Sistema de Recomendación Híbrido

Christopher Guerra Herrero, Amanda Cordero Lezcano, and Alfredo Nuño Oquendo

Facultad de Matemática y Computación, Universidad de La Habana, Cuba

Resumen En este proyecto se desarrolló un sistema híbrido de recomendación de películas utilizando un enfoque monolítico que combina filtrado colaborativo y filtrado basado en contenido. Se emplea el diseño *feature combination*, lo cual se aprovecha para proporcionar explicaciones en lenguaje natural sobre las recomendaciones. El sistema se alimenta de un repositorio público de películas de la Universidad Marta Abreu, y se complementa con un sistema de recuperación de películas.
Palabras clave: Sistemas de Recomendación, Filtrado Colaborativo, Filtrado Basado en Contenido, Feature Combination

1. Introducción

En la era actual, los sistemas de recomendación son esenciales para mejorar la experiencia del usuario en diversas plataformas, especialmente en el ámbito del entretenimiento. Este proyecto tiene como objetivo desarrollar un sistema híbrido de recomendación de películas que combine técnicas de filtrado colaborativo y basado en contenido, proporcionando así recomendaciones personalizadas a los usuarios. A lo largo de este informe, se detallarán los aspectos técnicos y la metodología utilizada para el desarrollo de este sistema.

2. Antecedentes y Estado del Arte

Los sistemas de recomendación tradicionales se dividen en tres categorías principales: filtrado colaborativo, filtrado basado en contenido y filtrado basado en conocimiento. Esta categorización es ampliamente aceptada en la literatura, siendo presentada de manera clara en la taxonomía de sistemas de recomendación de Robin Burke [2].

El *filtrado colaborativo* se basa en las preferencias de usuarios similares, utilizando técnicas como la similitud de vectores o la factorización de matrices para predecir las preferencias de un usuario basado en comportamientos de otros usuarios con intereses similares. Este enfoque tiene la ventaja de no requerir información explícita sobre los ítems, pero puede sufrir de problemas como la escasez de datos (*cold start problem*) y la falta de diversidad en las recomendaciones, aspectos detallados por Konstan y Riedl [4].

Por otro lado, el *filtrado basado en contenido* utiliza las características de los ítems para hacer recomendaciones. Esta técnica analiza atributos como el género,

el director o el año de estreno de una película, y compara estas características con las preferencias históricas del usuario. El filtrado basado en contenido es especialmente útil en contextos donde el historial de comportamiento del usuario es limitado o cuando se busca recomendar ítems que son nuevos en el sistema [5].

Finalmente, el *filtrado basado en conocimiento* utiliza reglas explícitas sobre las necesidades y preferencias del usuario, junto con el conocimiento del dominio, para realizar recomendaciones. Este enfoque es menos común que los anteriores, pero es especialmente útil en sistemas donde es crucial entender el contexto o en dominios donde las preferencias son altamente especializadas y no pueden inferirse fácilmente a partir de datos históricos [6].

Existen diferentes diseños para la implementación de sistemas híbridos de recomendación, cada uno con sus propias ventajas y desventajas [2]:

- *Diseño Monolítico.* En el diseño monolítico, las diferentes técnicas de recomendación (como el filtrado colaborativo y el basado en contenido) se integran en un solo modelo o sistema. Este enfoque es más sencillo de implementar y permite una integración estrecha entre las técnicas, lo que facilita la creación de recomendaciones que combinan múltiples fuentes de datos. Sin embargo, su principal desventaja es que puede ser menos flexible y más difícil de escalar a medida que aumenta la complejidad del sistema o el volumen de datos [2].
- *Diseño Paralelo.* El diseño paralelo emplea múltiples modelos de recomendación de manera independiente y luego combina sus resultados. Cada técnica opera de forma separada y sus salidas se fusionan para generar la recomendación final. Este enfoque ofrece mayor flexibilidad y permite que cada técnica funcione de manera óptima en su propio espacio. Además, facilita la adición o eliminación de técnicas sin afectar a las demás. No obstante, la combinación de resultados puede ser compleja y requiere un sistema adicional para ponderar o seleccionar las recomendaciones finales [7].
- *Diseño por Pipelines.* En el diseño por *pipelines*, las técnicas de recomendación se aplican en una secuencia, donde la salida de una técnica sirve como entrada para la siguiente. Este enfoque permite refinar las recomendaciones en cada paso, lo que puede resultar en una mayor precisión. Por ejemplo, un sistema puede primero aplicar filtrado colaborativo para identificar un conjunto de ítems recomendados y luego utilizar filtrado basado en contenido para ajustar esas recomendaciones según las características específicas de los ítems. La principal desventaja de este enfoque es que puede ser computacionalmente costoso y complicado de optimizar [1].

