16 %	15.42 %

Tabla 3.10: Resultados de experimento 5

Este experimento tiene como objetivo investigar los resultados **agregar ponderaciones adicionales a la cadena**. Podemos considerar este experimento como el uso de los resultados obtenidos en el experimento 2 y la aplicación adicional de la ponderación de penalización por visitas. Dado los resultados insatisfactorios del experimento 3, se espera que este modelo experimente un empeoramiento en comparación con el experimento 2 y 3, pero superior a la línea base popular.

Al analizar los resultados, detallados en el Cuadro 3.10, se confirma la expectativa de una mejora en comparación con la línea base popular provocado por el filtro basado en distancia. El  $MRR$  aumenta de 0.2937 obtenido en la línea base popular a 0.339 en el experimento 5, obteniendo una **mejora del 15.42 %**. Por otro lado, se sigue obteniendo una gran mejora de un 94 % respecto a la línea base aleatoria.

Además, al observar los diferentes valores de  $mAP$ , se observa una me-

jora en las primeras posiciones, aumentando en el caso del  $mAP1$  de 0.147 en la línea base popular a 0.201. Sin embargo, para los posteriores  $mAP$  se experimenta un empeoramiento generalizado. Por ejemplo, en el caso de  $mAP15$ , se obtiene un valor de 0.0509 en el experimento 5 frente a 0.0548 en la línea base popular, lo que indica una polarización a la hora de situar el hotel seleccionado por el usuario en las primeras posiciones o en las últimas.

### 3.4.9. Experimento 6: Mixto distancia - Bonificación por visita

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20	Mejora Aleatorio	Mejora popular
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043	N/A	N/A
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473	N/A	N/A
Experimento 6	0.3528	0.215	0.1002	0.0654	0.0512	0.0419	102.06 %	20.12 %

Tabla 3.11: Resultados de experimento 6

Este experimento tiene como objetivo comprobar los resultados al agregar más ponderaciones a la cadena. Surge, al igual que el experimento 4, debido a los malos resultados obtenidos en los Experimentos 3 y 5. Podría considerarse que este experimento trabaja con los resultados obtenidos en el Experimento 2 y les aplica la ponderación de bonificación por visita. Se espera que este experimento muestre una mejora en comparación con los resultados del Experimento 2 y 4.

Al analizar los resultados, detallados en el Cuadro 3.11, se observa, como se esperaba, una mejora sustancial en comparación con las líneas base aleatoria y popular, con un aumento de **102.06 %** y **20.12 %** respectivamente. El valor de  $MRR$  aumenta de 0.2937 en la línea base popular a 0.3528.

Por otro lado, al observar los diferentes valores de  $mAP$ , se aprecia una mejora en las primeras posiciones, lo que indica una clasificación más alta del hotel en comparación con la línea base popular. Por ejemplo,  $mAP1$  ha aumentado de 0.147 a 0.215. Sin embargo, se observa un ligero deterioro en los valores de  $mAP$  del 10 al 20. Por ejemplo, en el  $mAP10$ , el valor disminuye de 0.0655 a 0.0654, y en el  $mAP20$ , el valor disminuye de 0.0473 a 0.0419. Esto podría indicar que este modelo tiende a posicionar el hotel seleccionado por el usuario en las primeras posiciones o relegarlo a las últimas

posiciones, dejando de lado las posiciones intermedias.

### 3.4.10. Discusión general y resultados finales

Experimento	MRR	MAP1	MAP5	MAP10	MAP15	MAP20	Mejora Aleatorio	Mejora popular
Base aleatorio	0.1746	0.05	0.0504	0.048	0.0452	0.043	N/A	N/A
Base popular	0.2937	0.147	0.0856	0.0655	0.0548	0.0473	N/A	N/A
Experimento 1	0.2909	0.153	0.086	0.0625	0.05	0.0418	66.61 %	-0.95 %
Experimento 2	0.3465	0.204	0.0992	0.0656	0.0512	0.042	98.45 %	17.98 %
Experimento 3	0.2867	0.152	0.0838	0.0611	0.0497	0.0418	64.20 %	-2.38 %
Experimento 4	0.2965	0.155	0.088	0.0628	0.0503	0.0418	69.82 %	0.95 %
Experimento 5	0.339	0.201	0.096	0.0645	0.0509	0.0419	94.16 %	15.42 %
Experimento 6	0.3528	0.215	0.1002	0.0654	0.0512	0.0419	102.06 %	20.12 %

Tabla 3.12: Resultados de los experimentos

A la hora de comparar los distintos experimentos, es importante tener en cuenta que la ponderación basada en popularidad se considera el valor de referencia sobre el cual se realizan modificaciones para reorganizar el ranking. Por lo tanto, el experimento 1, que utiliza únicamente esta ponderación, puede ser considerado como un experimento base, mientras que los demás experimentos se centran en mejorar estos resultados.

Si observamos los resultados del experimento 2 respecto al experimento 1, podemos ver que la nueva ponderación añade valor, aumentando de 0.2909 en el experimento 1 a 0.3465 en el experimento 2. Esta mejora indica una mayor capacidad del modelo para posicionar correctamente el hotel seleccionado por el usuario en las primeras posiciones del ranking. Por otro lado, al examinar los valores de  $mAP$ , se constata una mejora notable en la precisión al asignar el hotel en posiciones superiores a la posición 5. El  $mAP1$  en el experimento 2 alcanza un valor de 0.204, en comparación con 0.153 en el experimento 1, lo que demuestra una mayor eficacia en colocar el hotel seleccionado en la posición más relevante. Asimismo, el  $mAP5$  muestra una mejora significativa, siendo de 0.0992 en el experimento 2 frente a 0.086 en el experimento 1. Sin embargo, los resultados en los siguientes valores de  $mAP$  son similares entre el experimento 2 y el experimento 1. Por ejemplo, el valor de  $mAP20$  en el experimento 2 es de 0.042, ligeramente superior al valor de 0.0418 en el experimento 1. Esto sugiere que el modelo tiende a posicionar el

hotel en posiciones superiores o inferiores, dejando las posiciones intermedias más libres.

En conclusión, el experimento 2 muestra una mejora significativa en el valor de  $MRR$  y una mayor precisión en la asignación del hotel seleccionado en las primeras posiciones, particularmente en  $mAP1$  y  $mAP5$ . No obstante, se observa una tendencia a evitar las posiciones intermedias en los resultados posteriores de  $mAP$ .

Por otro lado, al analizar el experimento 3, se observa sorprendentemente un empeoramiento en los valores. El  $MRR$  disminuye de 0.2909 obtenido en el experimento 1 a 0.2867 en el experimento 3. Al examinar los distintos valores de  $mAP$ , se observa que son ligeramente peores en las primeras posiciones. Por ejemplo, el  $mAP5$  en el experimento 3 es de 0.0838 en comparación con 0.086 en el experimento 1. Sin embargo, estos valores se igualan en el  $mAP20$ , siendo 0.0418 en ambos casos. Esto indica que este modelo está posicionando los hoteles en posiciones inferiores, pero mantiene la misma precisión en las primeras 20 posiciones. Esto podría ser debido a que los cambios realizados en la ponderación no son suficientes para mover los hoteles de posición. En otras palabras, parece que la popularidad tiene más peso en el ranking que la penalización por visitas previas.

