Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC3633 - Sistemas Recomendadores

Sistemas de Recomendación de Libros Personalizados

2° Semestre - 14/12/2023

Profesor: Andrés Francisco Carvallo de Ferari

Ayudantes: Pablo Messina y Calos Muñoz

Estudiantes: **Gabriel Aguirre, Pablo Bahamondes e Ignacio Medel**

ÍNDICE

1. Abstract

Se aborda el desafío de mejorar los sistemas de recomendación de libros. Para empezar, vemos el contexto de los usuarios frente a la lectura, destacando que a pesar del creciente interés en la lectura, la recomendación de libros es un problema a resolver. Por ende, se propone una metodología basada en el Filtrado Basado en Contenido, utilizando vectores caracterizados por información relevante de los libros.

Asimismo, con la creación y evaluación de modelos baselines, como *Random*, *Best Rating* y *Most Popular*, sin embargo los tres modelos son insuficientes, por lo que es imperativo un enfoques más personalizados. Así llegamos a nuestros modelos propuestos, basados en vectores de libros (Titulo-Autor-Editorial), ademas con y sin abstract, los cuales demuestran un desempeño superior. No obstante, el modelo que utiliza el abstract no siempre es mejor, por lo tanto hay que buscar un equilibrio entre enriquecer y sobrecargar la información del modelo.

2. Introducción

En el año 2022 en Chile, la Asociación de Investigadores de Mercado y Opinión Pública de Chile (AIM), realizó una **encuesta de 1.719 personas**, con el objetivo de generar estadísticas con el fin de **promover el desarrollo sociocultural y educativo del País de Chile**.

Mediante la encuesta, se caracterizó a los lectores del país, donde lo más atingente para este documento es lo siguiente:

1. El **76 % chilenos declaran leer algún material todas las semanas**.
2. De los consultados, el **23 % afirma leer todos los días**.
3. De los consultados, el **27 % afirma leer casi todos los días**.
4. De los consultados, el **26 % afirma leer 1 o 2 veces por semana**.

Los resultados anteriores, el **82 % de los chilenos que declaran leer, se sienten abrumados al momento de elegir qué libro leer a continuación, optando por libros de un mismo autor o recomendaciones de pares**.

Es decir, como podemos observar en las estadísticas anteriores, existe un claro **auge en el consumo de libros**, donde la **pobre recomendación de estos es un problema** tanto social como económico, ya que una **pobre recomendación incurre en una desincentivación a la lectura**.

Conjunto a lo anterior, en la literatura los **Sistemas Recomendadores actuales** [1] para la recomendación de libros, este problema es **tacleado de formas simples**, tales como: filtrado colaborativo en base a usuario e ítems, recomendación tipo *Most popular*, recomendación tipo *Highest rating*, etc.

La atención de este trabajo es **lograr una mejor caracterización de los libros**, con el fin de **recomendar atingentemente a los posibles usuarios** con el fin de **incentivar la lectura**.

3. Estado del Arte

El área **relacionada a recomendación de libros**, ha sido estudiada durante bastante tiempo, esto **fomentado por distintas razones**, algunas mencionadas en la sección anterior, siendo la más destacable la del **comercio**.

Conforme a la **aparición de nuevas tecnologías**, se ha **democratizado el consumo de libros**, puesto que el costo del formato físico es por muy mayor a la del formato virtual del mismo. Lo anterior, ha instigado una **presión comercial sobre la recomendacion de los libros**, desarrollándose así los modelos usados hoy en día.

[1] Se profundiza la idea en la siguiente sección

Dado lo anterior, **existen diversas estrategias para la creación de sistemas recomendadores para libros**, en sí para **fomentar el consumo de estos**. Las estrategias más utilizadas son: **Filtrado Colaborativo**, **Recomendación Basada en Contenido**, además con el auge actual de las inteligencias artificiales, hay **modelos vinculados a redes neuronales**.

Sin embargo en este *paper*, vamos a **profundizar en el estudio del Filtrado Basado en Contenido**, para ser más específico se analiza la situación de si **incorporar información adicional, mejora o entorpece al modelo recomendador**, en qué **casos es beneficioso o en que casos afecta negativamente**, es por esto que decidimos buscar el Abstract de los libros para poder realizar la comparación, esto quedo mejor explicado en las secciones posteriores.