En este proyecto, se ha optado por un enfoque híbrido, combinando las técnicas de filtrado colaborativo y filtrado basado en contenido en un sistema monolítico utilizando el diseño de combinación de características (*feature combination*). Este diseño permite aprovechar las ventajas de ambos enfoques y mitigar sus desventajas. Además, la combinación de características permitió proporcionar explicaciones en lenguaje natural sobre las recomendaciones, lo que mejora la transparencia del sistema.

3. Metodología

En esta sección, se describe el enfoque metodológico adoptado para el desarrollo del sistema de recomendación híbrido de películas. Primero, se presenta el diseño conceptual del sistema, detallando cómo los distintos componentes trabajan en conjunto para proporcionar recomendaciones personalizadas y precisas a los usuarios. Posteriormente, se explican las técnicas matemáticas empleadas para evaluar similitudes y analizar las características de los elementos en la base de datos, lo que garantiza un funcionamiento eficaz y ajustado a las preferencias de cada usuario.

3.1. Diseño del Sistema

El diseño del sistema de recomendación híbrido se basa en una arquitectura monolítica, donde el frontend y el backend están integrados de manera cohesionada para proporcionar recomendaciones personalizadas. Esta estructura permite gestionar eficientemente el flujo de información entre ambos componentes y asegurar una respuesta rápida y precisa a las solicitudes de los usuarios.

El sistema combina dos de los enfoques principales de recomendación: el filtrado colaborativo y el filtrado basado en contenido. La integración de estos enfoques se realiza mediante un diseño de combinación de características (feature combination), en el que las características de los usuarios y las películas se fusionan en un espacio compartido, optimizando así las recomendaciones. Esta estrategia permite generar recomendaciones que son personalizadas y fundamentadas, y facilita además la generación de explicaciones en lenguaje natural, mejorando la transparencia del sistema y la experiencia del usuario.

3.2. Técnicas Matemáticas Utilizadas

En la implementación del sistema de recomendación, se han empleado varias técnicas matemáticas fundamentales para evaluar la similitud entre ítems, medir la relevancia de términos y gestionar la proximidad entre cadenas de texto. Estas técnicas son cruciales para asegurar la precisión y la eficacia del sistema de recomendación.

Similitud Coseno [1] La similitud coseno es una medida que cuantifica la semejanza entre dos vectores de características, calculando el coseno del ángulo entre ellos en un espacio multidimensional. Esta técnica es especialmente útil en sistemas de recomendación, donde se utiliza para comparar las preferencias de usuarios o las características de los ítems.

Dados dos vectores $\mathbf{A} = (A_1, A_2, \ldots, A_n)$ y $\mathbf{B} = (B_1, B_2, \ldots, B_n)$, que representan las características de dos ítems o usuarios, la similitud coseno $\mathbf{Sim}_{\cos}(\mathbf{A}, \mathbf{B})$ se define como:

[1] Disponible en https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html el 30 de septiembre de 2024

$$\mathbf{Sim}_{\cos}(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

El valor de la similitud coseno oscila entre -1 y 1, donde un valor de 1 indica que los vectores son idénticos (es decir, el ángulo entre ellos es 0 grados), un valor de 0 indica que son ortogonales (sin relación) y un valor de -1 indica que son diametralmente opuestos.

En el contexto del sistema de recomendación, la similitud coseno se emplea para identificar ítems similares a los que un usuario ha preferido en el pasado, o para encontrar usuarios con preferencias similares a las del usuario objetivo.

