Recomendación a grupos: impacto de métricas de similaridad y métodos de agrupación en rendimiento de modelos.

César Olguín
csolguin@uc.cl
Pontifica Universidad Católica de Chile
Santiago, Chile

Francisca Ibarra
faibarra1@uc.cl
Pontifica Universidad Católica de Chile
Santiago, Chile

ABSTRACT

Los sistemas recomendadores para grupos son sistemas cuyo enfoque es el generar recomendaciones de ítems a grupos de personas, tomando en cuenta las preferencias personales de cada una y cómo estas influyen en la percepción del grupo sobre el ítem. Este paper estudia la influencia que posen las métricas de similaridad al momento de generar *clusters* de usuarios de entrenamiento y los métodos de agrupación para resumir las preferencias del grupo sobre un ítem sobre las recomendaciones que entrega un modelo. Se obtiene que ambos factores afectan de forma significativa la calidad de las recomendaciones entregadas, lo que implica la necesidad de tomar en cuenta dichos factores al momento de crear recomendadores a grupos.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; *Personalization*; • **General and reference** → **Empirical studies**.

KEYWORDS

recommender systems, collaborative filtering, group recommender systems, clustering

1 INTRODUCCIÓN

El área de los sistemas recomendadores ha tomado vital importancia en los últimos años a medida que aumenta la cantidad de usuarios e ítems con los que interactúa cada uno, debido a que sin el apoyo de un sistema de recomendación es muy díficil que un usuario pueda revisar todo el catálogo de una plataforma en particular y seleccionar los ítems que sean afines a sus preferencias.

Una subárea del tema es la **Recomendación a grupos**. El objetivo de esta es recomendar items a grupos de personas que están realizando una tarea o actividad en común. Para esto, se debe construir un modelo que represente y resuma las preferencias del grupo para poder aplicar los modelos de recomendación a estas, siendo planteadas diversas formas de generar dichos modelos.

A modo general, los problemas que deben abordar los sistemas recomendadores para grupos tienen que ver con el encontrar combinaciones adecuadas de modelación del problema tal que presenten resultados óptimos al momento de recomendar a un grupo de entrada. En particular, algunos aspectos que pueden variar son: *Clustering*, que corresponde a la forma de encontrar y conformar los grupos de personas que se utilizarán como base para el entrenamiento del modelo; *Similarity*, que corresponde a la métrica que se usará para calcular la similaridad de las personas con el fin de formar los *clusters* mencionados anteriormente y *Group Modelling Aggregation*, correspondiente a la forma que se utilizará para condensar los *ratings* de cada persona perteneciente a un grupo en un único *rating* que represente la opinión del grupo respecto a un ítem.

Los distintos aspectos variables de los modelos de recomendaciones individuales se han abordado extensamente en la literatura, mientras que desde el lado de las recomendaciones grupales, investigaciones de este tipo son más escasas. Debido a esto, y considerando que es importante realizar una investigación análoga de variables para recomendaciones grupales, decidimos investigar las posibles formas de aumentar el rendimiento de un sistema de recomendaciones a grupos, con un enfoque en *Similarity* y *Group Modelling Aggregation*.

2 ESTADO DEL ARTE

Según Jameson y Smyth [8], las principales tareas respecto a la recomendación a grupos se dividen en cuatro: adquirir la información de las preferencias de cada usuario individual, general las recomendaciones, explicar las recomendaciones y ayudar al grupo a decidir en una recomendación final.

Respecto a la tarea de adquirir las preferencias de los usuarios individuales para las recomendaciones a grupos, dichos sistemas recolectan dicha información con estrategias bastante similares a las que usan sistemas de recomendación a usuarios individuales [1] [8]. La información explícita se recolecta vía evaluaciones directas de cada usuario individual sobre un ítem, como lo son los sistemas de *ratings* binarios o de valores dentro de una escala predefinida, o los sistemas que permiten reseñas personalizadas de cada usuario sobre el ítem. Mientras que la información implícita se obtiene a partir de atributos como la ubicación geográfica o las reacciones físicas que puede tener un usuario respecto al ítem.

Sin embargo, al momento de recomendar a grupos, se deben tener en cuenta factores adicionales respecto a las preferencias individuales recolectadas. A modo general, se ha demostrado que el visibilizar las evaluaciones que han hecho otros usuarios sobre un ítem tiene un impacto significativo sobre la calificación que entregará un nuevo usuario a dicho ítem [4], mientras que cuando se trata de tomar decisiones en grupo, el revelar las preferencias individuales puede afectar de gran manera las decisiones o preferencias finales de cada miembro del grupo [6].

Una forma de obtener las preferencias de un grupo, basándose en las preferencias individuales de cada usuario perteneciente a este, es mediante estrategias de agregación basadas en decisiones sociales, explicadas por Masthoff (2011) en el *Recommender Systems Handbook* [11]. En dicho texto se explican once estrategias de agrupación diferentes, junto a ejemplos de implementaciones reales que utilizan una o más de estas en sus sistemas. Respecto a la influencia que tienen estos métodos en la calidad de recomendaciones, se han

realizado comparaciones entre los resultados que entregan estas estrategias de agregación y las decisiones tomadas por usuarios reales, junto con la lógica que siguieron para llegar a dicha decisión, con el fin de argumentar cuál estrategia sería mejor utilizar. También se han realizado investigaciones donde se compararon algunas estrategias de agregación aplicadas a métodos de filtrado colaborativo, en base a métricas de evaluación para sistemas recomendadores [9].

Tradicionalmente los sistemas recomendadores a grupos trabajan recomendando a grupos pre-formados [5]. Algunos sistemas [3], detectan y forman estos grupos de forma automática (es decir, el usuario no debe insertarse en un grupo de forma manual o no se debe introducir manualmente los grupos en el sistema). Estos grupos se forman de manera automatizada mediante técnicas de *clustering* sobre la matriz usuario-ítem. Sin embargo, no hay muchas investigaciones que se enfoquen en la formación de grupos [5], por lo que es necesario explorar al respecto, con el fin de analizar su impacto en los modelos de recomendación a grupos.

3 DATASET

Se realizó un análisis preliminar y exploración del *dataset*, con el fin de tener un conocimiento preliminar sobre los datos a trabajar, su distribución, y así poder entender de mejor manera los resultados finales obtenidos.

El *dataset* utilizado para esta investigación corresponde a MovieLens 1M [7], el cual almacena información de preferencias de usuarios respecto a películas. En particular, el *dataset* contiene información sobre 1 millón de *ratings* acerca de 4000 películas, evaluadas por 6000 usuarios. Este fue elegido debido a que permite simular un caso de uso real para un sistema recomendador para grupos, donde se necesita recomendar un ítem a un grupo que participará de una actividad social en conjunto.

De forma general, se obtiene que el dataset tiene una *sparsity* del 95.5%, lo que implica que no existe información sobre la gran mayoría de las combinaciones usuario-película.

Con respecto a los usuarios presentes en el dataset, se obtiene que la mayoría son hombres que conforman alrededor del 72% de la cantidad de usuarios totales, versus un 28% de usuarios mujeres. Por otro lado, la edad promedio de estos usuarios es de 30.6 años. Para analizar cómo se comportan los usuarios en relación a los *ratings*, la Figura 1 presenta la cantidad de películas que evaluó cada usuario del dataset, ordenados de mayor a menor.

A partir de la distribución obtenida, se puede apreciar que no todos los usuarios evaluan una cantidad similar de ítems. Por lo contrario, en realidad existe un grupo reducido de usuarios con una cantidad significativamente superior de *ratings* a su nombre que la del resto de usuarios presentes en el *dataset*. En particular, se puede ver que los cinco usuarios con mayor cantidad de interacciones tienen en conjunto aproximadamente 1% del total de los *ratings* totales.

De forma análoga, un análisis sobre las películas entrega una distribución similar, como se puede apreciar en la Figura 2. En este caso, se puede interpretar que nuevamente existe un grupo reducido de películas que se llevan la mayoría de los *ratings* totales del *dataset*, mientras que la mayoría de las películas tienen un número mucho más reducido de *ratings*, llevándose las cinco películas con mayor cantidad de interacciones un total de 1.5% de los ratings del *dataset*.