4. Exploración de datos

Utilizamos la base de datos *Book Recommendation Dataset* proveída Kaggle, la cual consiste en **más de 1 millón de reviews de libros**. Escogimos esta base de datos en particular por su **extensión de uso en múltiples países**, **rica caracterización de usuario** y **gran variedad de libros**. Lo anterior es indicativo que esta base de datos es lo **suficientemente democrática para que sea representativa para la población Chilena**.

La base de datos contempla las siguientes tablas y atributos:

1. **Users**: Información de usuario (ID, Locación y edad). Mediante esta, haremos nuestro vector de usuario.
2. **Books**: Información de los libros (ISBN, Titulo, Autor, Año de publicación, Editorial, URL Imagen Portada). Mediante esta, haremos nuestro vector de libros.
3. **Ratings**: Información de las interacciones entre las dos tablas anteriores (User-ID,ISBN,Book-Rating). Mediante esta, haremos el entrenamiento, validación y testeo de nuestros modelos.

Sobre los datos anteriores, hicimos un **preprocesamiento estándar de los datos**, el cual consistió en:

1. **Descartamos aproximadamente el 10.32 % de los ratings** por tener **información defectuosa** (ISBN no válidos o ISBN que no están en la tabla Books).
2. **Descartamos aproximadamente el 0.45 % de los libros** por tener **información defectuosa** (ISBN no válidos).
3. **Descartamos aproximadamente el 66.97 % de los usuarios**, ya que estos solo **hicieron 2 o menos ratings**, con el fin de **disminuir el ruido por outliers**. Quedando así con **92.106 usuarios en total**, con la distribución:

Cantidad Interacciones	Cantidad Usuarios
3	91975
4	105
5	20
6	2

Tabla 1: Distribución Interacciones de Usuario

Además, **mediante los ISBN, scrappeamos de la internert los abstractos de los libros**, con el fin de **complementar y enriquecer nuestros vectores del mismo**.

Finalmente, **editamos tuplas** que tuvieran **errores de información**, es decir, que su **información estuviera 1 o más columnas desfasadas** o que **estuvieran mal tipificadas**.

5. Metodología

5.1. Métricas y Evaluación

En el caso de la métrica, usamos una bastante estándar, denominada HIT@k, su fórmula se representa con la siguiente ecuación.

$$\text{HIT@}k = \frac{1}{|U|} \sum_{i=1}^{|U|} 1_{r(u_i) \leq k}$$

Que es bastante popular, para evaluar sistemas recomendadores y en este caso no es la excepción.

Donde U sería el set de usuarios y $r(u_i)$ la clasificación del ítem que será recomendado para el usuario u_i

Para la evaluación de los modelos, adoptamos un enfoque típico, bastante común para estos casos, que es el **leave-one-out**, que básicamente tienes una serie de tiempo con las acciones de cada usuario, y el último ítem lo dejas aparte para que sea del conjunto de testeo y lo restante lo utilizas para la fase de entrenamiento.

No obstante, en nuestro caso, no tenemos una serie de tiempo del historial del usuario, como se pudieron percatar en la sección de arriba, sino que tenemos una lista de rating de los libros, es parte que escogimos sacar al último ítem de cada usuario. Es algo bastante estándar.

Pasando a la metodología de nuestro proyecto, lo primero que pensamos como equipo, fue en la creación de modelos base, estos que te dicen tu modelo no puede ser más malo que esto, ya que estas diseñados para eso.

En este sentido, realizamos tres: **Random**, **Best Rating** y **Most Popular** , en la siguiente sección entraremos en detalle.

