# Sistema de recomendación

Carlos Mario Chang Jardinez C312
Ernesto Rousell Zurita C312
Carlos Manuel García Rodríguez C312

Facultad de Matemática y Computación
Universidad de La Habana. Cuba

**Resumen:** En este trabajo se presenta una propuesta de algoritmo de recomendación para una plataforma de streaming de películas, basado en técnicas híbridas. El cual combina técnicas de filtrado colaborativo, filtrado basado en contenido y un sistema que tiene en cuenta la secuencialidad de las recomendaciones. Se pretende hacer un análisis inicial sobre el estado del arte, analizando y valorando las técnicas mas recientes y efectivas. Para luego explicar el sistema propuesto: como funciona, como se implementó y por qué se hizo de esa manera, que resultados de efectividad obtuvimos, y que ventajas y deficiencias tiene con respecto a otros sistemas o implementaciones.

# 1 Introducción

A partir de hace algunos años el auge de la tecnología ha ocasionado nuevos problemas a los que la sociedad se ha tenido que enfrentar, y uno de ellos es la cantidad masiva de datos disponible digitalmente para cualquiera. La cantidad de información disponible que hay en internet para el consumo humano es demasiada. Tantas películas y videos que nunca podrías ver ni aunque dedicaras la vida a ello, tanta música que ni aunque varias personas trabajen juntas con el propósito de oirlas todas lograrían ni si quiera abarcar la punta del iceberg, tantos productos disponibles en el comercio electrónico... Y entonces empezamos a entrever el problema aquí, porque aunque es poco probable que alguien quiera consumir todo ese contenido: ¿Que pasa cuando lo que quieres consumir debes buscarlo entre esa montaña de información?. ¿Y si no sabes qué es lo que que te interesa buscar o consumir?. Pues de esta necesidad surgen los sistemas de recomendación, para ayudar a los usuarios a encontrar información posiblemente relevante para ellos, incluso aunque los usuarios no sepan lo que quieren.

## 1.1 Definición de un sistema de recomendación

Se abordarán ahora términos e ideas más formales del ámbito de los sistemas de recomendación. Primero, a lo que antes nos referíamos como información (películas, música, productos, etc) se les denominan items. Y tenemos un conjunto de usuarios, que pueden tener varios tipos de interacciones con los items. Por ejemplo, en el caso de una plataforma de streaming de películas, los usuarios pueden ver, comentar o valorar la película. Hay sistemas que incluso pueden llegar a recoger más datos: cuanto tiempo el usuario se demoró viendo el item, si lo ha visto repetidamente, si lo compartió, etc. Y claro cada interacción usuario-item podría tener diferentes niveles de importancia.

Podemos definir un sistema de recomendación de forma sencilla de la siguiente forma: un algoritmo que dado un usuario $u$ y los items $\{i_1, i_2, ...i_k\}$ con los que $u$ ha interactuado el algoritmo devuelve otra serie de items $\{j_1, j_2, ...j_m\}$ que podrían ser relevantes o interesantes para el usuario $u$.

## 1.2 Plantando el problema

Nuestro objetivo es trabajar sobre la base de un sistema de recomendación de películas, que es algo que se utiliza en plataformas como Netflix. El objetivo es brindarle recomendaciones personalizadas a los usuarios, de películas que podrían gustarles, basado en las películas que ya han visto. En secciones posteriores ahondaremos en los detalles del problema: que datos tenemos, tanto de los usuarios como de las películas, que técnicas se utilizaron y por qué, como se implementaron, etc.

## 2 Estado del arte

Ahora abordaremos que técnicas o estrategias se utilizan en el mundo actual para implementar los sistemas de recomendación. Primero debemos mencionar que las estrategias o métodos usados varian dependiendo de las características de los datos, el contexto en el que se está trabajando, y que enfoque quiere lograrse. El sistema para recomendar películas no debe ser necesariamente el mismo que para recomendar productos en Amazon. También abordaremos un poco sobre los principales problemas que surgen de los sistemas de recomendación y como se tratan de resolver, en mayor o menor medida.