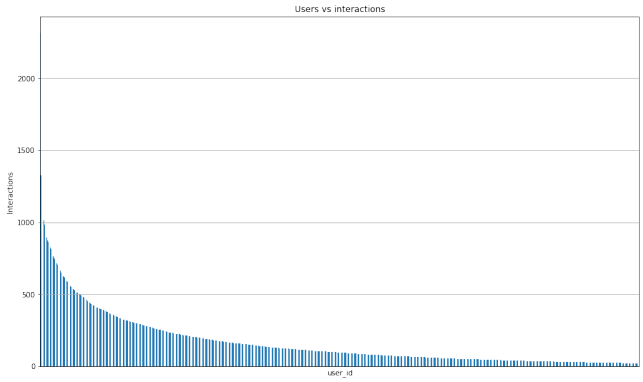

Figure 1: Gráfica de la cantidad de películas evaluadas por cada usuario, ordenados de mayor a menor según la cantidad de interacciones.

Table 1: Interacciones de los top 5 usuarios

ID usuario	Interacciones	Porcentaje
4169	2314	0.002314
1680	1850	0.001850
4277	1743	0.001743
1941	1595	0.001595
1181	1521	0.001521

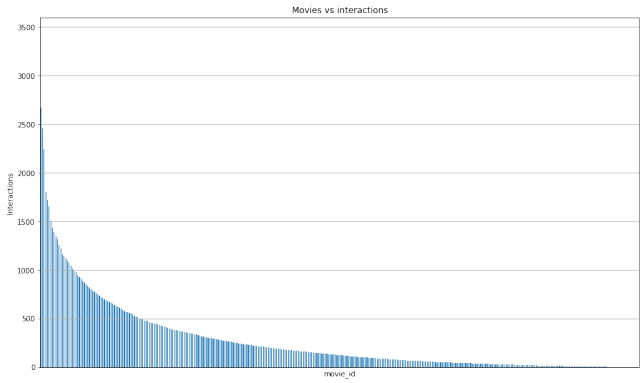

Figure 2: Gráfica de la cantidad de evaluaciones que contiene cada película, ordenados de mayor a menor según la cantidad de interacciones.

Finalmente, al realizarse un análisis sobre los tipos de *ratings* recolectados, representado en la Figura 3, se tiene que la mayoría de estos son calificaciones positivas sobre las películas (es decir, mayor o igual a 3 en una escala del 1 a 5). Esto, en conjunto con el alto porcentaje de *sparsity* encontrado, implica que los usuarios tienden a evaluar únicamente las películas que consideraron positivas o representativas de sus gustos, mientras que una cantidad minoritaria de estos evaluó a las películas que no están dentro de sus preferencias personales.

Table 2: Interacciones de las top 5 películas

ID película	Interacciones	Porcentaje
2858	3428	0.003427
260	2991	0.002990
1196	2990	0.002989
1210	2883	0.002882
480	2672	0.002671

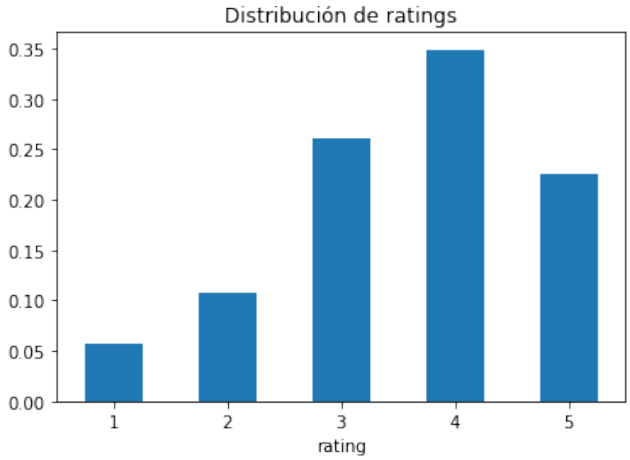

Figure 3: Distribución de los ratings entregados en el dataset.

4 METODOLOGÍA

Tomando como inspiración el trabajo de Nawi [9], y con el fin de comparar la influencia de las métricas de similitud entre usuarios y las estrategias de agregación en el rendimiento de un sistema recomendador a grupos, se decide trabajar implementando diferentes modelos de *Collaborative Filtering*, los cuales mantienen los mismos hiperparámetros excepto por las dos variables mencionadas anteriormente.

Para esto, inicialmente se procesa el *dataset* formando una matriz usuario-ítem. Debido al problema de la *sparsity* mencionado anteriormente, se necesita llenar toda combinación usuario-película no disponible en el *dataset* inicial. Tomando como base el análisis respecto a los *ratings* previo, se decide llenar toda combinación faltante con la calificación mínima (1), debido al patrón encontrado donde los usuarios tendían a evaluar las películas que sí fueran acordes a sus gustos por sobre las que no lo eran.