5.2. Modelos Baselines

Para nuestra **elección de modelos Baselines**, como equipo buscamos capturar el **comportamiento de un lector promedio**, es decir, cuando este va a una librería y escoge un libro.
Dado lo anterior, decidimos utilizar los siguientes modelos:

1. **Random**: Uno de los **modelos bases más populares**, en el cual es escogio **cualquier libro del conjunto disponible de ítems y se lo recomienda al usuario**, es por esto que es esperable que las **métricas sean bajas ya que las recomendaciones no sigue ninguna regla o lógica**.
2. **Most Popular**: Corresponde a la **recomendación más típica u obvia**, es la del **libro más popular o trending**.
3. **Best Rating**: Corresponde a la **recomendación de "libros de culto"**, en estas clasificaciones también entrarían los **libros históricos**, aquellos libros que perduran durante generaciones.

De esta forma, de manera preliminar, se buscó capturar el **comportamiento del usuario al momento de aquirir un libro**. Escogimos estos modelos Baselines dado que tendrán un **desempeño pobre y esperable**, dado que **no es recomendación personalizada al usuario**. Lo anterior dado que buscamos **contrastar nuestros modelos propuestos**, esperando que el **desempeño de estos sea superior a lo obtenido por los Baselines**.

5.3. Modelos Propuestos

Los modelos que proponemos para abordar el problema de *¿A quién y cual libro recomendar?*, consideramos pertinente el **Filtrado basado en contenido sobre un vector de libro**, es decir, **recomendar libros semejantes a otros libros dado su semejanza vectorial**. Lo anterior, fomentado por la idea de que un libro es esencialmente un gran **vec-**

tor de información, dado que cuanta con: **palabras**, **autores**, **editoriales**, **abstracto**, entre otros. Dicha informacion, puede ser **lematizada**, **tokenizada** y posteriormente **vectorizada**; vectores que podemos utilizar para realizar la recomendación.

Dado lo anterior, consideramos pertinente **caracterizar los libros como vectores y hacer recomendaciones en base a estos**, considerando como métrica de similitud la **similaridad coseno**. La información que decidimos tomar para cada vector es la siguiente:

1. (1) Vector **Título-Autor-Editorial**.
2. (2) Vector **Título-Autor-Editorial-Abstract**.

La diferencia entre ambos vectores radica en que **uno contempla el abstract el libro y el otro no**. Lo anterior nos pareció interesante, dado que queremos **encontrar el punto óptimo en la caracterización del libro**, donde somos conscientes que **mientras más información, más rico es el vector**, pero asimismo, también **aumenta la entropía de información**, encontrándonos en un *tradeoff* entre con cuanto más información podemos enriquecer nuestro vector versus cuando empieza a empobrecerlo por redundancia.

6. Análisis

En base a nuestra metodología, primero se **obtuvo las métricas** de los **modelos Baselines**. Obtenemos el resultado para la métrica HIT@k, con un $k \in \{10, 50, 100, 1000\}$, que esta representado en las siguientes tablas:

	Hit@10	Hit@50
Random	0.035 %	0.158 %
Best Rating	0.177 %	0.283 %
Most Popular	0.901 %	2.110 %

Tabla 2: HIT@k con k $=\{10,50\}$

	Hit@100	Hit@1000
Random	0.247 %	2.807 %
Best Rating	0.512 %	3.982 %
Most Popular	3.117 %	**13.103 %**

Tabla 3: HIT@k con k $=\{100,1000\}$

De manera evidente, podemos notar como el **modelo Random es el modelo con peor desempeño para cualquier k**, seguido del **modelo Best Rating**, donde **ni uno de los 2 alcanza ni el 1 $\%$ de éxito**. Lo que es indicativo que para ambos modelos **no existe**, para los 4 valores de k disponible, **un elemento relevante que se encuentre dentro de los primeros K elementos recomendados**. En simples palabras, los modelos **Random** y **Best Rating no recomiendan activamente al usuario para cualquier k**

Por otro lado, podemos afirmar que el **Most Popular es el mejor de los modelos Baselines**, ya que es posible notar un **mejor desempeño conforme el valor de k crece**, lo que es indicativo que **mientras más grande es la lista, más probable es encontrar un elemento relevante en esta**, lo que hace sentido, dado que la recomendación *Most Popular* recomienda, valga la redundancia, los más populares y por tanto, tendrán mayor probabilidad de consumo/acierto.