En resumen, el experimento 3 muestra un empeoramiento en el  $MRR$  y una ligera disminución en la precisión de las primeras posiciones en comparación con el experimento 1. Sin embargo, en las primeras 20 posiciones, ambos experimentos tienen resultados similares, lo que sugiere que la penalización por visitas previas no tiene un impacto significativo en el posicionamiento de los hoteles menos populares. **La popularidad sigue siendo el factor dominante en el ranking.**

Como se esperaba, al contrario que con el experimento 3, el experimento 4 confirma la expectativa de una mejora en los valores en comparación con el experimento 3. El  $MRR$  aumenta de 0.2909 obtenido en el experimento 1 a 0.2965 en el experimento 4. Al examinar los distintos valores de  $mAP$ , se observa una ligera mejora en las primeras posiciones, siendo el  $mAP1$  de 0.155 en el experimento 4 frente a 0.153 en el experimento 1. Sin embargo, estos valores se igualan a medida que avanzamos en el ranking, llegando a ser idénticos en el  $mAP20$ , con un valor de 0.0418 en ambos casos. Esto indica que este modelo está posicionando los hoteles en posiciones superiores, pero mantiene la misma precisión en las primeras 20 posiciones.

Parece que los experimentos 3 y 4 tienen un cambio limitado que puede deberse a que los valores calculados en la ponderación no son suficientes para modificar significativamente la posición de los hoteles. Parece que la popularidad sigue siendo el factor dominante en el ranking y tiene más peso que la penalización o bonificación por visitas previas.

Por último, nos encontramos con los experimentos 5 y 6, que utilizan los resultados obtenidos en el experimento 2 y aplican una ponderación adicional de penalización y bonificación por visitas, respectivamente. En el caso del experimento 5, se confirma la expectativa de un empeoramiento en comparación con el experimento 2. El  $MRR$  disminuye de 0.3465 obtenido en el experimento 2 a 0.339 en el experimento 5. Además, al observar los diferentes valores de  $mAP$ , se observa un deterioro generalizado desde  $mAP1$  hasta  $mAP15$  (siendo  $mAP1$  de 0.201 en el experimento 5 frente a 0.204 en el experimento 2, y en el caso de  $mAP15$ , 0.0509 en el experimento 5 frente a 0.0512 en el experimento 2), lo que indica una clasificación de los hoteles en posiciones más bajas en comparación con el experimento 2.

Por otro lado, en el caso del experimento 6, se observa, como se esperaba, una mejora en comparación con el experimento 2. El valor de  $MRR$  aumenta de 0.3465 en el experimento 2 a 0.3528. Y al observar los diferentes valores de  $mAP$ , se aprecia una mejora en las primeras posiciones, lo que indica una clasificación más alta del hotel en comparación con el experimento 2. Por ejemplo,  $mAP1$  ha aumentado de 0.204 a 0.215. Sin embargo, se observa un ligero deterioro en los valores de  $mAP$  del 10 al 20. Por ejemplo, en el  $mAP10$ , el valor disminuye de 0.0656 a 0.0654, y en el  $mAP20$ , el valor disminuye de 0.042 a 0.0419. Esto podría indicar que este modelo tiende a posicionar el hotel seleccionado por el usuario en las primeras posiciones o relegarlo a las últimas posiciones, dejando de lado las posiciones intermedias.

En general, los resultados obtenidos en los experimentos demuestran mejoras significativas en la efectividad de los métodos de recomendación utilizados en comparación con la línea base aleatoria. Estos hallazgos indican que los modelos implementados y la aplicación de FCA, aportan valor en el proceso de recomendación.

El mayor salto de mejora se ha observado al incluir la ponderación basada en la distancia entre nodos del retículo, lo cual ha generado mejoras sustanciales en todas las métricas evaluadas en los experimentos 2 y 6 en comparación con los demás métodos. Por otro lado, parece que considerar

el historial de visitas del usuario al generar el ranking, ya sea mediante penalización o bonificación, no ha tenido un impacto significativo en la mejora de los resultados. Por lo tanto, tener en cuenta esta propiedad específica del historial de visitas no parece ser determinante en el rendimiento de los modelos.

Entre todos los experimentos, los dos que han obtenido los mejores resultados son el experimento 2 y el 6, beneficiándose por la ponderación basada en la distancia. Es importante señalar que, en el caso de la tarea propuesta por (RecSys, 2019), el experimento 6 ha sido el que ha obtenido los mejores resultados al ser capaz de, en general, posicionar los hoteles seleccionados por el usuario en posiciones superiores.

En resumen, los resultados de los experimentos indican que la aplicación de la ponderación de FCA puede mejorar la calidad de las recomendaciones en comparación con métodos más simples como la popularidad o la aleatoriedad. Es interesante destacar que la ponderación basada en la distancia entre nodos del retículo ha sido la que ha aportado mayores mejoras, lo cual refuerza la utilidad de FCA en el contexto de la recomendación. Sin embargo, aún existen oportunidades para afinar y optimizar los enfoques de ponderación con el objetivo de lograr resultados aún mejores en términos de precisión y clasificación de las recomendaciones. Esta ponderación aprovecha una propiedad inherente de los retículos, lo cual valida la propuesta de utilizar FCA como método de recomendación.

## Capítulo 4

# Conclusiones y trabajo futuro

El objetivo principal de este estudio consistió en aplicar eficientemente el Análisis de Conceptos Formales en el contexto de recomendación, con el propósito de abordar la propuesta presentada en (RecSys, 2019). Nuestra propuesta incluyó seis posibles enfoques, los cuales demostraron la validez del FCA en este contexto. El experimento más destacado consistió en la utilización de tres ponderaciones (basadas en popularidad, bonificación por visitas y en la distancia entre nodos). Se trata del experimento 6, que mejoró los resultados respecto a la línea base en un 20.12% y utilizó diversas propiedades del retículo de conceptos formales, tales como la extensión del concepto atributo del hotel, la intención del concepto objeto del usuario y la distancia entre nodos, para asignar puntuaciones a los diferentes hoteles y generar un ranking. Estas aproximaciones nos permitieron alcanzar los siguientes objetivos de investigación:

Fue necesario adquirir un profundo conocimiento de la teoría del FCA, lo cual nos permitió comprender el retículo de conceptos formales y la información que proporcionaba, y así idear las distintas aproximaciones presentadas en este trabajo.

Se realizó un análisis exhaustivo del problema planteado y del conjunto de datos, lo que nos permitió diseñar la solución desarrollada.

Después de estudiar el conjunto de datos, se tomaron decisiones sobre qué datos utilizar en la construcción del retículo, intentando explotar la capacidad del FCA para encontrar relaciones entre los distintos datos.

Se diseñaron seis enfoques distintos que buscaban extraer información del retículo de conceptos formales generado, lo que nos permitió ordenar los hoteles en un ranking basado en el interés del usuario.

Se emplearon distintos métodos de evaluación para las aproximaciones, a fin de verificar su funcionamiento y utilidad.

Se compararon los diferentes resultados de las evaluaciones de las aproximaciones con dos líneas base para determinar su eficacia en comparación con modelos más básicos.

Se compararon las distintas propuestas analizando sus resultados para llegar a una conclusión sobre cuál era la más efectiva.

Además de los objetivos de investigación mencionados, también se cumplieron los objetivos técnicos siguientes:

Se identificaron los requisitos técnicos y organizativos necesarios para el estudio. Esto incluyó el estudio y uso de bibliotecas en Java para generar el retículo, así como el uso de bibliotecas de manejo de conjuntos de datos en Python, como Pandas. Además, se empleó una base de datos basada en grafos para almacenar temporalmente el grafo, garantizando su accesibilidad en ambos sistemas.

Se enfrentaron problemas de memoria relacionados con la generación del retículo y el manejo de datos del conjunto de datos. Para abordar esto, se realizó un preprocesamiento del conjunto de datos para reducir su tamaño, y se diseñó una solución con un sistema de checkpoints, lo que permitió reiniciar fácilmente la ejecución en caso de interrupciones o problemas.

A pesar de los resultados positivos obtenidos, se reconoce la existencia de áreas de mejora y oportunidades para futuras investigaciones en el ámbito del presente trabajo. En primer lugar, durante la creación del retículo, se encontraron dificultades que indican la posibilidad de mejorar el rendimiento de la librería de Análisis de Conceptos Formales utilizada. Se sugiere considerar posibles aproximaciones, como la migración a un lenguaje de programación más eficiente que Java, la paralelización de la implementación o el aprovechamiento de la potencia de cálculo de una unidad de procesamiento gráfico (GPU).

Otra línea de investigación prometedora puede consistir en explorar una aproximación que aborde el problema de rendimiento mediante la creación de un retículo con todos los datos presentes en el modelo de datos desde el inicio. Esta estrategia concentraría la tarea más exigente, la creación del retículo, al principio, evitando su creación para cada usuario y agilizando las predicciones subsiguientes. Además, se puede implementar un sistema de retroalimentación que permita que el retículo crezca con la adición de nuevos



usuarios. Sin embargo, es importante tener en cuenta que esta solución puede ser costosa en términos de memoria, por lo que no se ha implementado en el presente trabajo.

Aunque se optó por construir un retículo que relaciona a los usuarios y los hoteles en el presente documento, existen otras estructuras reticulares interesantes que pueden ser exploradas. Por ejemplo, en esta solución no se aprovechó la información disponible en el archivo de Metadatos proporcionado por los organizadores. Un retículo interesante puede ser aquel que relacione los hoteles con estos metadatos, lo que brindaría información sobre las características que interesan a un usuario.

Además, se pueden realizar pruebas con diferentes enfoques para recorrer el árbol y generar el ranking, así como con diferentes métodos de ponderación. Estas ponderaciones pueden no limitarse al uso exclusivo del FCA, ya que se puede aprovechar información externa o, incluso, las valoraciones encontradas en los metadatos para enriquecer el ranking. Por ejemplo, puede ser beneficioso explorar la combinación de análisis de conceptos formales con técnicas de aprendizaje automático y minería de datos para obtener recomendaciones aún más precisas y contextualizadas. Asimismo, sería interesante evaluar la solución propuesta en un entorno de producción y recopilar retroalimentación de los usuarios para evaluar su aceptación y efectividad a largo plazo.



# Bibliografía

## Bibliografía

- [Abel et al.2011] Abel, Fabian, Qi Gao, Geert-Jan Houben, y Ke Tao. 2011. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. En *Proceedings of the 3rd international web science conference*, páginas 1–8.
- [Abello2004] Abello, James. 2004. Hierarchical graph maps. *Comput. Graph.*, 28:345–359.
- [Adomavicius y Kwon2011] Adomavicius, Gediminas y YoungOk Kwon. 2011. Maximizing aggregate recommendation diversity: A graph-theoretic approach. En *ACM Conference on Recommender Systems*.
- [Adomavicius y Tuzhilin2005] Adomavicius, Gediminas y Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17:734–749.
- [Agarwal et al.2011] Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, y Rebecca J Passonneau. 2011. Sentiment analysis of twitter data. En *Proceedings of the workshop on language in social media (LSM 2011)*, páginas 30–38.
- [Agarwal, Chen, y Long2011] Agarwal, Deepak K., Bee-Chung Chen, y Bo Long. 2011. Localized factor models for multi-context recommendation. En *Knowledge Discovery and Data Mining*.
- [Aggarwal2016] Aggarwal, C.C. 2016. *Recommender Systems: The Textbook*. Springer International Publishing.

- [Aggarwal y Parthasarathy2001] Aggarwal, Charu C. y Srinivasan Parthasarathy. 2001. Mining massively incomplete data sets by conceptual reconstruction. En *Knowledge Discovery and Data Mining*.
- [Alam y Napoli2014] Alam, Mehwish y Amedeo Napoli. 2014. Lattice-based views over sparql query results. En *Proceedings of the 1st Workshop on Linked Data for Knowledge Discovery co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014)*.
- [Allan et al.1998] Allan, James, Jaime G Carbonell, George Doddington, Jonathan Yamron, y Yiming Yang. 1998. Topic detection and tracking pilot study final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, páginas 194–218.
- [AlSumait, Barbará, y Domeniconi2008] AlSumait, Loulwah, Daniel Barbará, y Carlotta Domeniconi. 2008. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. *2008 Eighth IEEE International Conference on Data Mining*, páginas 3–12.
- [Anta et al.2013] Anta, Antonio Fernández, Luis Núñez Chiroque, Philippe Morere, y Agustín Santos. 2013. Sentiment analysis and topic detection of spanish tweets: A comparative study of of nlp techniques. *Procesamiento del lenguaje natural*, 50:45–52.
- [Anthes2010] Anthes, Gary. 2010. Topic models vs. unstructured data. *Commun. ACM*, 53(12):16–18, dec.
- [Asmus et al.2014] Asmus, Josefine, Daniel Borchmann, Ivo F. Sbalzarini, y Dirk Walther. 2014. Towards an fca-based recommender system for black-box optimization. En *European Conference on Artificial Intelligence*.
- [Balabanovic y Shoham1997] Balabanovic, Marko y Yoav Shoham. 1997. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40:66–72.
- [Basilico y Hofmann2004] Basilico, Justin D. y Thomas Hofmann. 2004. Unifying collaborative and content-based filtering. *Proceedings of the twenty-first international conference on Machine learning*.

- [Becker, Naaman, y Gravano2011] Becker, Hila, Mor Naaman, y Luis Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. En *Proceedings of the international AAAI conference on web and social media*, volumen 5-1, páginas 438–441.
- [Bell y Koren2007] Bell, Robert M. y Yehuda Koren. 2007. Lessons from the netflix prize challenge. *SIGKDD Explor.*, 9:75–79.
- [Belohlavek2008] Belohlavek, RADIM. 2008. Introduction to formal concept analysis. *Olomouc, UPOL, Faculty of Science, Department of Computer Science*.
- [Bengel et al.2004] Bengel, Jason, Susan Gauch, Eera Mittur, y Rajan Vijayaraghavan. 2004. Chattrack: Chat room topic detection using classification. En *Intelligence and Security Informatics: Second Symposium on Intelligence and Security Informatics, ISI 2004, Tucson, AZ, USA, June 10-11, 2004. Proceedings 2*, páginas 266–277. Springer.
- [Bennett y Lanning2007] Bennett, James y Stan Lanning. 2007. The netflix prize. En *The Netflix Prize*.
- [Bidart et al.2014] Bidart, Ruhan, Adriano M. Pereira, Jussara M. Almeida, y Anísio Mendes Lacerda. 2014. Where should i go? city recommendation based on user communities. *2014 9th Latin American Web Congress*, páginas 50–58.
- [Billsus y Pazzani1998] Billsus, Daniel y Michael J. Pazzani. 1998. Learning collaborative information filters. En *International Conference on Machine Learning*.
- [Billsus y Pazzani2000] Billsus, Daniel y Michael J. Pazzani. 2000. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10:147–180.
- [Blei y MCAULIFFE2010] Blei, DM y JD MCAULIFFE. 2010. Supervised topic models [ol]. *ArXiv e-prints*.
- [Bobadilla et al.2013] Bobadilla, Jesús, Fernando Ortega, Antonio Hernando, y Abraham Gutiérrez. 2013. Recommender systems survey. *Knowl. Based Syst.*, 46:109–132.

- [Brand2005] Brand, Matthew. 2005. A random walks perspective on maximizing satisfaction and profit. En *SDM*.
- [Burke2002] Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370.
- [Burke2007] Burke, R. 2007. Hybrid web recommender systems. En *The Adaptive Web*.
- [Candillier, Meyer, y Fessant2008] Candillier, Laurent, Frank Meyer, y Françoise Fessant. 2008. Designing specific weighted similarity measures to improve collaborative filtering systems. En *Industrial Conference on Data Mining*.
- [Canny2002] Canny, John F. 2002. Collaborative filtering with privacy via factor analysis. En *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [Cantador2008] Cantador, Iván. 2008. Exploiting the conceptual space in hybrid recommender systems: a semantic-based approach. En *Exploiting the conceptual space in hybrid recommender systems: a semantic-based approach*.
- [Carpineto y Romano2004a] Carpineto, Claudio y Giovanni Romano. 2004a. Concept data analysis - theory and applications. En *Concept data analysis - theory and applications*.
- [Carpineto y Romano2004b] Carpineto, Claudio y Giovanni Romano. 2004b. *Concept data analysis: Theory and applications*. John Wiley & Sons.
- [Castellanos, Cigarrán, y Garcia-Serrano2021] Castellanos, Angel, Juan Cigarrán, y Ana Garcia-Serrano. 2021. Formal concept analysis for topic detection: a clustering quality experimental analysis. <https://edatos.consociomadrone.es/dataset.xhtml?persistentId=doi:10.21950/ML90I9>.
- [Castells et al.2014] Castells, Pablo, Jun Wang, Rubén Lara, y Dell Zhang. 2014. Introduction to the special issue on diversity and discovery in recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5:1 – 3.

- [Cataldi, Caro, y Schifanella2010] Cataldi, Mario, Luigi Di Caro, y Claudio Schifanella. 2010. Emerging topic detection on twitter based on temporal and social terms evaluation. En *MDMKDD '10*.
- [Chakrabarti et al.2008] Chakrabarti, Soumen, Rajiv Khanna, Uma Sawant, y Chiranjib Bhattacharyya. 2008. Structured learning for non-smooth ranking losses. En *Knowledge Discovery and Data Mining*.
- [Chen, Gan, y Song2013] Chen, Hong, Mingxin Gan, y Meng Zhao Song. 2013. A graph model for recommender systems. En *A Graph Model for Recommender Systems*.
- [Cheng, He, y Liu2014] Cheng, Dehua, Xinran He, y Yan Liu. 2014. Model selection for topic models via spectral decomposition. *arXiv: Machine Learning*.
- [Codocedo y Napoli2015] Codocedo, Victor y Amedeo Napoli. 2015. Formal concept analysis and information retrieval—a survey. En *International Conference on Formal Concept Analysis*, páginas 61–77. Springer.
- [Cordobés et al.2013] Cordobés, Héctor, Antonio Fernández Anta, Luis Felipe Núñez, Fernando Pérez, Teófilo Redondo, y Agustín Santos. 2013. Técnicas basadas en grafos para la categorización de tweets por tema. *Díaz Esteban et al.[57]*, páginas 160–166.
- [Cornejo et al.2019] Cornejo, María Eugenia, László T. Kóczy, Jesús Medina, y Antonio Eduardo De Barros Ruano. 2019. Recommendation solution for a locate-based social network via formal concept analysis. En *Trends in Mathematics and Computational Intelligence*, páginas 131–138.
- [Cramer et al.2008] Cramer, Henriette, Vanessa Evers, Satyan Ramlal, Maarten van Someren, Lloyd Rutledge, Natalia Stash, Lora Aroyo, y Bob J. Wielinga. 2008. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-Adapted Interaction*, 18:455–496.
- [Devi y Venkatesh2009] Devi, M. K. Kavitha y P. Venkatesh. 2009. An improved collaborative recommender system. *2009 First International Conference on Networks & Communications*, páginas 386–391.

- [Devooght, Kourtellis, y Mantrach2015] Devooght, Robin, Nicolas Kourtellis, y Amin Mantrach. 2015. Dynamic matrix factorization with priors on unknown values. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [Diaz-Aviles et al.2012] Diaz-Aviles, Ernesto, Lucas Drumond, Lars Schmidt-Thieme, y Wolfgang Nejdl. 2012. Real-time top-n recommendation in social streams. En *Proceedings of the sixth ACM conference on Recommender systems*, páginas 59–66.
- [du Boucher-Ryan y Bridge2005] du Boucher-Ryan, Patrick y Derek G. Bridge. 2005. Collaborative recommending using formal concept analysis. En *Knowledge-Based Systems*.
- [Eldén y Berr2008] Eldén, Lars y Michael W Berr. 2008. Matrix methods in data mining and pattern recognition. En *Fundamentals of algorithms*.
- [Fang y Zheng2009] Fang, Peici y Siyao Zheng. 2009. A research on fuzzy formal concept analysis based collaborative filtering recommendation system. *2009 Second International Symposium on Knowledge Acquisition and Modeling*, 3:352–355.
- [Fellbaum1998] Fellbaum, Christiane. 1998. Wordnet: An electronic lexical database: Bradford book.
- [Feng et al.2015] Feng, Shanshan, Xutao Li, Yifeng Zeng, G. Cong, Yeow Meng Chee, y Quan Yuan. 2015. Personalized ranking metric embedding for next new poi recommendation. En *International Joint Conference on Artificial Intelligence*.
- [Figuerola et al.2005] Figuerola, Carlos G., José Luis Alonso Berrocal, Ángel F. Zazo Rodríguez, y Emilio Rodríguez. 2005. Reina at the webclef task: Combining evidences and link analysis. En *Conference and Labs of the Evaluation Forum*.
- [Fiscus y Doddington2002] Fiscus, Jonathan G y George R Doddington. 2002. Topic detection and tracking evaluation overview. *Topic detection and tracking: event-based information organization*, páginas 17–31.
- [Fouss, Pirotte, y Saerens2005] Fouss, François, Alain Pirotte, y Marco Saerens. 2005. A novel way of computing similarities between nodes



- of a graph, with application to collaborative recommendation. *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, páginas 550–556.
- [Ganter y Obiedkov2016] Ganter, Bernhard y Sergei A. Obiedkov. 2016. Conceptual exploration. En *Springer Berlin Heidelberg*.
- [Ganter y Wille1998] Ganter, Bernhard y Rudolf Wille. 1998. Formal concept analysis: Mathematical foundations. En *Formal Concept Analysis: Mathematical Foundations*.
- [Ge, Delgado-Battenfeld, y Jannach2010] Ge, Mouzhi, Carla Delgado-Battenfeld, y D. Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. En *ACM Conference on Recommender Systems*.
- [Geng et al.2008] Geng, Liqiang, Larry Korba, Yunli Wang, Xin Wang, y Yonghua You. 2008. Finding topics in email using formal concept analysis and fuzzy membership functions. En *Canadian Conference on AI*.
- [Godin et al.2013] Godin, Frédéric, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, y Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. *Proceedings of the 22nd International Conference on World Wide Web*.
- [González et al.2016] González, Angel Castellanos, Ana García, Serrano Juan, y Cigarrán Recuero. 2016. Tesis doctoral 2016 recommender systems in social settings: Proposal, development and testing in real scenarios. doctoral programme in intelligent systems. En *TESIS DOCTORAL 2016 RECOMMENDER SYSTEMS IN SOCIAL SETTINGS: PROPOSAL, DEVELOPMENT AND TESTING IN REAL SCENARIOS. DOCTORAL PROGRAMME IN INTELLIGENT SYSTEMS*.
- [Gori y Pucci2007] Gori, Marco y Augusto Pucci. 2007. Itemrank: A random-walk based scoring algorithm for recommender engines. En *International Joint Conference on Artificial Intelligence*.
- [Gradmann2010] Gradmann, Stefan. 2010. Knowledge= information in context: on the importance of semantic contextualisation in europeana (delhi). En *ICDL 2010, Date: 2010/02/26-2010/02/26, Location: New Delhi*.

- [Guo et al.2013] Guo, Xin, Yang Xiang, Qian Chen, Zhenhua Huang, y Yongtao Hao. 2013. Lda-based online topic detection using tensor factorization. *Journal of Information Science*, 39:459 – 469.
- [Hernando et al.2013] Hernando, Antonio, Jesús Bobadilla, Fernando Ortega, y Abraham Gutiérrez. 2013. Trees for explaining recommendations made through collaborative filtering. *Inf. Sci.*, 239:1–17.
- [Hernando et al.2014] Hernando, Antonio, Ricardo Moya, Fernando Ortega, y Jesús Bobadilla. 2014. Hierarchical graph maps for visualization of collaborative recommender systems. *Journal of Information Science*, 40:106 – 97.
- [Hofmann2004] Hofmann, Thomas. 2004. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst.*, 22:89–115.
- [Honarkhah y Caers2010] Honarkhah, Mehrdad y Jef K. Caers. 2010. Stochastic simulation of patterns using distance-based pattern modeling. *Mathematical Geosciences*, 42:487–517.
- [Hu, Fang, y Liang2014] Hu, Zhengyin, Shu Fang, y Tian Liang. 2014. Empirical study of constructing a knowledge organization system of patent documents using topic modeling. *Scientometrics*, 100(3):787–799.
- [Huang et al.2012] Huang, Bo-Tsuang, Yan Yang, Amjad Mahmood, y Hongjun Wang. 2012. Microblog topic detection based on lda model and single-pass clustering. En *International Conference on Rough Sets and Current Trends in Computing*.
- [Huang y Bian2009] Huang, Yuxia y Ling Bian. 2009. A bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet. *Expert Syst. Appl.*, 36:933–943.
- [Huang y Bian2015] Huang, Yuxia y Ling Bian. 2015. Using ontologies and formal concept analysis to integrate heterogeneous tourism information. *IEEE Transactions on Emerging Topics in Computing*, 3:172–184.
- [Ignatov et al.2013] Ignatov, D., Alexandra Yu. Kaminskaya, Anastasya A. Bezzubtseva, Andrey V. Konstantinov, y Jonas Poelmans. 2013. Fca-based models and a prototype data analysis system for crowdsourcing platforms. En *International Conference on Conceptual Structures*.

- [Ignatov y Kuznetsov2009] Ignatov, D. y Sergei O. Kuznetsov. 2009. Concept-based recommendations for internet advertisement. *ArXiv*, abs/0906.4982.
- [Ignatov, Kuznetsov, y Poelmans2012] Ignatov, D., Sergei O. Kuznetsov, y Jonas Poelmans. 2012. Concept-based biclustering for internet advertisement. *2012 IEEE 12th International Conference on Data Mining Workshops*, páginas 123–130.
- [Indira et al.2021] Indira, K, S Karthiga, C. V. Nisha Angeline, C., y Santhiya. 2021. Parallel clarans algorithm for recommendation system in multi-cloud environment. En *Computer Networks and Inventive Communication Technologies*, páginas 461–472.
- [Iyyer et al.2014] Iyyer, Mohit, Jordan L. Boyd-Graber, Leonardo Max Battista Claudino, Richard Socher, y Hal Daumé. 2014. A neural network for factoid question answering over paragraphs. En *Conference on Empirical Methods in Natural Language Processing*.
- [Jain y Dhillon2013] Jain, Prateek y Inderjit S. Dhillon. 2013. Provable inductive matrix completion. *ArXiv*, abs/1306.0626.
- [Jannach et al.2010] Jannach, D., M. Zanker, A. Felfernig, y G. Friedrich. 2010. *Recommender Systems: An Introduction*. Cambridge University Press.
- [Ji y Shen2015] Ji, Ke y Hong Shen. 2015. Making recommendations from top-n user-item subgroups. *Neurocomputing*, 165:228–237.
- [Kashnitsky y Ignatov2015] Kashnitsky, Yury y D. Ignatov. 2015. Can fca-based recommender system suggest a proper classifier? *ArXiv*, abs/1504.05473.
- [Kim et al.2011] Kim, Heung-Nam, Inay Ha, Kee-Sung Lee, Geun-Sik Jo, y Abdulmotaleb El Saddik. 2011. Collaborative user modeling for enhanced content filtering in recommender systems. *Decis. Support Syst.*, 51:772–781.
- [Kim y Compton2004] Kim, Mihye y Paul Compton. 2004. Evolutionary document management and retrieval for specialized domains on the web. *International Journal of Human-Computer Studies*, 60(2):201–241.

- [Kompan y Bieliková2010] Kompan, Michal y Mária Bieliková. 2010. Content-based news recommendation. En *International Conference on Electronic Commerce and Web Technologies*.
- [Konstan et al.2006] Konstan, Joseph A., Sean M. McNee, Cai-Nicolas Ziegler, Roberto Torres, Nishikant Kapoor, y John Riedl. 2006. Lessons on applying automated recommender systems to information-seeking tasks. En *AAAI Conference on Artificial Intelligence*.
- [Koren2008] Koren, Yehuda. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. En *Knowledge Discovery and Data Mining*.
- [Koren2009a] Koren, Yehuda. 2009a. The bellkor solution to the netflix grand prize. En *The BellKor Solution to the Netflix Grand Prize*.
- [Koren2009b] Koren, Yehuda. 2009b. Collaborative filtering with temporal dynamics. En *Knowledge Discovery and Data Mining*.
- [Koren y Bell2011] Koren, Yehuda y Robert M. Bell. 2011. Advances in collaborative filtering. En *Recommender Systems Handbook*.
- [Kumaran y Allan2004] Kumaran, Giridhar y James Allan. 2004. Text classification and named entities for new event detection. En *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, páginas 297–304.
- [Loboreiro et al.2010] Loboreiro, Gustavo, Luís Sarmiento, Jorge Teixeira, y Eugénio Oliveira. 2010. Tokenizing micro-blogging messages using a text classification approach. En *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, páginas 81–88.
- [Lau et al.2007] Lau, Raymond Y. K., Albert Y. K. Chung, Dawei Song, y Qiang Huang. 2007. Towards fuzzy domain ontology based concept map generation for e-learning. En *International Conference on Advances in Web-Based Learning*.
- [Leskovec, Backstrom, y Kleinberg2009] Leskovec, Jure, Lars Backstrom, y Jon M. Kleinberg. 2009. Meme-tracking and the dynamics of the news cycle. En *Knowledge Discovery and Data Mining*.

- [Li y Murata2010] Li, Xiaohui y Tomohiro Murata. 2010. A knowledge-based recommendation model utilizing formal concept analysis and association. *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, 4:221–226.
- [Linden, Smith, y York2003] Linden, Greg, Brent Smith, y Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.*, 7:76–80.
- [Lops, Degemmis, y Semeraro2011] Lops, Pasquale, Marco Degemmis, y Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. En *Recommender Systems Handbook*.
- [Lu et al.2015] Lu, Jie, Dianshuang Wu, Mingsong Mao, Wei Wang, y Guangquan Zhang. 2015. Recommender system application developments: A survey. *Decis. Support Syst.*, 74:12–32.
- [Luo et al.2014] Luo, Yang, Boyi Xu, Hongming Cai, y Fenglin Bu. 2014. A hybrid user profile model for personalized recommender system with linked open data. *2014 Enterprise Systems Conference*, páginas 243–248.
- [Ma, King, y Lyu2007] Ma, Hao, Irwin King, y Michael R. Lyu. 2007. Effective missing data prediction for collaborative filtering. En *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [Maio et al.2012] Maio, Carmen De, Giuseppe Fenza, Matteo Gaeta, Vincenzo Loia, Francesco J. Orciuoli, y Sabrina Senatore. 2012. Rss-based e-learning recommendations exploiting fuzzy fca for knowledge modeling. *Appl. Soft Comput.*, 12:113–124.
- [Manuel y Recuero2008] Manuel, Juan y Cigarrán Recuero. 2008. Organización de resultados de búsqueda mediante análisis formal de conceptos. En *ORGANIZACIÓN DE RESULTADOS DE BÚSQUEDA MEDIANTE ANÁLISIS FORMAL DE CONCEPTOS*.
- [McAuley y Leskovec2013] McAuley, Julian y Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems*.

- [Mcauliffe y Blei2007] Mcauliffe, Jon y David Blei. 2007. Supervised topic models. *Advances in neural information processing systems*, 20.
- [McNee, Riedl, y Konstan2006] McNee, Sean M., John Riedl, y Joseph A. Konstan. 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. *CHI '06 Extended Abstracts on Human Factors in Computing Systems*.
- [Meijer, Frasincar, y Hogenboom2014] Meijer, Kevin, Flavius Frasincar, y Frederik Hogenboom. 2014. A semantic approach for extracting domain taxonomies from text. *Decision Support Systems*, 62:78–93.
- [Miyahara y Pazzani2000] Miyahara, Koji y Michael J. Pazzani. 2000. Collaborative filtering with the simple bayesian classifier. En *Pacific Rim International Conference on Artificial Intelligence*.
- [Mobasher et al.2007] Mobasher, Bamshad, R. Burke, Runa Bhaumik, y Chad Williams. 2007. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Techn.*, 7:23.
- [Mobasher, Jin, y Zhou2003] Mobasher, Bamshad, Xin Jin, y Yanzan Zhou. 2003. Semantically enhanced collaborative filtering on the web. En *European Web Mining Forum*.
- [Moore et al.2013] Moore, Joshua L., Shuo Chen, Douglas Turnbull, y Thorsten Joachims. 2013. Taste over time: The temporal dynamics of user preferences. En *International Society for Music Information Retrieval Conference*.
- [Nenova, Ignatov, y Konstantinov2013] Nenova, Elena, D. Ignatov, y Andrey V. Konstantinov. 2013. An fca-based boolean matrix factorisation for collaborative filtering. *ArXiv*, abs/1310.4366.
- [Ning y Karypis2012] Ning, Xia y George Karypis. 2012. Sparse linear methods with side information for top-n recommendations. En *ACM Conference on Recommender Systems*.
- [Noulas et al.2012] Noulas, Anastasios, Salvatore Scellato, Neal Lathia, y Cecilia Mascolo. 2012. A random walk around the city: New venue recommendation in location-based social networks. *2012 International Confe-*

- rence on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, páginas 144–153.
- [O’Sullivan, Smyth, y Wilson2004] O’Sullivan, Derry, Barry Smyth, y David C. Wilson. 2004. Preserving recommender accuracy and diversity in sparse datasets. En *Int. J. Artif. Intell. Tools*.
- [Park et al.2012] Park, Deuk Hee, Hyea-Kyeong Kim, Il Young Choi, y Jae Kyeong Kim. 2012. A literature review and classification of recommender systems research. *Expert Syst. Appl.*, 39:10059–10072.
- [Parra y Brusilovsky2010] Parra, Denis y Peter Brusilovsky. 2010. Improving collaborative filtering in social tagging systems for the recommendation of scientific articles. *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 1:136–142.
- [Paterek2007] Paterek, Arkadiusz. 2007. Improving regularized singular value decomposition for collaborative filtering. En *Improving regularized singular value decomposition for collaborative filtering*.
- [Pazzani1999] Pazzani, Michael J. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408.
- [Pazzani y Billsus1997] Pazzani, Michael J. y Daniel Billsus. 1997. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27:313–331.
- [Pazzani y Billsus2007] Pazzani, Michael J. y Daniel Billsus. 2007. Content-based recommendation systems. En *The Adaptive Web*.
- [Pereira y Hruschka2015] Pereira, André Luiz Vizine y Eduardo R. Hruschka. 2015. Simultaneous co-clustering and learning to address the cold start problem in recommender systems. *Knowl. Based Syst.*, 82:11–19.
- [Petkos et al.2014] Petkos, Georgios, Symeon Papadopoulos, Luca Maria Aiello, Ryan Skraba, y Yiannis Kompatsiaris. 2014. A soft frequent pattern mining approach for textual topic detection. En *Web Intelligence, Mining and Semantics*.

- [Phan, Nguyen, y Horiguchi2008] Phan, Xuan-Hieu, Le-Minh Nguyen, y Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. En *Proceedings of the 17th international conference on World Wide Web*, páginas 91–100.
- [Phelan et al.2011] Phelan, Owen, Kevin McCarthy, Mike Bennett, y Barry Smyth. 2011. Terms of a feather: Content-based news recommendation and discovery using twitter. En *European Conference on Information Retrieval*.
- [Pirasteh, Hwang, y Jung2015] Pirasteh, Parivash, Dosam Hwang, y Jason J. Jung. 2015. Exploiting matrix factorization to asymmetric user similarities in recommendation systems. *Knowl. Based Syst.*, 83:51–57.
- [Pla y Hurtado2013] Pla, Ferran y Lluís-F Hurtado. 2013. Elirf-upv en tass-2013: Análisis de sentimientos en twitter. En *XXIX Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN 2013)*. TASS, páginas 220–227.
- [Poelmans et al.2013] Poelmans, Jonas, Dmitry I Ignatov, Sergei O Kuznetsov, y Guido Dedene. 2013. Formal concept analysis in knowledge processing: A survey on applications. *Expert systems with applications*, 40(16):6538–6560.
- [Porcel et al.2012] Porcel, Carlos, Álvaro Tejeda-Lorente, María Ángeles Martínez, y Enrique Enrique Herrera-Viedma. 2012. A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. *Inf. Sci.*, 184:1–19.
- [Priss2000] Priss, Uta. 2000. Lattice-based information retrieval. *KO Knowledge Organization*, 27(3):132–142.
- [Quercia, Askham, y Crowcroft2012] Quercia, Daniele, Harry Askham, y Jon A. Crowcroft. 2012. Tweetlda: supervised topic classification and link prediction in twitter. En *Web Science Conference*.
- [Rahman y Chow2010] Rahman, MKM y Tommy WS Chow. 2010. Content-based hierarchical document organization using multi-layer hybrid network and tree-structured features. *Expert Systems with Applications*, 37(4):2874–2881.



- [Ramage, Dumais, y Liebling2010a] Ramage, Daniel, Susan Dumais, y Daniel J. Liebling. 2010a. Characterizing microblogs with topic models. En *Fourth international AAAI conference on weblogs and social media*.
- [Ramage, Dumais, y Liebling2010b] Ramage, Daniel, Susan T. Dumais, y Daniel J. Liebling. 2010b. Characterizing microblogs with topic models. *Proceedings of the International AAAI Conference on Web and Social Media*.
- [Ramage et al.2009] Ramage, Daniel, David Hall, Ramesh Nallapati, y Christopher D. Manning. 2009. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. En *Conference on Empirical Methods in Natural Language Processing*.
- [Rao2008] Rao, K. Nageswara. 2008. Application domain and functional classification of recommender systems—a survey. *DESIDOC Journal of Library & Information Technology*, 28:17–35.
- [RecSys2019] RecSys. 2019. Recsys challenge 2019. <https://www.recsyschallenge.com/2019/>.
- [Ren et al.2011] Ren, Yuyang, Yajun Du, Xiaoping Huang, y Yong Xu. 2011. Topic detection of news stories with formal concept analysis. En *Topic Detection of News Stories with Formal Concept Analysis*.
- [Resnick et al.1994] Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, y John Riedl. 1994. Grouplens: an open architecture for collaborative filtering of netnews. En *Conference on Computer Supported Cooperative Work*.
- [Ricci, Rokach, y Shapira2011] Ricci, Francesco, Lior Rokach, y Bracha Shapira. 2011. Introduction to recommender systems handbook. En *Recommender Systems Handbook*.
- [Salakhutdinov, Mnih, y Hinton2007] Salakhutdinov, Ruslan, Andriy Mnih, y Geoffrey E. Hinton. 2007. Restricted boltzmann machines for collaborative filtering. En *International Conference on Machine Learning*.
- [Sarwar et al.2000] Sarwar, Badrul Munir, George Karypis, Joseph A. Konstan, y John Riedl. 2000. Application of dimensionality reduction in

- recommender system - a case study. En *Application of Dimensionality Reduction in Recommender System - A Case Study*.
- [Sayyadi, Hurst, y Maykov2009] Sayyadi, Hassan, Matthew F. Hurst, y Aleksey Maykov. 2009. Event detection and tracking in social streams. *Proceedings of the International AAAI Conference on Web and Social Media*.
- [Schafer, Konstan, y Riedl2004] Schafer, J. Ben, Joseph A. Konstan, y John Riedl. 2004. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5:115–153.
- [Schein et al.2002] Schein, Andrew I., Alexandrin Popescul, Lyle H. Ungar, y David M. Pennock. 2002. Methods and metrics for cold-start recommendations. En *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [Sedhain et al.2015] Sedhain, Suvash, Aditya Krishna Menon, Scott Sanner, y Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. *Proceedings of the 24th International Conference on World Wide Web*.
- [Senatore y Pasi2013] Senatore, Sabrina y Gabriella Pasi. 2013. Lattice navigation for collaborative filtering by means of (fuzzy) formal concept analysis. En *ACM Symposium on Applied Computing*.
- [Shani y Gunawardana2011] Shani, Guy y Asela Gunawardana. 2011. Evaluating recommendation systems. En *Recommender Systems Handbook*.
- [Shaw, Xu, y Geva2010] Shaw, Gavin, Yue Xu, y Shlomo Geva. 2010. Using association rules to solve the cold-start problem in recommender systems. En *Pacific-Asia Conference on Knowledge Discovery and Data Mining*.
- [Shi2013] Shi, Lei. 2013. Trading-off among accuracy, similarity, diversity, and long-tail: a graph-based recommendation approach. *Proceedings of the 7th ACM conference on Recommender systems*.
- [Shoval, Maidel, y Shapira2008] Shoval, Peretz, Veronica Maidel, y Bracha Shapira. 2008. An ontology-content-based filtering method. En *AN ONTOLOGY-CONTENT-BASED FILTERING METHOD*.
- [Shyu et al.2005] Shyu, Mei-Ling, Choochart Haruechaiyasak, Shu-Ching Chen, y Na Zhao. 2005. Collaborative filtering by mining association

- rules from user access sequences. *International Workshop on Challenges in Web Information Retrieval and Integration*, páginas 128–135.
- [Simovici, Fomenky, y Kunz2012] Simovici, Dan A., Paul Fomenky, y Werner Kunz. 2012. Polarities, axialities and marketability of items. *Marketing Science eJournal*.
- [Singh y Gordon2008] Singh, Ajit Paul y Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. En *Knowledge Discovery and Data Mining*.
- [Smyth y Cotter2000] Smyth, Barry y Paul Cotter. 2000. A personalised tv listings service for the digital tv age. *Knowl. Based Syst.*, 13:53–59.
- [Sobhanam y Mariappan2013] Sobhanam, Hridya y Anitha Mariappan. 2013. Addressing cold start problem in recommender systems using association rules and clustering technique. *2013 International Conference on Computer Communication and Informatics*, páginas 1–5.
- [Socher y Fei-Fei2010] Socher, Richard y Li Fei-Fei. 2010. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, páginas 966–973.
- [Socher et al.2014] Socher, Richard, Andrej Karpathy, Quoc V. Le, Christopher D. Manning, y A. Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218.
- [Spertus, Sahami, y Buyukkokten2005] Spertus, Ellen, Mehran Sahami, y Orkut Buyukkokten. 2005. Evaluating similarity measures: a large-scale study in the orkut social network. En *Knowledge Discovery and Data Mining*.
- [Sriram et al.2010] Sriram, Bharath, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, y Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. En *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, páginas 841–842.

- [Srivastava y Salakhutdinov2012] Srivastava, Nitish y Ruslan Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. *J. Mach. Learn. Res.*, 15:2949–2980.
- [Su y Khoshgoftaar2009] Su, Xiaoyuan y Taghi M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Adv. Artif. Intell.*, 2009:421425:1–421425:19.
- [Takács et al.2009] Takács, Gábor, István Pilászy, Bottyán Németh, y Domonkos Tikk. 2009. Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.*, 10:623–656.
- [Tao et al.2012] Tao, Ke, Fabian Abel, Qi Gao, y Geert-Jan Houben. 2012. Tums: twitter-based user modeling service. En *Extended Semantic Web Conference*, páginas 269–283. Springer.
- [Teh et al.2006] Teh, Yee Whye, Michael I. Jordan, Matthew J. Beal, y David M. Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566 – 1581.
- [Tejeda-Lorente et al.2014] Tejeda-Lorente, Álvaro, Carlos Porcel, Eduardo Peis, Rosa Sanz, y Enrique Enrique Herrera-Viedma. 2014. A quality based recommender system to disseminate information in a university digital library. *Inf. Sci.*, 261:52–69.
- [Tian2006] Tian, Ye. 2006. *A system for automated content organization*. Case Western Reserve University.
- [Ticha et al.2014] Ticha, Sonia Ben, Azim Roussanaly, Anne Boyer, y Khaled Bsaïes. 2014. User semantic model for dependent attributes to enhance collaborative filtering. En *Conference on Web Information Systems and technologies*.
- [Valverde-Albacete y Peláez-Moreno2007] Valverde-Albacete, Francisco J y Carmen Peláez-Moreno. 2007. Galois connections between semimodules and applications in data mining. En *International Conference on Formal Concept Analysis*, páginas 181–196. Springer.
- [Valverde-Albacete y Peláez-Moreno2013] Valverde-Albacete, Francisco J. y Carmen Peláez-Moreno. 2013. Systems vs . methods : an analysis of the affordances of formal concept analysis for information retrieval? En

*Systems vs . Methods : an Analysis of the Affordances of Formal Concept Analysis for Information Retrieval?*

- [Villena-Román y Garcia-Morera2013] Villena-Román, Julio y Janine Garcia-Morera. 2013. Tass 2013—workshop on sentiment analysis at sepln 2013: An overview. En *Proceedings of the TASS workshop at SEPLN*, páginas 112–125.
- [Wang y Wang2014] Wang, Guangxin y Yinglin Wang. 2014. Probabilistic attribute mapping for cold-start recommendation. En *Probabilistic Attribute Mapping for Cold-Start Recommendation*.
- [Wang, Terrovitis, y Mamoulis2013] Wang, Hao, Manolis Terrovitis, y Nikos Mamoulis. 2013. Location recommendation in location-based social networks using user check-in data. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.
- [Wang, Wang, y Yeung2014] Wang, Hao, Naiyan Wang, y D. Y. Yeung. 2014. Collaborative deep learning for recommender systems. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [Watson y Weber2020] Watson, Ian y Rosina Weber. 2020. A user-centric evaluation to generate case-based explanations using formal concept analysis. *Proceedings*, páginas 195–210.
- [Wayne2000] Wayne, Charles L. 2000. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. En *LREC*.
- [Wille1992] Wille, Rudolf. 1992. Concept lattices and conceptual knowledge systems. *Computers & Mathematics With Applications*, 23:493–515.
- [Wille1995] Wille, Rudolf. 1995. The basic theorem of triadic concept analysis. *Order*, 12:149–158.
- [Wille2009] Wille, Rudolf. 2009. Restructuring lattice theory: An approach based on hierarchies of concepts. En *International Conference on Formal Concept Analysis*.

- [Wu et al.2013] Wu, Xiang, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, y Guoping Hu. 2013. Personalized next-song recommendation in online karaokes. *Proceedings of the 7th ACM conference on Recommender systems*.
- [Wu et al.2016a] Wu, Yao, Christopher DuBois, Alice X. Zheng, y Martin Ester. 2016a. Collaborative denoising auto-encoders for top-n recommender systems. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*.
- [Wu et al.2016b] Wu, Yao, Xudong Liu, Min Xie, Martin Ester, y Q. Yang. 2016b. Cccf: Improving collaborative filtering via scalable user-item co-clustering. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*.
- [Xu et al.2012] Xu, Bin, Jiajun Bu, Chun Chen, y Deng Cai. 2012. An exploration of improving collaborative recommender systems via user-item subgroups. *Proceedings of the 21st international conference on World Wide Web*.
- [Yu et al.2014] Yu, Hsiang-Fu, Cho-Jui Hsieh, Si Si, y Inderjit S. Dhillon. 2014. Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, 41:793–819.
- [Yuan et al.2013] Yuan, Quan, G. Cong, Zongyang Ma, Aixun Sun, y Nadia Magnenat-Thalman. 2013. Time-aware point-of-interest recommendation. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*.
- [Zhang et al.2006] Zhang, Sheng, Weihong Wang, James C. Ford, y Fillia Makedon. 2006. Learning from incomplete ratings using non-negative matrix factorization. En *SDM*.
- [Zhang, Callan, y Minka2002] Zhang, Yi, Jamie Callan, y Thomas P. Minka. 2002. Novelty and redundancy detection in adaptive filtering. En *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [Zhang et al.2013] Zhang, Yongfeng, Min Zhang, Yiqun Liu, y Shaoping Ma. 2013. Improve collaborative filtering through bordered block diagonal

form matrices. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*.

[Zheng, Yan, y Zhou2015] Zheng, Hai-Tao, Yang-Hui Yan, y Ying-Min Zhou. 2015. Graph-based hybrid recommendation using random walk and topic modeling. En *Asia-Pacific Web Conference*, páginas 573–585. Springer.

[Zhou, Hui, y Chang2004] Zhou, Baoyao, Siu Cheung Hui, y Kuiyu Chang. 2004. A formal concept analysis approach for web usage mining. En *Intelligent Information Processing*.

**Todo list**