TF-IDF[2] El término TF-IDF (Term Frequency - Inverse Document Frequency) es una medida ampliamente utilizada en recuperación de información y minería de texto para evaluar la relevancia de una palabra en un documento dentro de un conjunto de documentos (corpus). En sistemas de recomendación basados en contenido, TF-IDF se emplea para asignar pesos a las características textuales, como títulos o descripciones, de los ítems, lo cual permite hacer recomendaciones más precisas.

La **Frecuencia de Término** (TF) de un término t en un documento d se define como la frecuencia con la que t aparece en d, en relación con el total de términos t' en el documento:

$$\mathbf{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

donde $f_{t,d}$ es el número de veces que el término t aparece en el documento d, y t' representa cualquier término dentro del mismo documento d.

La **Frecuencia Inversa de Documentos** (IDF) se define como:

$$\mathbf{IDF}(t, D) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

donde $|D|$ es el número total de documentos en el corpus y $|\{d \in D : t \in d\}|$ es el número de documentos en los que aparece el término t.

Estas fórmulas son ampliamente usadas en literatura especializada, como se describe en [**?**], y permiten identificar las palabras más relevantes para describir un ítem, ayudando al sistema a realizar recomendaciones basadas en las características textuales de los ítems.

Algoritmo de Levenshtein [3] El algoritmo de Levenshtein, también conocido como distancia de edición, mide la distancia entre dos cadenas de caracteres.

[2] Disponible en https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/ el 30 de septiembre de 2024

[3] Artículo de la Universidad de Santiago de Compostela disponible en http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_2142.pdf el 30 de septiembre de 2024

Específicamente, calcula el número mínimo de operaciones (inserciones, eliminaciones o sustituciones de un carácter) necesarias para transformar una cadena en otra. Este algoritmo es útil en sistemas de recomendación para gestionar errores tipográficos y encontrar coincidencias aproximadas entre los nombres de las películas.

Crear una matriz d de tamaño $(m+1) \times (n+1)$
for i desde 0 hasta m **do**
  $d[i][0] \leftarrow i$
end for
for j desde 0 hasta n **do**
  $d[0][j] \leftarrow j$
end for
for i desde 1 hasta m **do**
  for j desde 1 hasta n **do**
    if $A[i-1] = B[j-1]$ **then**
      $coste \leftarrow 0$
    else
      $coste \leftarrow 1$
    end if
    $d[i][j] \leftarrow \text{mín}(d[i-1][j] + 1,\ d[i][j-1] + 1,\ d[i-1][j-1] + coste)$
  end for
end for
Retornar $d[m][n]$

3.3. Funcionamiento del sistema

Se realizó un proceso de scraping para obtener la base de datos de películas desde un repositorio público de la Universidad Marta Abreu[4].

El sistema recomienda películas aleatoriamente a los usuarios (que aún no han calificado ninguna) y les permite calificarlas, lo que se utiliza para generar recomendaciones basadas en usuarios similares. Además, las películas se clasifican por características como género, director y año de producción para las recomendaciones basadas en contenido. A cada película se le es asignado un valor que depende de la significación que el sistema le otorga a la misma en cuanto a ambos filtrados. Por este valor son ordenadas para mostrarlas como recomendación al usuario, acompañadas de una breve explicación en lenguaje natural de las razones por las cuales es recomendada.

[4] Universidad Central "Marta Abreu"de Las Villas. Sitio visuales.uclv.cu. Disponible en: https://visuales.uclv.cu el 30 de septiembre de 2024.

4. Implementación

La implementación incluyó el desarrollo de varias funcionalidades clave, incluyendo el sistema de búsqueda en la base de datos por nombre, la gestión de usuarios y la integración del sistema de calificación con el motor de recomendaciones.

4.1. Flujo de trabajo

El sistema utiliza un conjunto de datos que contiene información detallada sobre las películas, como títulos, géneros, director y datos de la interacción de los usuarios (películas calificadas).

El sistema utiliza un enfoque híbrido que combina filtrado colaborativo y filtrado basado en contenido.

- **Filtrado colaborativo:** Recomienda ítems basados en patrones de comportamiento de usuarios similares. Para un usuario, se identifican otros usuarios con preferencias similares y se recomienda contenido que esos usuarios hayan visto o calificado positivamente.
- **Filtrado basado en contenido:** Recomienda ítems similares a aquellos con los que el usuario ya ha interactuado. El sistema analiza las características de las películas preferidas por el usuario y busca contenido similar.