Dado el desempeño anterior, podemos afirmar que hemos **escogido correctamente nuestros modelos Baselines**, dado que **todos muestran una tendencia positiva al aumentar el valor del k, pero así mismo tienen un desempeño pobre**

Continuamos con el análisis de desempeño para nuestros **modelos propuestos** mediante la métrica HIT@k, con un $k \in \{10, 50, 100, 1000\}$:

	Hit@10	Hit@50
Sin Abstract	25.56 %	35.56 %
Con Abstract	30.56 %	35.64 %

Tabla 4: HIT@k con k $=\{10,50\}$

	Hit@100	Hit@1000
Sin Abstract	38.09 %	**41.28 %**
Con Abstract	36.67 %	39.35 %

Tabla 5: HIT@k con k ={100,1000}

Podemos apreciar de forma inmediata un **mejor desempeño para ambos modelos**, dado que **conforme aumenta el valor de k, mayor es la relevancia de los ítems recomendados**.

Además, es destacable señalar que el **modelo recomendador** con **mejor desempeño** entre los dos es **aquel sin el abstract**, dado que **converge para valores más altos de k en la métrica**, donde el **modelo con abstract presenta mejores métricas pero tan solo hasta k = 50**, lo que es indicativo de que **agregar el abstract introduce mayor entropía de datos**, lo que se **evidencia conforme aumenta el valor de k, empobreciendo su desempeño**.

7. Resultados

Como se comento en la sección anterior, podemos notar como el **desempeño de nuestros modelos propuestos esta muy por encima de los modelos Baselines**. Lo anterior, podemos visualizarlo en la figura 7.1.

Acá podemos ver de manera gráfica como el **desempeño de nuestros modelos propuestos supera notoriamente a los Baselines**, es decir, hemos podido **validar el desempeño de nuestros modelos**.

Además, se puede apreciar de manera visual como **el modelo con abstract es mejor para valores de k más pequeños** y como el **modelo sin abstract converge para valores de k más grande**.

8. Conclusiones y Trabajo Futuro

De las secciones anteriores vistas sobre el contexto del problema, planteamiento de solución y resolución del mismo, podemos concluir que:

1 Podemos afirmar que los **modelos basados en contenido para la recomendación de libros son exitosos**, es decir, nuestra idea de aprovechar la **lematización**, **tokenización** y **vectorización** de los libros, si nos da una **ventaja al momento de recomendarlos**, esto comparando con los modelos bases.

2 Como hipótesis grupal, **determinábamos beneficioso incluir más información al modelo recomendador**, ya que al hacerlo **provocaría que este tuviese mejor desempeño**, sin embargo, **durante la validación de los modelos**, **pudimos apreciar y comprobar que esto no es necesariamente cierto**, ya que, al **incluir el Abstract pensamos que estábamos enriqueciendo el vector de libro**, lo que implicaría un mejor desempeño del modelo. A pesar de lo anterior, hubo un **declive en el desempeño general del modelo**, más notorio para valores de k sobre 50 (donde suponíamos que debería ser mejor). Más aun, el **modelo sin Abstract, tuvo menos desempeño con k pequeño y logro superar al otro modelo a medida que k era lo suficientemente grande**, es decir, muestra una tendencia a converger conforme aumenta el valor de k.
Por lo tanto, es pertinente destacar que existe un *tradeoff* con la información que estás incorporando al modelo. Dado lo anterior, hay que encontrar un equilibrio con la cantidad de información que se añade, debido a que no todo afecta positivamente al mismo.

3 Como equipo se llego a la conclusión que un paso a seguir para posiblemente mejorar el desempeño del modelo, es la incorporación de imágenes, (se podría hacer utilizando embeddings de imágenes de las portadas preprocesador por ResNet50). Lo anterior presenta ser un desafió en si mismo, ya que el *Dataset* cuenta con 259.000 libros aproximadamente, por lo mismo, hay que tener en cuenta la capacidad de memoria y optimización de código, ya que no es una tarea sencilla incorporar tal cantidad

de embeddings al modelo, no obstante, consideramos que puede ser una buena fuente de información para la mejora del modelo, dada la presión comercial que existe sobre el marketing de las portadas de los mismos.