### 2.1 Tipos de sistemas de recomendación

- **Filtrado basado en contenido:** es uno de las estrategias más sencillas basada en identificar items similares a partir de características similares, y recomendar items parecidos a los que han sido del agrado del usuario en el pasado. Algunas de las herramientas que se utilizan para implementar este tipo de estrategias son minería de textos para identificar las preferencias de los usuarios, Tf-idf para encontrar similitud entre los items, Naive Bayes, redes neuronales, entre otros.
- **Filtrado colaborativo:** Exiten dos tipos o podría decirse dos variantes de filtrado colaborativo.
  - **Basado en memoria:** los basados en memoria se dividen en basado en usuarios o basados en items y se apoyan de una matriz de usuarios por items con las valoraciones de cada usuario le ha dado a cada item. Se centra en tratar de predecir los espacios en blanco. Hay muchas formas de hacerlo pero la idea central es usar técnicas de agrupación como KNN, Pearson Correlation y Similitud de cosenos para agrupar a usuarios o items similares, tal que pueda recomendarle a un usuario lo que le pareció atractivo a los usuarios parecidos.
  - **Basado en modelos:** Los filtrados basados en modelos usan algoritmos un poco más complejos tales como SVD, PCA o algoritmos de clusterización.
- **Modelos secuenciales:** los modelos secuenciales son diferentes de los modelos tradicionales mencionados anteriormente, ya que no solo tratan de captar las preferencias generales de los usuarios sino que intentan encontrar ciertas dependencias entre los items comprados por un usuario a lo largo del tiempo. Estos tipos de modelos se subdividen tambieé en ramas:
  - **Modelos secuenciales tradicionales:** usan las ideas intuitivas para encontrar patrones secuenciales, entre estos modelos clásicos se encuentran los modelos basados en las cadenas de Markov o la minería de patrones.
  - **Modelos basados en factores latentes:** usan representaciones de los usuarios o items en espacios latentes, permitiendo captar dependencias más complejas entre los items. Se usan los métodos de máquinas de factorización y algoritmos de embeedings.

- **Modelos basados en redes neuronales profundas:** usan redes neuronales convolucionales o recurrentes. O modelos más avanzados como modelos de atención.

## 2.2 Modelos híbridos

Los modelos híbridos nacieron para suplir las deficiencias de cada sistema combinándolos y aprovechando sus puntos fuertes. Existen varias maneras de combinar los sistemas, las explicaremos a continuación:

- **Combinación Ponderada:** Se asigna un peso a cada uno de los sistemas y se combinan las predicciones basadas en estos pesos. Es un enfoque sencillo en el que las recomendaciones se generan como un promedio ponderado de los resultados de cada sistema.
- **Combinación Mixta:** Se generan recomendaciones de múltiples sistemas al mismo tiempo y se presentan juntas al usuario. Es útil cuando cada método tiene distintas características relevantes que se desean incluir en la misma interfaz.
- **Conmutación:** Se selecciona uno de los métodos de recomendación según el contexto o el rendimiento esperado. Se elige el sistema más adecuado para cada situación, lo que permite aprovechar los puntos fuertes de cada método.
- **Ensamblado:** Se combinan los sistemas de recomendación en una estructura jerárquica, donde un sistema proporciona recomendaciones y otro las refina o mejora. Generalmente se usa un sistema inicial y luego se aplican otros métodos para filtrar o clasificar.
- **Modelo Unificado:** Diferentes métodos se combinan y se integran en un único modelo, como puede ser un algoritmo de aprendizaje automático que utiliza características de distintos sistemas. Esto permite una integración profunda y un tratamiento uniforme de los datos.
- **Filtrado Secuencial:** Los sistemas de recomendación se aplican de forma secuencial, donde un método filtra el conjunto de elementos, y luego otro método actúa sobre este conjunto reducido para hacer recomendaciones más precisas.
- **Metamétodos:** Se utilizan métodos de aprendizaje para aprender cómo combinar las salidas de varios sistemas de recomendación, a menudo conocidos como blending o stacking. Estos métodos son entrenados para encontrar la mejor combinación de predicciones.

## 2.3 Principales problemas y dificultades de los sistemas de recomendación

Vamos a estar retratando un poco los retos que presentan los sistemas de recomendación, se debe tener en cuenta que hay dificultades propias de cada sistema, y que otros sistemas no tienen o arreglan de algún modo. Pero en general vamos a hablar de los más comunes.