Una vez obtenida la matriz usuario-ítem, se realiza el siguiente procesamiento al *dataset* una vez por cada combinación de métrica de similitud y técnica de agregación evaluada, obteniendo finalmente doce *sub-datasets* distintos con los que entrenar modelos.

Se genera una matriz de similitud usuario-usuario, utilizando la métrica de similitud para el cálculo entre usuarios. Esta matriz es entregada a un algoritmo de *Spectral Clustering*, el cual genera una cantidad fija de *clusters* o grupos basados en la matriz de similitud.

Teniendo los grupos de usuarios, se procede a resumir sus preferencias en un *rating* para cada película por grupo utilizando la estrategia de agregación, obteniendo una matriz grupo-película. A estas matrices se les aplica una separación *train-test*, y se proceden a entrenar dos modelos con el *dataset* de entrenamiento. En específico, los modelos utilizados fueron:

- Modelo de *Alternating Least Squares* (ALS), con 50 factores latentes, 5 iteraciones de entrenamiento, y un parámetro de regularización 100.
- Modelo de *Singular Value Decomposition* (SVD), con 100 factores latentes, un tope de 100 iteraciones como máximo, una tasa de entrenamiento de 0.01 y un parámetro de regularización de 0.1.

Utilizando el *dataset* de prueba, finalmente se obtienen las métricas de evaluación pertenecientes a esa combinación de métrica de similitud y estrategia de agregación en específico.

Ambos modelos utilizados corresponden a implementaciones realizadas por la librería pyRecLab [12], mientras que para el algoritmo de *Spectral Clustering* se utiliza la implementación de scikit-learn [10].

5 ANÁLISIS DE PARÁMETROS

En este trabajo, se manejaron dos parámetros distintos, con el fin de analizar la influencia que tenía variar cada uno de ellos en la calidad de las recomendaciones finales entregadas por los modelos.

El primer parámetro corresponde a la métrica de similitud utilizada para calcular las similitudes entre usuarios y posteriormente formar los grupos con los que se entrenaría el modelo. Las métricas utilizadas fueron:

- *Cosine Similarity*, la cual mide el coseno del ángulo entre dos vectores, sin tomar en cuenta su magnitud. Se elige esta fórmula debido su baja complejidad computacional para vectores con alta *sparsity* y su fórmula esta dada por:

$$sim(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||}$$

- *Adjusted Cosine Similarity*, la cual toma en cuenta el hecho de que diferentes usuarios poseen diferentes escalas personales al momento de evaluar un ítem, por lo que primero resta el promedio de las evaluaciones que hizo cada usuario a su vector, y calcula el coseno del ángulo de los vectores resultantes. Esta métrica tiene el beneficio de ser más sensible que la anterior y su fórmula es:

$$sim(A, B) = \frac{(A - \overline{A}) \cdot B - \overline{B}}{||A - \overline{A}|| \cdot ||B - \overline{B}||}$$

- *Jaccard Similarity*, mide la similaridad entre conjuntos. Esta métrica posee un alto costo computacional y actualmente no es escalable con *datasets* de gran tamaño [2], por lo que se incluye en este trabajo con el fin de concluir si los beneficios respecto a su rendimiento permiten obviar su costo computacional. Su fórmula es:

$$sim(A, B) = \frac{A \cap B}{A \cup B}$$

Cabe notar que inicialmente se había incluido una métrica adicional, *Pearson Similarity* sin embargo esta es matematicamente igual a *Adjusted Cosine Similarity*, por lo que se decidió eliminarla de las posibles métricas a analizar para evitar redundancia.

Respecto a las estrategias de agregación, se seleccionaron cuatro estrategias, de las cuales tres están presentes dentro del listado hecho por Masthoff [11] y una fue diseñada para ese trabajo. Las estrategias evaluadas fueron:

- *Average*, donde el *rating* del grupo respecto a un ítem corresponde al promedio de los *ratings* entregados por cada usuario del grupo a ese ítem.
- *Most Pleasure*, donde el *rating* del grupo respecto a un ítem corresponde al máximo de los *ratings* entregados por cada usuario del grupo a ese ítem.
- *Least Misery*, donde el *rating* del grupo respecto a un ítem corresponde al mínimo de los *ratings* entregados por cada usuario del grupo a ese ítem.
- *Personalizada*, donde el *rating* del grupo respecto a un ítem corresponde a la suma entre el promedio de los datos y el promedio entre el mínimo y máximo de los ratings entregados por los miembros del grupo, dividido en 2.

La función *personalizada*, se crea debido a que pueden presentarse casos donde el promedio suele ser engañoso, por ello se trata de disminuir el sesgo promediándolo con el rango medio, que es el promedio entre el valor máximo y el mínimo.

6 RESULTADOS OBTENIDOS

Tras entrenar las distintas configuraciones de modelos, que en este caso corresponde a 24 combinaciones (3 *Similarity* $*$ 4 *Group Modelling Aggregation* $*$ 2 *recommendation models*), se midió el rendimiento de cada uno en base a diferentes métricas de evaluación para sistemas recomendadores, con el fin de obtener un análisis lo más amplio posible.

Respecto a los resultados de las métricas MAP@5 y NDCG@5 presentes en la Figura 4, se puede apreciar que en ambos modelos las configuraciones que utilizan el método de agregación *Most Pleasure* posee resultados positivos, siendo en SVD la estrategia de agregación con mejores resultados, y en ALS una de las dos mejores estrategias, en conjunto con *Average*. En el caso de las métricas de similitud, se puede apreciar un cambio en el valor de las métricas de evaluación dependiendo de la métrica de similitud que utilizara cada modelo, sin embargo la diferencia entre las métricas es mucho menor que la observada en las estrategias de agregación. En particular, *Adjusted Cosine* es la métrica que tiende a tener mejores resultados en SVD, mientras que en ALS la mejor métrica depende de qué estrategia de agrupación se esté utilizando: en el modelo con agregación *Most Pleasure*, la métrica *Jaccard* es la que presenta mejor rendimiento, mientras que con *Average*, la métrica *Adjusted Cosine* es la con mejores resultados.

Tras incrementar el número de recomendaciones que entrega cada módelo, se presentan los resultados de MAP@10 Y NDCG@10 en la Figura 5, en donde el orden de rendimiento entre distintas configuraciones se mantiene e incluso se acentúa más, permitiendo mostrar de forma más clara las diferencias entre los rendimientos de ambos modelos. Esto permite corroborar las diferencias en rendimiento que se generan a partir de las diferentes configuraciones de métricas de distancia y estrategias de agregación usadas. La excepción a este patrón ocurre en el modelo ALS, con estrategia de agregación *Most Pleasure*, donde al aumentar el número de recomendaciones, la métrica de similitud *Jaccard* deja de ser la más efectiva y es reemplazada por *Adjusted Cosine*. Este nuevo orden respecto a las métricas tiene mayor sentido al ser comparado con el resto de las configuraciones, donde de forma consistente existe un mejor rendimiento presente en *Adjusted cosine* con *Most Pleasure*.

Con el fin de analizar la cantidad de falsos positivos y negativos, se evaluaron los modelos con las métricas *Precision@5* y *Recall@5*, obteniendo los resultados de la Figura 6. En estas métricas también se nota el alto rendimiento de la estrategia *Most Pleasure* en ambos modelos, y en el caso especifico de ALS la superioridad en rendimiento de *Average* para 5 recomendaciones.

A modo general, se puede apreciar que ALS posee un mayor rendimiento que SVD, cuando debería ocurrir lo opuesto, al ser ALS un método para *ratings* implícitos y SVD uno para *ratings* explícitos. Esto se puede explicar con el hecho de que al no ser los modelos en sí el foco de la investigación, no se dedicó una cantidad extensa de tiempo a elegir los mejores hiperparámetros para cada modelo, por lo que se puede haber elegido una configuración de hiperparámetros errónea para SVD, provocando su baja en rendimiento con respecto a ALS. Sin embargo, como estos hiperparámetros no varían entre las distintas configuraciones de modelos, la diferencia en rendimiento entre ambos tipos de modelos no afecta de forma negativa en los resultados finales obtenidos.

7 CONCLUSIONES Y TRABAJO FUTURO

Los resultados obtenidos permiten confirmar que las variables seleccionadas para estudio sí afectan en el rendimiento general de un sistema de recomendación a grupos. Es decir, la métrica de similitud y el método de agregación que se seleccione sí son relevantes en la calidad de las recomendaciones que se obtengan.

Se obtuvo que la mejor métrica de similitud fue *Adjusted Cosine*, lo que se condice con el hecho de que es una métrica más sensible a las diferencias entre vectores, mientras que la mejor estrategia de agregación fue *Average* o *Most Pleasure*, dependiendo el modelo y la cantidad de recomendaciones pedidas, siendo la primera estrategia una que busca nivelar o promediar los gustos de los usuarios mientras que la segunda apela a un único usuario.

Con respecto a *Most Pleasure*, puede resultar en una de las mejores estrategias de agregación debido a que al realizar el cluster se agrupan usuarios similares en el grupo de entrenamiento y test, por lo tanto, al tener gustos similares, maximizar la "felicidad" de los usuarios del grupo tiende a un mayor resultado, sin embargo, se espera si se presentan grupos *random* la estrategia *most pleasure* disminuya su resultado.

Es posible mejorar o avanzar en este trabajo de diversas formas. Se puede abordar la hipotesis que se planteó respecto a las diferencias en rendimiento entre los modelos SVD y ALS haciendo un proceso de optimización de hiperparámetros, encontrando de esta forma la mejor configuración de hiperparámetros para cada modelo y repetir el proceso de evaluación, para comprobar si se mantiene la diferencia en rendimiento entre modelos. También se puede expandir el trabajo actual implementando otras métricas de distancia, u otras estrategias de agregación para ver si alguna obtiene mejor resultado que las evaluadas. Por otra parte, es posible evaluar nuevas variables y cómo estas afectan en las recomendaciones, como el método usado para hacer *clustering* y el permitir o no que usuarios pertenezcan a más de un grupo. Finalmente, se necesita evaluar el

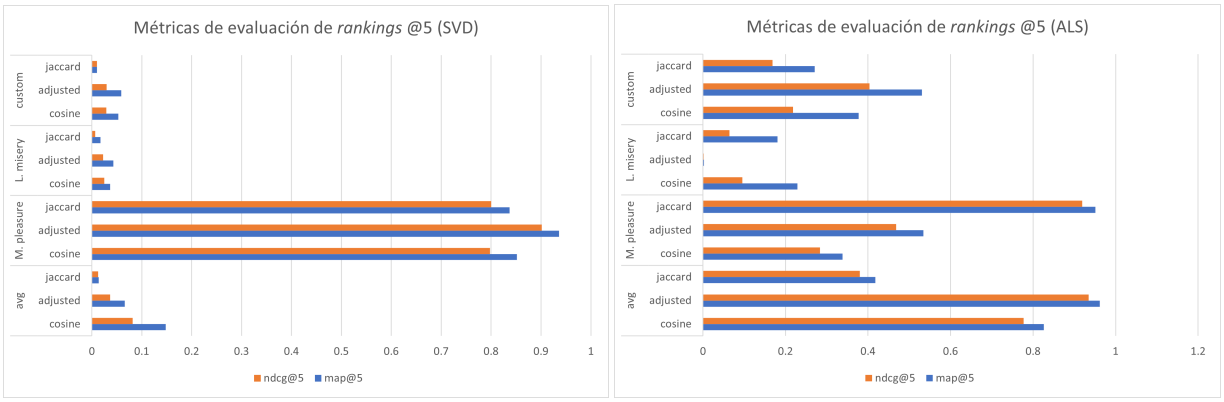

Figure 4: Resultados de NDCG@5 y MAP@5 para el modelo SVD y ALS.

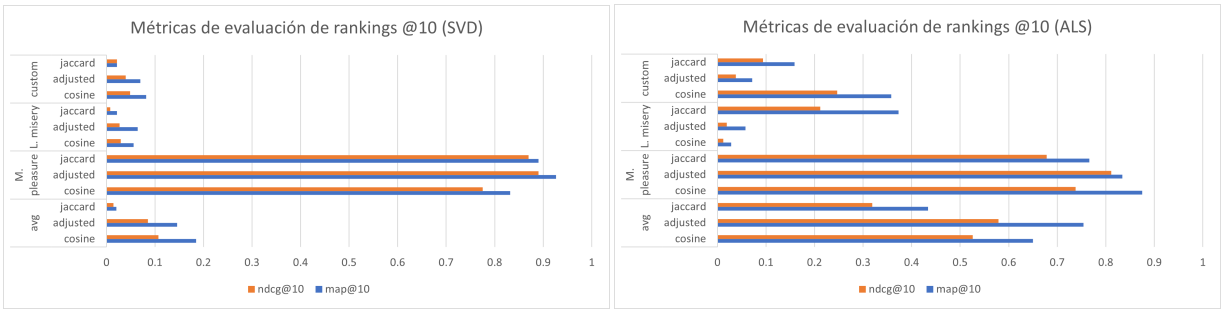

Figure 5: Resultados de NDCG@10 y MAP@10 para el modelo SVD y ALS.

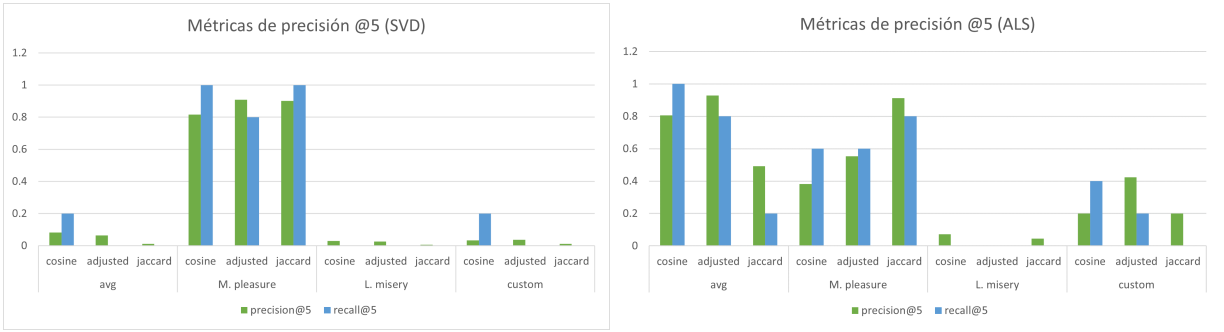

Figure 6: Resultados de *Precision*@5 y *Recall*@5 para el modelo SVD y ALS.

procedimiento realizado en otro *dataset* con distinta temática, con el fin de comprobar si los resultados obtenidos pueden aplicarse a cualquier contexto de recomendación a grupos, o son únicamente atingentes a la recomendación de películas.

REFERENCES

[1] Manoj K. Agarwal and David A. Reid. 2015. Predicting Group Choice: an Experimental Study Using Conjoint Analysis. In *Proceedings of the 1984 Academy of Marketing Science (AMS) Annual Conference*, Jay D. Lindquist (Ed.). Springer International Publishing, Cham, 445–449.

[2] Maciej Besta, Raghavendra Kanakagiri, Harun Mustafa, Mikhail Karasikov, Gunnar Rätsch, Torsten Hoefler, and Edgar Solomonik. 2019. Communication-Efficient Jaccard Similarity for High-Performance Distributed Genome Comparisons. *CoRR* abs/1911.04200 (2019). arXiv:1911.04200 http://arxiv.org/abs/1911.04200

[3] Ludovico Boratto and Salvatore Carta. 2011. *State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups.* Springer Berlin Heidelberg, Berlin, Heidelberg, 1–20. https://doi.org/10.1007/978-3-642-16089-9_1

[4] Cosley, Lam, Shyong K, John Tsibouklis, Albert, Istvan, Konstan, Joseph A, and Riedl. 2003. Is seeing believing?: how recommender system interfaces affect users' opinions. https://doi.org/10.1145/642611.642713

[5] Harsha Dara, Ravindranath Chowdary, and Chintoo Kumar. 2020. A survey on group recommender systems. *Journal of Intelligent Information Systems* (04 2020). https://doi.org/10.1007/s10844-018-0542-3

[6] A. Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič. 2018. *Group Recommender Systems - An Introduction.*

[7] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[8] Anthony Jameson and Barry Smyth. 2007. Recommendation to Groups, Vol. 4321. 596–627. https://doi.org/10.1007/978-3-540-72079-9_20

[9] Rosmamalmi Mat Nawi, Shahrul Azman Mohd Noah, and Lailatul Qadri Zakaria. 2020. Evaluation of Group Modelling Strategy in Model-Based Collaborative Filtering Recommendation. *International Journal of Machine Learning and Computing* 10 (02 2020), 330–338. https://doi.org/10.18178/ijmlc.2020.10.2.939

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[11] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2010. *Recommender Systems Handbook* (1st ed.). Springer-Verlag, Berlin, Heidelberg.

[12] Gabriel Sepulveda, Vicente Dominguez, and Denis Parra. 2017. pyRecLab: A Software Library for Quick Prototyping of Recommender Systems. arXiv:arXiv:1706.06291v2