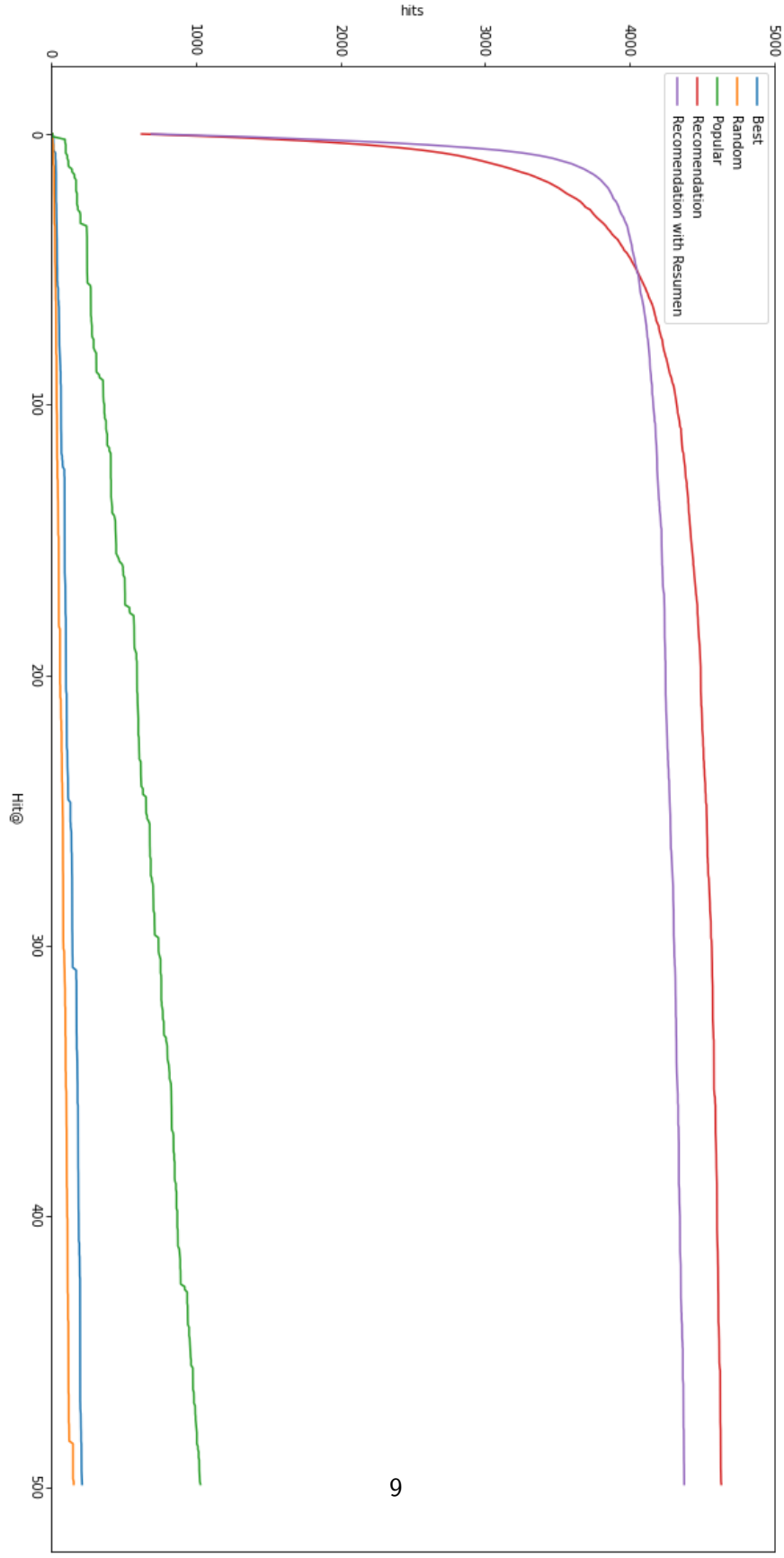

Figura 7.1: Gráfico desempeño modelos