Por cada uno de estos filtrados se le otorga a la película un valor que denominaremos *score*. El *score* otorgado a la película mediante filtrado colaborativo y el otorgado mediante filtrado basado en contenido se promedian dando lugar al *score* de la película para ese usuario. Por este valor se ordenan las recomendaciones que se muestran al usuario.

Para mayor transparencia del sistema se ofrece al usuario una breve explicación en lenguaje natural de las razones por las que esta película es recomendada. Para esto se comparan los *score* de ambos filtrados si alguno es significativamente mayor que el otro se explica al usuario:

- En caso del filtrado colaborativo, que varios usuarios que tienen gustos similares a los suyos han calificado bien esa película.
- En caso del filtrado basado en contenido, ;as características que tiene esa película que la hace similar a otras que el usuario ha calificado positivamente.

Si ninguno de los dos filtrados prima sobre el otro se ofrece una explicación combinada.

4.2. Limitaciones

A pesar de que el sistema de recomendación desarrollado es funcional y capaz de ofrecer recomendaciones personalizadas, presenta algunas limitaciones que pueden afectar su rendimiento y escalabilidad:

- **Rendimiento y velocidad de procesamiento:** El sistema utiliza operaciones intensivas de cálculo, especialmente en el módulo de filtrado colaborativo, donde se manejan múltiples matrices de similitud entre usuarios e ítems. Estas operaciones pueden volverse lentas a medida que el número de usuarios y películas crece, dado que el cálculo de similitudes y distancias en matrices grandes es computacionalmente costoso.

  Esta limitación afecta la rapidez con la que el sistema genera recomendaciones, especialmente cuando se deben recalcular las matrices para actualizar las recomendaciones. La carga de procesamiento puede ser optimizada en el futuro utilizando técnicas como la reducción dimensional o el uso de índices eficientes para búsquedas rápidas.
- **Actualización manual de la base de datos:** Actualmente, la base de datos del sistema no se actualiza automáticamente cuando se añaden nuevas películas al repositorio de la Universidad Central "Marta Abreu"de Las Villas (UCLV). Esto significa que, cada vez que se incorporan nuevas películas al repositorio, el sistema requiere una actualización manual de la base de datos para incluir este contenido. Esta falta de actualización automática puede resultar en recomendaciones menos precisas o desactualizadas, ya que los nuevos ítems no son considerados en las sugerencias.

4.3. Métricas

Para evaluar el sistema de recomendaciones de manera efectiva, hemos utilizado un conjunto de métricas tradicionales en el ámbito de los sistemas de recomendación: precisión, recall y F1-score. Sin embargo, en lugar de evaluar el sistema sobre recomendaciones en tiempo real, se decidió aplicar estas métricas usando solo el conjunto de entrenamiento.

Enfoque basado en el conjunto de entrenamiento

Dado que el sistema está diseñado para recomendar nuevas películas que el usuario aún no ha visto, tiene poco sentido evaluar las recomendaciones en un conjunto de prueba que incluya ítems ya conocidos por el usuario. Devolver al usuario películas ya vistas no sería útil ni cumpliría el objetivo de un sistema de recomendaciones efectivo. Por esta razón, se utiliza únicamente el conjunto de entrenamiento para generar recomendaciones y evaluar el sistema.

Justificación del enfoque

Usar el conjunto de entrenamiento para las métricas permite verificar que el sistema es capaz de reconocer patrones en las preferencias del usuario y seleccionar ítems similares a aquellos que el usuario ha valorado positivamente en el pasado. Esto es una forma indirecta de asegurar el buen funcionamiento de los métodos implementados, ya que una precisión alta en el conjunto de entrenamiento sugiere que el sistema entiende y responde adecuadamente a los gustos y preferencias del usuario. Este enfoque también evita la redundancia de sugerir ítems ya conocidos y enfoca el sistema en generar recomendaciones personalizadas que sean verdaderamente útiles y novedosas para el usuario.

5. Conclusiones y Trabajo Futuro

El proyecto presentado ha logrado cumplir con los objetivos propuestos, desarrollando un sistema híbrido de recomendación de películas que integra de manera efectiva técnicas de filtrado colaborativo y filtrado basado en contenido mediante un enfoque monolítico. La implementación del sistema ha demostrado ser funcional y capaz de ofrecer recomendaciones personalizadas con un nivel satisfactorio de precisión. Este éxito se debe en gran medida a la combinación de características (*feature combination*) que permite al sistema proporcionar recomendaciones basadas en la integración de múltiples fuentes de datos, mejorando así la experiencia del usuario.

No obstante, el desarrollo de este proyecto ha permitido identificar varias áreas de mejora y posibles extensiones que podrían incrementar significativamente la eficacia y eficiencia del sistema:

- **Implementación de técnicas paralelas:** Actualmente, el sistema utiliza un enfoque monolítico que, aunque efectivo, puede beneficiarse de la implementación de técnicas paralelas. Estas técnicas permitirían la ejecución simultánea de los filtrados utilizados en el sistema híbrido, lo que optimizaría el tiempo de procesamiento.
- **Mejora en la captura y limpieza de datos:** La calidad de las recomendaciones está directamente relacionada con la calidad de los datos utilizados. A lo largo del desarrollo, se observó que la captura y limpieza de los datos podrían mejorarse considerablemente. Específicamente, se podrían implementar procesos automatizados de limpieza de datos que eliminen inconsistencias, errores tipográficos y datos redundantes, lo que resultaría en un conjunto de datos más robusto y confiable.
- **Migración a bases de datos más robustas:** Si bien SQLite3 ha sido suficiente para la fase inicial del proyecto, una migración a bases de datos más robustas como MySQL o PostgreSQL sería beneficiosa para soportar un mayor volumen de datos y mejorar la escalabilidad del sistema. Estas bases de datos ofrecen mejores herramientas para la optimización de consultas, manejo de transacciones y soporte de operaciones concurrentes, aspectos críticos para un sistema de recomendación en producción.
- **Expansión del sistema de recomendaciones:** Se podría explorar la inclusión de técnicas más avanzadas de aprendizaje automático, como modelos basados en redes neuronales o técnicas de *deep learning*, como autoencoders, para capturar patrones más complejos en las preferencias de los usuarios. Estos modelos podrían complementarse con embeddings de contenido para mejorar la precisión y personalización de las recomendaciones.
- **Integración de nuevas fuentes de datos:** Para mejorar la personalización y relevancia de las recomendaciones, se podría considerar la integración de nuevas fuentes de datos, como información demográfica adicional de los usuarios (edad, ubicación geográfica, etc.), o datos contextuales como la hora del día o la estacionalidad, que podrían influir en las preferencias de visualización.

- **Aumentar la diversidad de recomendaciones:** Implementar un algoritmo de diversificación que evite recomendar ítems muy similares consecutivamente. Esto se puede lograr utilizando un modelo de penalización basado en la similitud de características, para que el sistema elija ítems menos similares en cada ciclo de recomendación.
- **Mejoras en la explicación de las recomendaciones:** Aunque el sistema actual proporciona explicaciones en lenguaje natural sobre las recomendaciones, futuras mejoras podrían enfocarse en hacer estas explicaciones más detalladas y transparentes.

La implementación de estas mejoras no solo fortalecería el sistema actual, sino que también lo posicionaría mejor para adaptarse a las demandas de un entorno de producción más complejo y dinámico.

Referencias

1. Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, 2010.
2. Robin Burke. "Hybrid Recommender Systems: Survey and Experiments." *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
3. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Ïtem-based Collaborative Filtering Recommendation Algorithms.Ïn *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295, 2001.
4. Joseph A. Konstan and John Riedl. Recommender Systems: From Algorithms to User Experience." *User Modeling and User-Adapted Interaction*, 22(1-2):101–123, 2012.
5. Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Çontent-based Recommender Systems: State of the Art and Trends.Ïn *Recommender Systems Handbook*, pp. 73–105, Springer, 2011.
6. Gediminas Adomavicius and Alexander Tuzhilin. "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(6):734–749, 2005.
7. Gediminas Adomavicius, Jesse Bockstedt, Shawn Curley, and Jingjing Zhang. Çontext-Aware Recommender Systems: A Review and Future Directions.Ïn *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 311–312, 2015.