- **Cold start:** o arranque en frío, hemos hablado de que los sistemas de recomendación se basan en las interacciones usuario-item, pero qué pasa cuando un usuario nuevo entra al sistema y no ha interactuado con ningún item. No se puede hacer una recomendación personalizada si no se tiene información del usuario. Este problema se conoce como cold start.
- **Gray sheep:** o oveja gris, es un problema que surge cuando para un usuario hay muy pocos usuarios parecidos, este es un problema que los sistemas de recomendación que tratan de agrupar usuarios tienen dificultades para resolver.
- **Escacez de datos:** es un problema que surge generalmente cuando un usuario no se molesta en interactuar con los items del sistema, proporcionando poca información sobre si para poder usarla en las recomendaciones.
- **Escalabilidad:** este es un problema un poco mas fácil de entender, a medida que el número de usuarios e items crece, el sistema cada vez necesita más recursos. Solo poniendo un ejemplo de un e-commerce donde hay cientos de miles de productos e usuarios el costo computacional de hacer recomendaciones personalizadas para cada usuario puede ser muy alto.
- **Diversidad:** que pasa en los sistemas que usan la similitud entre los items para hacer las recomendaciones si todos los items son muy similares entre sí. Las recomendaciones se verían muy empobrecidas debido a que no hay una buena forma de elegir items por sobre otros
- **Problemas de los sistemas secuenciales:** además de los problemas antes mencionados, los sistemas secuenciales también presentan su propio conjunto de problemas como por ejemplo: como captar dependencias entre items si la distancia que los separa en la secuencia es muy grande, o como captar cuando un item depende varios items anteriores, o cuando la dependencia de esos varios items anteriores no depende del orden en que aparezcan, entre otros.

## 3 Implementación y arquitectura del modelo

A continuación entraremos en detalles sobre la implementación del modelo propuesto. Como se ha mencionado antes se implementó un sistema híbrido que combina técnicas de filtrado colaborativo, filtrado basado en contenido y tiene cierta consciencia de la secuencialidad.

### 3.1 Datos

Para evaluar nuestro sistema hemos usado una base de datos de películas, y de rating de las películas proporcionada por Movie Lens. La base de datos se puede obtener en el siguiente sitio web. La base de datos contiene 100,000 ratings de 9,000 películas por 600 usuarios. Escogimos esta sobre otras base de datos mas extensas que ofrece el mismo sitio debido a que es más manejable y nos permite hacer pruebas más rápido.

### 3.2 Filtrado colaborativo

Para le filtrado colaborativo implementamos dos sistemas diferentes con el objetivo de comparar la efectividad de cada uno:

### 3.3 Similitud de cosenos

Sea $R_{nxm}$ una matriz de usuarios contra items, donde $n$ es el número de usuarios y $m$ es el número de items. $R[i, j]$ es la valoración que el usuario $i$ le ha dado al item $j$. A partir de esta matriz podemos usar las filas como un vector de un usuario reflejando sus gustos. A partir de estos vectores podemos encontrar usuarios similares a partir de la similitud de cosenos entre los vectores de usuarios. Se define la similitud de coseno como :

$$SimCos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{1}$$

Y a partir de aquí se encuentran los usuarios más similares a un usuario dado y se le recomiendan los items que mejor han valorado en promedio estos usuarios y que el usuario objetivo no haya visto.

**K-means** Para esta solución de sistema utilizamos un algoritmo de clusterización llamado k-means. Más formalmente, k-means a partir de un grupo de $\{x_1, x_2, ...x_n\}$ vectores de un espacio r-dimensional tratar de encontrar :

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 = \arg\min_{S} \sum_{i=1}^{k} |S_i| \, Var \, S_i \tag{2}$$

Donde $\mu_i$ es la media o centroide de puntos en el cluster $S_i$. Es decir agrupa los vectores en k conjuntos a partir de la distancia media euclideana entre los puntos y el centroide. Tomando estos clusters como grupos de usuarios similiares, podemos hacer recomendaciones a un usuario dado las películas que le gustaron a los usuarios del cluster al que pertenece.

### 3.4 Filtrado basado en contenido

Nuestro modelo basado en contenido usa la similitud de cosenos antes mencionada y explicada para hallar similitudes entre las películas a partir de sus etiqutas y géneros. Y dado un usuario le recomienda las películas mas parecidas a las que mejor ha valorado el usuario.

### 3.5 Modelo secuencial

El modelo secuencial implementado es un modelo estadístico sencillo, dado un usuario mira las k últimas películas que ha visto, después para cada usuario busca las próximas 5 películas que vió después de esas. Asignándoles un peso a partir de que tanto se demoró en verla a partir de las otras películas. Y después se recomiendan las películas que más valores tienen.

### 3.6 Hibridación

Para combinar los sistemas anteriores usamos dos formas distintas de hibridar para evaluar su rendimiento y compararlos. La primera fue una hibridación por peso entre los modelos, cada uno sugiere películas y a partir del peso que tiene cada sistema y el valor del ranking que tiene cada sistema queda conformado el ranking conjunto y devuelve los 10 primeros. La segunda fue una mezcla entre entre la hibridación por peso y una hibridación vertical. Es decir, los sistemas de filtrado colaborativo y basado en contenido hacen una recomendación por peso pero devolviendo el doble de películas, y despues el secuencial recomienda pero sobre la salida de los modelos anteriores como su espacio de busqueda.

