

Universidad del Valle de Guatemala

Facultad de Ingeniería

Algoritmos y Estructura de Datos

Fase 1 de Proyecto

Sistema de Recomendaciones

Jose Ovando – 24701

Sergio Alvares – 24254

Ricardo Rodríguez - 18907

Sistema de Recomendaciones

Introducción

En un mundo saturado de contenido audiovisual, elegir una película o serie adecuada puede convertirse en una tarea abrumadora. Plataformas de streaming ofrecen catálogos extensos, pero sus algoritmos de recomendación muchas veces no reflejan fielmente los gustos del usuario, lo que conlleva a la pérdida de tiempo, desinterés o incluso frustración. Frente a esta situación, surge la necesidad de diseñar un sistema inteligente y personalizado de recomendaciones. Este proyecto propone el desarrollo de un motor de recomendaciones basado en grafos, donde se considere no solo el historial del usuario, sino también las valoraciones de terceros, géneros favoritos y similitudes con otros usuarios. A través de la implementación de técnicas de Design Thinking y el uso de Neo4j como base de datos, se espera construir un prototipo funcional que mejore la experiencia de entretenimiento de los usuarios universitarios.

Planteamiento del problema

Actualmente, las personas enfrentan dificultades para decidir qué película o serie ver, debido a la amplia oferta de contenido digital. Esta indecisión provoca una pérdida significativa de tiempo, especialmente cuando los algoritmos de recomendación tradicionales no consideran adecuadamente los gustos individuales ni la influencia de recomendaciones de terceros. Además, muchos sistemas son genéricos y no toman en cuenta relaciones contextuales como afinidades por géneros, actores, o comentarios de otros usuarios con intereses similares. Por lo tanto, se identifica la necesidad de crear un sistema que ofrezca recomendaciones relevantes, personalizadas y útiles para cada usuario, basándose en sus gustos, valoraciones previas y la experiencia de otros usuarios con perfiles similares.

Ideación

Para abordar esta problemática, se propone lo siguiente:

Diseño de un sistema de recomendaciones mediante grafos:

- Crear nodos que representen usuarios, películas y géneros.
- Establecer relaciones como “vio”, “le gusta”, “recomienda”, y “pertenece a”.
- Asignar pesos a las relaciones según el grado de preferencia o calificación.

Algoritmo de recomendación propuesto:

- Implementar un modelo híbrido que combine recomendaciones basadas en contenido (etiquetas, géneros) y recomendaciones colaborativas (similitudes entre usuarios).
- Utilizar técnicas como caminatas aleatorias, filtrado colaborativo o propagación de influencia en grafos para determinar sugerencias más personalizadas.

Uso de Design Thinking:

- Realizar entrevistas para entender los hábitos de consumo audiovisual.
- Definir necesidades concretas a partir del análisis de usuarios reales.
- Prototipar una interfaz sencilla y visualmente atractiva que muestre recomendaciones dinámicas.
- Evaluar la utilidad del sistema mediante pruebas con usuarios potenciales.

Base de datos en Neo4j:

- Utilizar esta tecnología para representar la red de relaciones entre usuarios y contenido.
- Consultar datos mediante Cypher para obtener las recomendaciones de forma eficiente y escalable.

Investigación

Las bases de datos basadas en grafos representan la información como una red de nodos (entidades) y relaciones (conexiones), lo que las hace especialmente útiles para modelar contextos en los que las relaciones son tan importantes como los datos mismos. A diferencia de las bases relacionales tradicionales, donde los datos se almacenan en tablas, en una base de grafos como Neo4j es posible navegar directamente entre elementos conectados, como un usuario que ha visto una película, que pertenece a un género, o que ha sido recomendada por otro usuario. Esto permite consultas más eficientes y expresivas, especialmente en escenarios como los sistemas de recomendación (Neo4j, 2024).

1. **Filtrado colaborativo:** este algoritmo sugiere elementos a un usuario basándose en lo que otros usuarios similares han visto o calificado. Es útil para encontrar patrones de comportamiento compartidos entre comunidades de usuarios. Por ejemplo, si varios usuarios que vieron *Pulp Fiction* también vieron *Reservoir Dogs*, el sistema podría sugerir esta última a un nuevo usuario que haya visto la primera (Dataconomy, 2015).
2. **Recomendación basada en contenido:** este enfoque sugiere elementos similares a los que un usuario ya ha consumido, basándose en atributos como el género, actores, etiquetas o director. En un grafo, esto se traduce en buscar películas que compartan relaciones o propiedades con las que ya le gustaron al usuario (Solusoft, 2020).
3. **Algoritmo de Louvain:** detecta comunidades dentro del grafo optimizando una medida llamada modularidad. Es ideal para agrupar usuarios con intereses similares y luego usar esas agrupaciones para hacer recomendaciones más precisas, por ejemplo, películas populares dentro de una comunidad específica ((Neo4j, 2024a).
4. **Algoritmo de Leiden:** una mejora del Louvain, este algoritmo asegura que las comunidades estén internamente bien conectadas. Es más robusto y preciso,

especialmente útil cuando se trabaja con datos complejos o muy densos ((Neo4j, 2024b).

5. **Label Propagation:** funciona propagando etiquetas entre nodos basándose en la mayoría de etiquetas de los vecinos. Es rápido y fácil de implementar, útil para detectar comunidades de forma dinámica, aunque menos preciso que Louvain o Leiden (Neo4j, 2024).
6. **Algoritmo Girvan-Newman:** identifica comunidades eliminando las conexiones más influyentes del grafo (aquellas con mayor centralidad de intermediación). Es eficaz para redes pequeñas, aunque no es tan eficiente en grafos grandes (Dgraph, 2021).
7. **Algoritmo PageRank (modificado):** originalmente diseñado para medir la importancia de páginas web, puede adaptarse para recomendar películas que estén conectadas con muchos nodos relevantes (usuarios, etiquetas, recomendaciones), otorgando mayor prioridad a los contenidos más "centrales" del grafo (Neo4j, 2024).
8. **Algoritmo A*:** aunque se usa comúnmente en búsqueda de rutas, en el contexto de grafos puede ser útil para encontrar las conexiones más cortas o significativas entre un usuario y contenidos que aún no ha explorado, considerando pesos o prioridades (DataCamp, 2024).
9. **Algoritmo Personalized PageRank:** es una versión personalizada del algoritmo PageRank que evalúa la importancia de los nodos desde la perspectiva de un nodo específico. A diferencia del PageRank tradicional, este algoritmo introduce un sesgo hacia ciertos nodos al reiniciar las caminatas aleatorias desde ellos, lo que permite generar recomendaciones personalizadas, identificar comunidades locales o priorizar resultados en búsquedas enfocadas. Es ampliamente utilizado en redes sociales, sistemas de recomendación y análisis de grafos dirigidos a intereses particulares (Memgraph, 2023; Neo4j, 2019).

Conclusiones

- El uso de bases de datos basadas en grafos permite modelar de forma eficiente las relaciones entre usuarios y contenidos, lo cual mejora la calidad de los sistemas de recomendación. Gracias a su estructura flexible, es posible representar gustos, etiquetas, valoraciones y comunidades de manera natural y directa.
- Los algoritmos como el filtrado colaborativo, las recomendaciones por contenido y la detección de comunidades (Louvain, Leiden, Label Propagation) aportan personalización y precisión al sistema. Además, el pseudocódigo facilita el diseño previo a la implementación, asegurando claridad en la lógica del sistema.
- En conjunto, los grafos y estos algoritmos ofrecen una base sólida para construir sistemas de recomendación adaptables, precisos y útiles para los usuarios.

Sitios de Referencia

- Dataconomy. (2015, 19 de febrero). An introduction to recommendation engines. <https://dataconomy.com/an-introduction-to-recommendation-engines/>
- DataCamp. (2024). Algoritmo A: guía completa. <https://www.datacamp.com/es/tutorial/a-star-algorithm>
- Dgraph. (2021, 24 de junio). Community detection algorithms. <https://dgraph.io/blog/post/community-detection-algorithms/>
- Neo4j. (2024). Graph algorithms for community detection & recommendations. <https://neo4j.com/blog/graph-data-science/graph-algorithms-community-detection-recommendations/>
- Neo4j. (2024). Guide: Build a recommendation engine. <https://neo4j.com/docs/getting-started/appendix/tutorials/guide-build-a-recommendation-engine/>
- Neo4j. (2024). Label Propagation algorithm. <https://neo4j.com/docs/graph-data-science/current/algorithms/community/>
- Solusoft. (2020). Sistemas de recomendación basados en grafos. <https://www.solusoft.es/innovacion/sistemas-recomendacion-basados-en-grafos>
- Stack Overflow. (2020, 14 de diciembre). Write a pseudo code for a graph algorithm. <https://stackoverflow.com/questions/65268543/write-a-pseudo-code-for-a-graph-algorithm>