### 3.7 Otros detalles de la implementación

Uno de los detalles extras que agregamos fue un algoritmo para recomendar películas a los usuarios nuevos, tratando de resolver el problema del cold start en este sistema, usamos un algoritmo sencillo de recomendación por popularidad. Recomendando las películas más populares por genero en base a las valoraciones de otros usuarios.

### 3.8 Deficiencias de la implementación

Hablaremos ahora un poco sobre los problemas del modelo y que se podría mejorar y cómo.

- Escalabilidad y eficiencia de recursos: nuestro sistema funciona bien para base de datos pequeñas pero empieza a caer en rendimiento a medida que aumenta la cantidad de información a manejar empieza a demorarse mucho en computar las respuestas.

- Eficiencia de los sistemas: probablemente con mayor tiempo y recursos será mucho mejor implementar soluciones más complejas para cada sistema, como factorización de matrices mediante SVD o usando modelos secuenciales usando factores latentes o cadenas de Markov.
- Tenemos limitaciones con los datos, son relativamente pobres ya que no tenemos metadatos de los usuarios ni otros tipos de interacciones de usuario-item. Podrían hacerse mejores sistemas teniendo en cuenta datos como tiempo de visualización de películas, o más datos sobre las películas cómo director, actores, premios obtenidos, etc.
- Debido a que al filtrado basado en contenido le cuesta recomendar items con mucha variedad, si los grupos de usuarios parecidos son muy parecidos entonces el filtrado colaborativo también falla en recomendar items diversos.

## 4 Evaluando el sistema

Para evaluar el sistema dividimos el conjunto de datos en dos partes, un grupo actuaría como los datos que posee el sistema, y el otro conjunto de datos realizaría la función de los items relevantes para los usuarios. Así logramos que el sistema recomiende solo usando el primer conjunto y el segundo conjunto lo usamos para evaluar los resultados.

### 4.1 Métricas

Para evaluar la eficiencia del sistema usamos diversas métricas ampliamente usadas en estos ámbitos:

- **Presición** : Se refiere al porcentaje de items recomendados por el sistema que son relevantes para el usuario. $Precision = \frac{Recuperados \cap Relevantes}{Recuperados}$
- **Recall :** Porcentaje de items recomendados de los items relevantes para el usuario $Precision = \frac{Recuperados \cap Relevantes}{Relevantes}$
- **nGDC:** Que básicamente mide que tan bien ordenados están los items en el ranking. Es decir compara la lista con una supuesta lista ideal.
- **Hit rate:** Mide simplemente a cuantos usuarios del total el sistema les recomendó al menos un item relevante.

### 4.2 Valores alcanzados

Como mencionamos antes hicimos variaciones tanto en la hibridación como en el sistema de filtado colaborativo para comparar los resultados y quedarnos con los de mejor eficiencia. El sistema que mejor desempeño tuvo fue el filtrado colaborativo basado en similitud de cosenos junto a la hibridación por peso de los sistemas alcanzando los siguientes valores:

**Table 1.** Resultados obtenidos

| Precision | Recall | nDCG | HR@10 |
|---|---|---|---|
| 0.2430 | 0.0356 | 0.2544 | 0.7574 |

### 4.3 Análisis de resultados

Los resultados arrojados por el modelo son relativamente buenos, una precisión del 0.24 significa que de cada 4 películas recomendadas el usuario consider relevante una de ellas. Y al igual que ese el hit rate indica que a tres de cada cuatro usuarios le es útil al menos uno de los items recomendados. El valor bajo del recall, apenas un 3% es esperable, ya que la cantidad de items que se recomienda es bastante bajo comparado con la cantidad de items que pueden resultar relevantes para el usuario. Y un valor del nDCG de 0.25 también es relativamente bueno dando a entender que es al menos parcialmente bien ordenada la lista.

## 5 Conclusiones

El sistema propuesto ha demostrado un desempeño modesto pero funcional en la tarea de recomendar películas a usuarios. Tiene la ventaja de ser sencillo y necesitar poco recursos a cambio de un desempeño menor. Puede manejar a usuarios nuevos con estrategias básicas. Por supuesto el sistema es ampliamente mejorable tanto en recursos como en técnicas utilizadas.

## References


1. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields Hyeyoung Ko, Suyeon Lee, Yoonseo Park and Anna Choi ,
2. Movies recommendation system using collaborative filtering and k-means Phongsavanh Phorasim and Lasheng Yu School of Information Science and Engineering, Changsha, Hunan, China
3. Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim and Rasha Kashef,
4. Movie Lens Database : https://grouplens.org/datasets/movielens/
5. Introduction to recommender systems.Overview of some major recommendation algorithms : https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada