



Métricas de similaridad y evaluación para sistemas de recomendación de filtrado colaborativo

Similarity and evaluation metrics for collaborative based recommender systems

Gustavo Mendoza Olgún

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México
lyon_db@hotmail.com

Yadira Laureano De Jesús

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México
star95ariday@hotmail.com

María Concepción Pérez de Celis Herrero

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, México
mcpcelis@gmail.com

doi: <https://doi.org/10.36825/RITI.07.14.019>

Recibido: Julio 17, 2019

Aceptado: Noviembre 14, 2019

Resumen: Los sistemas de recomendación son sistemas inteligentes que proporcionan a los usuarios una serie de sugerencias personalizadas sobre un determinado tipo de elementos (objetos). Para esto, se recaban por diferentes medios las características de cada usuario para, mediante un procesamiento de los datos, encontrar un subconjunto de ítems que pueden resultarle de interés. Mejorar la exactitud de las recomendaciones es una tarea crucial debido a la tendencia que han adoptado los sitios web, principalmente comerciales, de ofrecer a sus visitantes contenidos del catálogo que les podrían interesar acorde a sus necesidades o gustos. En este artículo se presenta un análisis comparativo entre algunas de las métricas de similitud y evaluación propuestas para los sistemas de recomendación basados en filtrado colaborativo; realizando pruebas sobre *datasets* comúnmente utilizados para determinar su eficiencia en producción.

Palabras clave: *Sistemas de Recomendación, Filtrado Colaborativo, Medidas de Similitud.*

Abstract: Recommender Systems are smart systems that bring to the users a set of personalized suggestions from an specific type of items(objects). In order to do this, several techniques are used to collect the user's characteristics for, using data processing, to find a subset of items that could be relevant to him. The improvement of the recommendation's accuracy is crucial because offering relevant content (based on needs or likes) to the visitors of web sites, mainly commercial ones, is trending. This article shows a comparative analysis among different similarity and evaluation metrics proposed for collaborative-filtering based recommender systems; doing tests on commonly used datasets to determine its efficiency on production time.

Keywords: *Recommender Systems, Collaborative Filtering, Similarity Metrics.*

1. Introducción

Cuando se utiliza un sitio de compras, un servicio de *streaming*, una red social o incluso para leer noticias en un portal, los usuarios requieren que los contenidos que les puedan interesar, estén a un clic de distancia. Lo anterior es posible si el sitio incorpora las técnicas adecuadas para analizar los gustos de sus usuarios y presentarles información que les sea relevante. Los sistemas de recomendación (SR) son software que combinan diferentes técnicas estadísticas para predecir cual contenido conviene mostrar a un usuario durante una sesión. La popularidad de estos sistemas radica en que conforme las recomendaciones mejoran en precisión, generan en el usuario un sentimiento de pertenencia; pues el sitio recuerda quién es y qué le gusta [1].

Existen diferentes técnicas para la implementación de un SR: pueden basarse en filtrado colaborativo, contenido, o conocimientos. De la técnica utilizada depende la información que será necesaria. Por ejemplo, si el sitio no maneja perfiles entonces puede usarse un sistema de recomendación que compare los ítems que el usuario ya posee contra los que se encuentren en el catálogo para presentar otros similares. Si se manejan entonces el SR puede elegir de entre su catálogo los ítems que serían interesantes a partir de lo que conoce del dueño del perfil. Usando otra técnica, el SR podría ofrecer nuevas sugerencias a partir de las calificaciones que otros usuarios con gustos similares hicieron sobre los objetos. O bien generar combinaciones de estos métodos para obtener cada vez predicciones más acertadas.

Cada una de estas formas de implementación tiene puntos fuertes y débiles que dependen de la información disponible. Por ejemplo, la falta de perfiles de usuario implicará que un SR basado en conocimiento no podrá ser utilizado y que un SR basado en contenido no tomará en cuenta los gustos de los usuarios, sino sólo las características de los objetos del catálogo. Si no se tienen calificaciones de los elementos a recomendar no podrá utilizarse un SR basado en filtrado colaborativo (FC) pues estos dependen de que los usuarios evalúen la mayor cantidad de ítems posibles del catálogo, de lo contrario la precisión de los resultados estará comprometida. Cada uno de los enfoques ha sido estudiado de manera exhaustiva y se ha refinado día a día. Es así que existe en la literatura una gama de técnicas para generar las recomendaciones basándose en diferentes herramientas matemáticas, desde la estadística hasta los campos vectoriales, comprobando la validez de los métodos y mejorando las métricas.

El objetivo de este trabajo es presentar un análisis de varias métricas de similitud y evaluación de SR colaborativos; por lo que inicialmente se hace una breve revisión de la evolución de estos sistemas, haciendo hincapié en las técnicas utilizadas para obtener la similitud y las métricas de evaluación propuestas por diferentes autores. En las secciones subsecuentes se describe primeramente el algoritmo general de un SR basado en un filtrado colaborativo *item-based* y a continuación los *datasets* utilizados para la implementación de los algoritmos. Finalmente se presenta la comparativa de los resultados obtenidos en pruebas de las técnicas mencionadas.

1.1. Justificación del trabajo

Se han presentado diversas implementaciones de sistemas de recomendación basados en filtrado colaborativo, cada una con sus propias métricas para similitud y evaluación. La efectividad de estos depende del dominio para el cual están diseñados. Esto significa que, cuando se tiene que crear una implementación para un catálogo de ítems diferente, debe seleccionarse la técnica que ofrezca una mayor precisión en sus recomendaciones. Generalmente, el usuario está acostumbrado a interactuar con los SR en redes sociales como Facebook [2], Twitter [3] o Pinterest [4]; en tiendas en línea como Amazon [5] o Ebay [6], o en sitios de *streaming* como Netflix [7] o Youtube [8]. En estos casos, aunque la precisión es apreciada, el obtener un resultado poco relevante tiene poco efecto en la decisión final del usuario. Es decir, si se recomienda un ítem no interesante, la consecuencia más catastrófica será que no se genere una nueva venta, en el caso de una tienda en línea, que no se interactúe con el contenido y no se muestre la publicidad asociada en el caso de una red social o bien que la película no sea vista en el caso de un servicio de video.

Sin embargo, si se desea llevar a los sistemas recomendadores a otras áreas como la medicina, la procuración de justicia o la educación, es claro que las necesidades de precisión y velocidad de respuesta en estas áreas es prioritario. Por ejemplo: si se desean recomendar fármacos basándose en interacciones medicamentosas, una mala recomendación puede tener efectos fatales en el paciente; si se desea recomendar una sentencia para un acusado, una mala consideración de alguna de las condiciones de la base de conocimientos puede hacer que se condene a un inocente o se libere a un culpable; en el caso de la educación, una mala recomendación de contenidos tendrá consecuencias en el aprendizaje de toda una generación de usuarios.

Por esto es importante tener un conocimiento adecuado de las diferentes métricas de similitud y precisión, de manera que pueda elegirse la más adecuada al considerar el tipo de ítems del catálogo y el rango de *ratings* que el usuario podrá asignar

2. Clasificación y funcionamiento de los sistemas de recomendación

Para abordar de lleno las métricas utilizadas en la implementación de SR es necesario entender cómo se componen y clasifican. La definición que presenta Dietmar en [1] es la que mejor engloba las diferentes definiciones existentes. En esta sección se presenta la definición formal de un SR, las clasificaciones de acuerdo al enfoque que pueden tener y los trabajos relacionados sobre los cuáles está basado en trabajo de las pruebas realizadas.

2.1. Definición y clasificación de los SR

Formalmente, puede definirse a un SR (ver Fig. 1) como una función que devuelve una lista de ítems ordenados por su utilidad con respecto al usuario u .

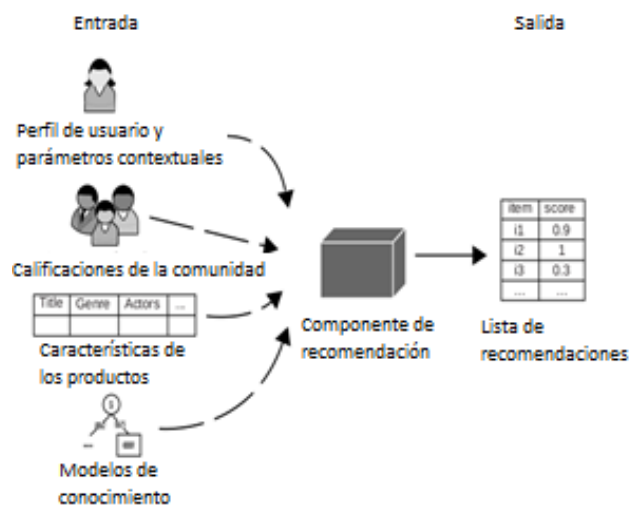


Figura 1. Esquema funcional de un RS. Fuente: [1].

Es decir: sea U el conjunto de usuarios del sistema, sea I el conjunto de ítems en el catálogo. Y sea $rec(u, i)$ la función de utilidad de $U \rightarrow I \cdot \mathbb{R}$ en $[0,1]$

$$RS(u, n) = \{i_1, i_2, i_3, \dots, i_n\}$$

donde $i_1, i_2, i_3, \dots, i_n \in I$ y además

$$\forall k \text{ } rec(u, i_k) > 0 \wedge rec(u, i_k) > rec(u, i_{k+1})$$

El mismo autor en [1], de acuerdo al enfoque utilizado para obtener la lista de recomendaciones, clasifica a los SR en:

- Basados en contenido: realizan recomendaciones basados en similitudes de los ítems de un catálogo. Dichas recomendaciones pueden realizarse usando calificaciones que han recibido los objetos de forma anónima o bien considerando sus características. Se basan en la premisa de que, si una persona escogió un elemento, entonces le gustarán otros similares. Estos sistemas tienen problemas con los usuarios nuevos, pues requieren saber los intereses del usuario en los objetos del catálogo para poder realizar su trabajo.
- Basados en conocimiento: realizan recomendaciones basados en la información que conocen a priori o por medio de la interacción con el usuario. Se basan en la premisa de satisfacer necesidades en vez de intereses. Estos sistemas requieren de una base de conocimientos que defina los criterios que los ítems deben cumplir para poder ser recomendados. En la mayoría de las implementaciones estos sistemas utilizan una estructura taxonómica facetada para multicategorizar los elementos del catálogo. Si ninguno de los ítems coincide con los requerimientos solicitados entonces puede procederse a hacer un relajamiento de alguna restricción o interactuar con el usuario para establecer otro encuadre. Tienen la desventaja de que requieren de un discriminante para ponderar las recomendaciones cuando existe más de un artículo que cumpla con las especificaciones solicitadas.
- Basados en colaboración: realizan recomendaciones basados en las calificaciones que los usuarios han hecho sobre los ítems. Estos sistemas trabajan sobre la matriz de calificaciones de los objetos y se basan en la premisa de que debido a que a X le han gustado estos elementos porque los ha calificado de esta manera, entonces le gustarán las cosas que a otros han calificado de forma similar. Estos sistemas tienen problemas con los comportamientos de las calificaciones, pues puede ser que no haya ninguna o bien fueron asignadas al azar; así también con los nuevos elementos en el catálogo puesto que no han recibido ninguna calificación.
- Híbridos: combinan diferentes enfoques para generar las recomendaciones. Pueden ser basados en conocimiento y contenido, para subsanar las debilidades de cada uno de los métodos por separado, o bien basados en colaboración y conocimiento, o inclusive combinar varios métodos para obtener mejores recomendaciones.

Sin importar la técnica a seguir, el proceso de obtención de recomendaciones se basa en similaridad. La similaridad es un valor (generalmente de -1 a 1) que indica cuan relacionados están dos componentes. Generalmente se usan las medidas de distancia entre ítems para representar esta medida, por ejemplo: la correlación de Pearson, la distancia cuadrática media, la distancia Manhattan, la distancia del coseno, la distancia del coseno ajustado, etc. Cada una de estas ha sido propuesta por diferentes investigadores como una forma de definir mejores relaciones entre los elementos basados en la información que se posee; aquellos más parecidos serán los que estarán involucrados dentro del cálculo de las recomendaciones.

Los SR basados en FC pueden encontrarse con situaciones que no permiten su correcto funcionamiento. Algunas de ellas son:

- Matriz de *ratings* dispersa: Esto sucede cuando los usuarios no califican los ítems que van visitando. Resulta en un cálculo de similaridad inexacto, o en casos como Pearson, no pueden ser calculados debido a que no existen objetos evaluados en común.
- Ítems o usuarios sin *ratings*: Cuando el sistema de recomendación no dispone de suficiente información acerca de un usuario o contenido, es muy difícil poder realizar recomendaciones, y sobre todo, que sean válidas para él. El problema del arranque en frío se debe a la existencia de ítems que nadie ha valorado explícita o implícitamente dentro de un catálogo. En general, si no se asigna una calificación no es posible hacer inferencia sobre el interés o gusto de los usuarios.
- *Ratings* inexactos: Esto sucede cuando el usuario evalúa siempre con el mismo puntaje, ya sea muy alto o muy bajo. En este caso es posible que las recomendaciones obtenidas no sean de relevantes, dando lugar a falsos positivos o falsos negativos.
- Usuarios que han calificado todo: Puede existir alguien que haya calificado todos los ítems del catálogo, en este caso no será posible hacer una recomendación, puesto que las recomendaciones se hacen sobre elementos que aún no han sido valorados.

2.2. Trabajos relacionados

La tesis de Moya García [9] realiza una revisión del estado de los SR basados en FC, estudia e implementa la técnica de descomposición en valores singulares (SVD) en el contexto de este tipo de sistemas. Utiliza el *dataset* de Movielens y compara con las métricas tradicionales de los k -vecinos obtenidas a través de diferentes métricas (la diferencia cuadrática media, la correlación de Pearson y el coseno). A final, aporta mejoras a los algoritmos para que estos obtengan predicciones en un tiempo menor.

En el artículo de Sarwar, Karypis, Konstan y Riedl [10] se presenta un análisis de técnicas de recomendación de filtrado colaborativo basados en ítems. Examinan diferentes formas para calcular similitudes de ítems (correlación de ítems frente a similitudes de coseno entre vectores de ítems) y para obtener recomendaciones (suma ponderada frente a modelo de regresión). Finalmente, evalúan de manera experimental los resultados, donde utilizan el *dataset* de MovieLens y los comparan con el enfoque básico del vecino más cercano. Los experimentos sugieren que los algoritmos basados en ítems proporcionan un rendimiento dramáticamente mejor que aquellos basados en usuarios mientras que, al mismo tiempo, proporcionan una mejor calidad de recomendaciones.

Gunawardana y Shani presentan en [11] una revisión de las métricas de evaluación (raíz del error cuadrático medio, *precision-recall*, error medio acumulado), donde sugieren un enfoque para decidir cuál de ellas es la más adecuada para una aplicación dada de acuerdo a la tarea que cumplirá el SR, analizando su rendimiento. En su trabajo demuestran que el uso de una métrica incorrecta puede llevar a la selección de un algoritmo inadecuado para la tarea de interés del SR. También discuten otras consideraciones importantes al diseñar experimentos fuera de línea experimentando con una amplia colección de conjuntos de datos y comparan los resultados obtenidos por diferentes técnicas de evaluación, mostrando que éstas clasifican a los algoritmos de manera diferente.

Haifeng, Zheng, Mian, Tian y Zhu [12] presentan un nuevo modelo para el cálculo de la similitud de usuarios con el objetivo de mejorar el rendimiento del SR cuando hay pocas calificaciones disponibles para computar similitudes. El modelo no solo considera la información de contexto local como las calificaciones otorgadas a los ítems, sino también la preferencia global del comportamiento del usuario. Los experimentos en tres conjuntos de datos reales se implementan y se comparan con muchas medidas de similitud de vanguardia. Los resultados muestran la superioridad del nuevo modelo de similitud en el rendimiento recomendado.

Ramírez Morales [13] presenta una implementación del algoritmo de descomposición en valores singulares (SVD) junto con técnicas de reducción de dimensionalidad y de cálculo de mínimos, a partir del enfoque dado por Simon Funk para la implementación de SR en el comercio. Se realizaron pruebas con el *dataset* de Movielens, de convergencia y error medio absoluto. Para la realización de las pruebas se utilizó el lenguaje de programación Python.

Castellano, Martínez, Barranco y Pérez [14] presentan una evaluación de la validez del uso del filtrado colaborativo como herramienta para orientar al alumnado a la hora de tomar decisiones que impliquen alguno de los siguientes puntos: elección individual de materias, elección de perfiles o modalidades académicas, e incluso detección de asignaturas con potenciales problemas y necesidades de refuerzo para el individuo. Basándose en el trabajo realizado con 744 alumnos y con asignaturas comprendidas entre valores enteros entre el 0 y el 10, se construye OriEB, un sistema encargado de orientar académicamente al alumnado cuando se enfrente a la complicada decisión de elegir un perfil académico y unas materias a cursar en su siguiente etapa educativa.

En el trabajo de Castellano y Rojas [15] se presenta una investigación sobre el análisis de técnicas de recomendación de filtrado colaborativo en referencia a los beneficios de una comunidad. Se realiza el análisis de algoritmos tanto basados en usuarios como en ítems, seleccionando aquellos mejor evaluados utilizando un conjunto de datos propio. También se identifican los principales problemas de las técnicas seleccionadas y se proponen soluciones. El sistema desarrollado es incorporado a la plataforma VideoWeb 1.0 que usa el CMS Drupal en la versión 6. Los resultados se calculan utilizando el método de error absoluto medio con un rango de 50 a 200 vecinos. La integración del módulo de recomendaciones a la plataforma proporciona un aumento en la personalización del contenido multimedia publicado para satisfacer las preferencias de cada usuario.

Del Pino, Salazar y Cedeño presentan en su trabajo [16] la adaptación de un SR de filtrado colaborativo basado en el usuario a uno basado en el historial. Mediante el análisis de dos métricas de similitud: *Tanimoto* y *LogLikelihood* en un conjunto de pruebas propio, miden la efectividad de éstas utilizando dos algoritmos distintos de similitud; con el objetivo de recomendar materias a un estudiante del sexto semestre de la carrera de Ingeniería en Electrónica y Telecomunicaciones. Los resultados muestran que es factible adaptar un SR colaborativo existente a un modelo basado en el historial del usuario.

En su trabajo, Zeng, Zhu, Lüb y Zhou [17] aplican una versión modificada de la llamada inferencia basada en red (NBI), que puede distinguir las contribuciones de las calificaciones positivas y negativas. De acuerdo con el extenso análisis numérico en tres conjuntos de datos de referencia, MovieLens, Netflix y Amazon, obtienen como resultados en el nivel macroscópico que las calificaciones negativas son más valiosas para los conjuntos de datos más dispersos, y en el nivel microscópico que las calificaciones negativas de los usuarios inactivos sobre objetos menos populares son más valiosas. Finalmente, se destaca la relevancia significativa de los resultados para los dos desafíos a largo plazo en el filtrado de la información: el problema de escasez y el problema de arranque en frío.

En el artículo de Taghavi, Bentahar, Bakhtiyari y Hanachi [18] se presenta una investigación estadística sobre los SR publicados por *Web of Science*. Para obtener una visión general del estado del arte, se introducen taxonomías impulsadas por las siguientes cinco fases: iniciación (técnicas de arquitectura y adquisición de datos), diseño (tipos y técnicas de diseño), desarrollo (métodos de implementación y algoritmos), evaluación (métricas y técnicas de medición) y aplicación (dominios de aplicaciones). Se proporcionan algunas ideas inspiradoras de la economía computacional y el aprendizaje automático para que los investigadores amplíen los aspectos novedosos de los SR.

La publicación de Giorgos Papachristoudis [19] analiza las métricas de evaluación de relevancia en los sistemas de SR, se habla sobre *recall*, *precision* y la necesidad de utilizar una métrica objetiva única para comparar entre diferentes modelos. Finalmente, se explica el concepto de precisión promedio que toma en cuenta no solo la relevancia sino también el orden relativo de una recomendación relevante y se proporciona el código *TensorFlow* para todas las métricas analizadas.

3. Métricas de similitud y evaluación de los sistemas de recomendación colaborativos

En esta sección se presentan algunas de las métricas presentes en las implementaciones de SR colaborativos. En la primera subsección se presentan las medidas de similitud y su aplicación. En la segunda se muestran las técnicas generadoras de recomendaciones más usuales. En la última, se mencionan las métricas de evaluación más aplicadas y su significado dentro del contexto de los SR.

3.1. Medidas de similitud

Un paso crítico en el algoritmo de FC basado en elementos es calcular qué tan parecidos son los elementos y luego seleccionar los que resulten más similares. La idea básica es escoger dos objetos i y j , aislar a los usuarios que los han calificado en común y luego aplicar una técnica para determinar la similitud s_{ij} . La Fig. 2 ilustra este proceso; aquí las filas de la matriz representan a los usuarios y las columnas representan los ítems del catálogo.

Para que pueda calcularse la similitud entre dos usuarios u_i y u_j , deben cumplirse dos condiciones: los usuarios deben tener ítems calificados y al menos un elemento debe haber sido evaluado en común; si alguna de estas condiciones no se cumple, entonces debe trabajarse con otras técnicas que permitan realizar recomendaciones, por ejemplo, utilizar una base de conocimientos, o crear un perfil del usuario.

Varios autores coinciden en presentar las siguientes medidas de similitud en común, considerándolas las más utilizadas en tiempo de producción ([9], [10], [11]).

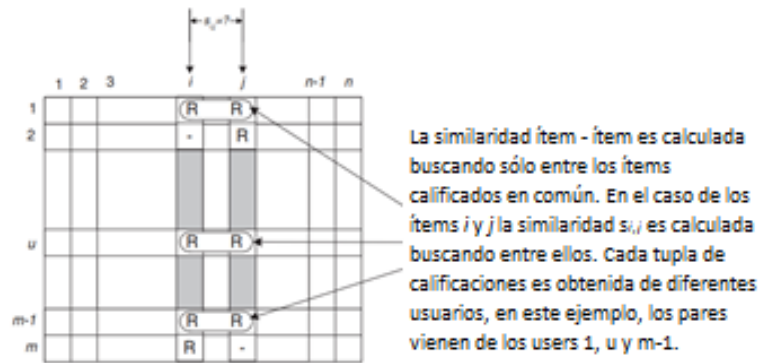


Figura 2. Aislamiento de los elementos valorados conjuntamente y cálculo de similitud. Fuente [9].

3.1.1. Diferencia cuadrática media (MSD)

Esta medida de similitud computa la diferencia euclidiana entre dos vectores de votos. La Ecuación 1 muestra la expresión a utilizar para el cálculo de la similitud cuadrática media [9].

$$sim(x, y) = 1 - \frac{1}{\#B_{x,y}} \sum \left(\frac{r_{x,i} - r_{y,i}}{max - min} \right)^2 \in [0,1] \quad (1)$$

Donde $B_{x,y}$ es el conjunto no vacío de ítems valorados en común por los usuarios x y y .

3.1.2. Similitud basada en coseno

En este caso, los elementos se consideran dos vectores en el espacio de usuario dimensional [9]. La similitud entre ellos se mide calculando el coseno del ángulo formado por la intersección de las representaciones vectoriales.

Formalmente, en la matriz de calificaciones $M \times N$ la similitud del coseno entre los elementos i y j se calcula como se muestra en la Ecuación 2.

$$sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2} \quad (2)$$

Donde “.” denota el producto punto de los dos vectores.

3.1.3. Similitud basada en correlación

En este caso, los objetos i y j se comparan calculando el coeficiente de correlación de Pearson [9]. Para que el cálculo sea preciso, primero se deben aislar los casos de evaluación conjunta (es decir, los casos en que los usuarios calificaron tanto i como j). Sea U el conjunto de usuarios que calificaron los ítems i y j entonces la similitud basada en correlación, se calcula como se expresa en la Ecuación 3.

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}} \quad (3)$$

Donde $R_{u,i}$ denota la calificación del usuario u en el artículo i , \bar{R}_i es la calificación promedio del i -ésimo elemento.

3.1.4. Similitud de coseno ajustado

Una diferencia fundamental entre el cálculo de similitud en el FC basado en el usuario y el basado en ítems es que, en el caso de primera, se calcula a lo largo de las filas de la matriz; en el caso de la segunda, la similitud se

calcula a lo largo las columnas, es decir, cada par en el conjunto de clasificación compartida corresponde a un usuario diferente. La técnica del coseno básica en un caso basado en artículos tiene un inconveniente importante: las diferencias en la escala de calificación entre diferentes evaluadores no se tienen en cuenta. La similitud de coseno ajustada compensa este inconveniente al restar el promedio de las calificaciones del usuario correspondiente de cada par clasificado conjuntamente [10]. Formalmente, la similitud entre los elementos i y j , se calcula como se expresa en la Ecuación 4.

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}} \quad (4)$$

Donde \bar{R}_u es el promedio de las calificaciones asignadas por el usuario u .

3.1.5. Coeficiente de similitud de Tanimoto (Jaccard)

El coeficiente de Tanimoto [12] es la cantidad de elementos por la que dos usuarios expresan cierta preferencia, dividida por la cantidad de elementos por los cuales el usuario expresa cierta preferencia (llamado también coeficiente de Jaccard).

En otras palabras, es la relación entre el tamaño de la intersección y el tamaño de la unión de los elementos preferidos de ambos usuarios. Puede utilizarse como una medida de similitud debido a que cumple con las propiedades requeridas: cuando los elementos favoritos de dos usuarios se superponen completamente, el resultado es 1.0. Cuando no tienen nada en común, es 0.0. Es posible ampliar los resultados en el rango de -1 a 1 con algunas matemáticas simples.

3.1.6. Similitud basada en Log-likelihood

Al igual que el coeficiente de Tanimoto, se basa en el número de elementos en común entre dos usuarios; pero su valor es más una expresión de lo poco probable que sea que dos usuarios tengan tanta superposición dado el número total de elementos que hay y el número de elementos que cada usuario tiene como preferencia [12].

Por ejemplo, supóngase dos fanáticos del cine que han visto y calificado varias películas, pero que solo han visto *Star Wars* y *Casablanca* en común. ¿Serán similares?

Si cada uno ha visto cientos de películas, no significaría mucho. Muchas personas las han visto, y si estos dos usuarios han visto muchas películas, pero solo han logrado superponerse en estas dos, probablemente no sean similares. Por otro lado, si cada uno ha visto solo unas pocas películas, y estas dos estaban en las listas de ambos, parecería implicar que son de gustos similares cuando se trata de películas; y en este caso la superposición sería significativa. El *Log-likelihood* evalúa si esta superposición entre los dos usuarios se debe al azar. Es decir, dos usuarios diferentes sin duda coincidirán para calificar un par de películas en común, pero dos similares mostrarán una superposición que parece poco probable que sea una mera casualidad. El valor de similitud resultante puede interpretarse como una probabilidad de que la superposición no se deba al azar.

3.1.7. Coeficiente de correlación de Pearson restringido (CPCC)

En [13] se presenta una variación de la correlación de Pearson que considera el impacto de las calificaciones positivas y negativas, la Ecuación 5 muestra el modelo a utilizar.

$$sim(u, v)^{CPCC} = \frac{\sum_{p \in I} (r_{u,p} - r_{med})(r_{v,p} - r_{med})}{\sqrt{\sum_{p \in I} (r_{u,p} - r_{med})^2} \sqrt{\sum_{p \in I} (r_{v,p} - r_{med})^2}} \quad (5)$$

Donde r_{med} es el valor de la mediana en la escala de calificación. Por ejemplo, r_{med} es 3 en una escala del 1 al 5 y es 4 en una escala del 1 al 7.

3.1.8. El coeficiente de correlación de Pearson ponderado (WPCC) y el coeficiente de correlación de Pearson basado en la función sigmoidea (SPCC)

La literatura presenta entre otras, dos variaciones a la correlación de Pearson. La versión ponderada se refiere a la utilización de pesos asignados a los sujetos en el cálculo de un coeficiente de correlación entre dos variables X e Y. Los pesos pueden estar disponibles de forma natural de antemano o pueden ser elegidos por el usuario para cumplir un propósito específico [13], [14], [20].

La segunda variación, basada en la función sigmoidea, (la cual se propuso para evitar favorecer el tamaño de los usuarios comunes) considera el tamaño del conjunto de los usuarios comunes en la métrica de similitud del elemento para reducir el error [13]. Las dos variaciones para el coeficiente de correlación de Pearson: ponderado (WPCC) y basado en la función sigmoidea (SPCC), se muestran en la Ecuación 6.

$$\begin{aligned} sim(u, v)^{WPCC} &= \begin{cases} sim(u, v)^{PCC} \cdot \frac{|I|}{H}, & |I| \leq H \\ sim(u, v)^{PCC}, & \text{en otro caso} \end{cases} \\ sim(u, v)^{SPCC} &= sim(u, v)^{PCC} \cdot \frac{1}{1 + \exp\left(-\frac{|I|}{2}\right)} \end{aligned} \quad (6)$$

Donde H es un valor experimental y se establece en 50 para la mayoría de los casos prácticos.

3.1.9. Proximity, Impact and Popularity (PIP)

Esta medida de similitud heurística se compone de tres factores de similitud: *Proximity*, *Impact* y *Popularity (PIP)*, es propuesta en [13] y la fórmula se presenta en la Ecuación 7.

$$sim(u, v)^{PIP} = \sum_{p \in I} PIP(r_{u,p}, r_{v,p}) \quad (7)$$

El primer factor, *Proximity*, no solo calcula la diferencia absoluta entre dos calificaciones, sino que también considera si éstas están de acuerdo o no, dando penalización a las evaluaciones en desacuerdo. El factor *Impact* representa la fuerza con que los usuarios prefieren o rechazan un elemento. Si dos de ellos han calificado 5 en un artículo, mostrarán una preferencia más fuerte que la calificación 4. Debe observarse que se penaliza repetidamente en el cálculo de estas métricas cuando dos calificaciones no están de acuerdo. El último factor es *Popularity*, el cual denota qué tan comunes son las puntuaciones de dos usuarios: si el *rating* promedio de ambos evaluadores tiene una gran diferencia con el promedio de las calificaciones totales de los usuarios, entonces los puntajes pueden proporcionar más información sobre la similitud de aquellos que las asignaron.

La Ecuación 8 ilustra el cálculo de la similitud a partir de los tres factores antes mencionados.

$$PIP(r_{u,p}, r_{v,p}) = Proximity(r_{u,p}, r_{v,p}) \cdot Impact(r_{u,p}, r_{v,p}) \cdot Popularity(r_{u,p}, r_{v,p}) \quad (8)$$

3.1.10. Proximity, Significance, Singularity (PSS)

De forma parecida a la métrica anterior, PSS se compone de tres factores de similitud: *Proximity*, *Significance* y *Singularity* [13]. El primer factor solo considera la distancia entre dos *ratings*. El segundo supone que las calificaciones son más significativas si éstas están más alejadas de la media. El último representa cómo un par de

valoraciones son diferentes con respecto a todas las demás. Las formalizaciones de los tres factores se definen en la Ecuación 9.

$$\begin{aligned} Proximity(r_{u,p}, r_{v,p}) &= 1 - \frac{1}{1 + e^{-|r_{u,p} - r_{v,p}|}} \\ Significance(r_{u,p}, r_{v,p}) &= \frac{1}{1 + e^{-|r_{u,p} - r_{v,p}|} + |r_{u,p} - r_{v,p}|} \\ Singularity(r_{u,p}, r_{v,p}) &= 1 - \frac{1}{1 + e^{-\left|\frac{r_{u,p} + r_{v,p}}{2} - \mu_p\right|}} \end{aligned} \quad (9)$$

Donde μ_p es la calificación promedio del ítem p y $r_{u,p}$ es el puntaje dado al artículo p por el usuario u . Cada factor puede tomar valores entre $[0,1]$.

Conociendo los factores, la similitud PSS de dos elementos se puede calcular con en la Ecuación 10.

$$sim(u, v)^{PSS} = \sum_{p \in I} PSS(r_{u,p}, r_{v,p}) \quad (10)$$

Donde $PSS(r_{u,p}, r_{v,p})$ es el valor del producto de los factores entre dos puntuaciones de los usuarios u y v para el mismo ítem, es decir:

$$\begin{aligned} PSS(r_{u,p}, r_{v,p}) &= Proximity(r_{u,p}, r_{v,p}) \\ &\quad \cdot Significance(r_{u,p}, r_{v,p}) \\ &\quad \cdot Singularity(r_{u,p}, r_{v,p}) \end{aligned} \quad (11)$$

3.1.11. New Heuristic Similarity Model (NHSM)

En [13] se propone una nueva medida de similitud basada en una combinación de otras medidas de similitud que han demostrado su funcionamiento en determinadas características de los ratings de los usuarios. Esencialmente, esta nueva medida de similitud está definida por la Ecuación 12.

$$sim(u, v)^{NHMS} = sim(u, v)^{JPSS} \cdot sim(u, v)^{URP} \quad (12)$$

Donde la similitud JPSS es una combinación de la similitud PSS y la similitud de Jaccard modificada, Los modelos se mencionan en las Ecuaciones 13 y 14 respectivamente.

$$sim(u, v)^{Jaccard} = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \quad (13)$$

Donde I es el conjunto de ítems evaluados por los usuarios u y v respectivamente.

$$sim(u, v)^{JPSS} = sim(u, v)^{PSS} \cdot sim(u, v)^{Jaccard} \quad (14)$$

Para tomar en cuenta las diferentes preferencias en las calificaciones, se hace uso de la medida de similitud URP. Esta toma en consideración que algunas personas prefieren dar altas calificaciones; otras tienden a valorar bajos valores. Para reflejar esta preferencia de comportamiento, se utiliza la media y la varianza de la calificación para modelar la preferencia del usuario. El modelo utilizado se presenta en Ecuación 15.

$$sim(u, v)^{URP} = 1 - \frac{1}{1 + e^{-|\mu_u - \mu_v| |\sigma_u - \sigma_v|}} \quad (15)$$

Donde μ_u y μ_v es la media de las evaluaciones asignadas por los usuarios u y v respectivamente. Por su parte, σ_u y σ_v representan la varianza estándar del usuario u y v . La calificación promedio de los usuarios está dada por la Ecuación 16 y la varianza estándar de los usuarios por la Ecuación 17.

$$\mu_u = \sum_{p \in I_u} \frac{r_{u,p}}{|I_u|} \quad (16)$$

$$\sigma_u = \sqrt{\sum_{p \in I_u} \frac{(r_{u,p} - \bar{r}_u)^2}{|I_u|}} \quad (17)$$

3.1.12. Descomposición en valores singulares (SVD)

La descomposición en valores singulares (*Singular Value Decomposition, SVD*) es una técnica de factorización de matrices que permite descomponer una matriz A en otras matrices U , S , V , como se ilustra en la Ecuación 18 [15], [21].

$$SVD(A) = UxSxV^t \quad (18)$$

Es decir, el producto matricial de U por S por V^t da como resultado la matriz A . Con respecto a las dimensiones se parte de que A será de $N \times M$ donde n representa las filas y m representa las columnas. U será una matriz ortogonal que tendrá dimensiones de $N \times N$; V será de $M \times M$, también ortogonal. S (ver Fig. 3), será una matriz diagonal de tamaño $M \times N$ la cual tendrá lo que se denomina valores singulares, puestos en orden decreciente siendo estos mayores o iguales a cero, es decir:

$$S = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$$

Figura 3. Matriz S . Fuente: [15].

En otras palabras, el procedimiento de descomposición de la matriz A , se describe en la Ecuación 19:

$$SVD(A)_{n \times m} = U_{n \times n} x S_{n \times m} x V_{m \times m}^t \quad (19)$$

La validez de SVD en los sistemas de recomendación, está basada en que, al reducir el número de valores singulares de la matriz S a los primeros k valores, se obtendrá una aproximación de la matriz original A (ver Ecuación 20), lo que permite reconstruirla a partir de las versiones reducidas de las otras matrices cometiendo un cierto error, pero disminuyendo el tamaño. Es decir:

$$SVD(A)_{n \times m} \cong U_{n \times k} x S_{k \times k} x V_{k \times m}^t \quad (20)$$

Esta importante propiedad es derivada del teorema de Eckart-Young que aborda la mejor aproximación a la matriz original A , obteniéndola al igualar a 0 los n valores singulares más pequeños, reduciendo las matrices al número de valores singulares no nulos que tenga la matriz S . Esto resulta entonces en la transformación de gran cantidad de datos en su representación reducida, siendo por lo tanto una propiedad muy importante que permite reducir considerablemente el tiempo de cómputo y de uso de memoria para las tres matrices.

3.2. Técnicas generadoras de recomendaciones.

Una vez obtenida la similaridad entre los usuarios, es posible entonces predecir a partir de los vecinos cercanos cuáles de los ítems que el usuario actual aún no ha calificado deben recomendarse. Para esto, un SR debe predecir

la calificación que sería asignada al ítem en caso de evaluarlo, para decidir si lo sugiere o no. Diferentes técnicas pueden ser aplicadas, aunque las más comunes son tres: Promedio aritmético, suma ponderada y suma ponderada con respecto a la media aritmética.

En cualquiera de estas técnicas, serán los puntajes de los usuarios similares al actual en el ítem i los que ayuden a predecir la calificación que el segundo podría asignar al objeto en cuestión, siempre que no haya sido calificado previamente.

3.2.1. La media

La media calcula el promedio de todos los valores de confianza inferidos por cada una de las rutas alternativas de acuerdo con la Ecuación 21. A pesar de que este enfoque es muy rentable, se considera demasiado ingenuo [10], [16], [17], [18].

$$p_{u,i} = \frac{1}{G_{u,i}} \cdot \sum_{n \in G_{u,i}} r_{n,i} \text{ cuando } G_{u,i} \neq \emptyset \quad (21)$$

3.2.2. La media ponderada

En la media ponderada se calcula el promedio ponderado de la confianza inferida por las rutas alternativas, utilizando como pesos la confianza propagada de cada asociación inferida entre los usuarios, como se muestra en la Ecuación 22. Este enfoque es más sofisticado ya que se toma la confianza de la ruta en consideración. El cálculo final estará sesgado a la confianza inferida por la ruta más segura. [10], [16], [17], [18].

$$p_{u,i} = \frac{\sum_{n \in G_{u,i}} \text{sim}(u,n) \cdot r_{n,i}}{\sum_{n \in G_{u,i}} \text{sim}(u,n)} \text{ cuando } G_{u,i} \neq \emptyset \quad (22)$$

3.2.3. La media ponderada de la desviación con respecto a la media

La media ponderada de la desviación con respecto a la media [18], calcula la suma ponderada de las desviaciones de la media de una variable. Se muestra en la Ecuación 23.

$$p_{u,i} = \bar{r}_u + \frac{\sum_{n \in G_{u,i}} \text{sim}(u,n) \cdot (r_{n,i} - \bar{r})}{\sum_{n \in G_{u,i}} \text{sim}(u,n)} \text{ cuando } G_{u,i} \neq \emptyset \quad (23)$$

3.3. Métricas de evaluación

Para poder analizar la validez de las recomendaciones es necesario establecer métricas que permitan conocer qué tan alejados están los valores predichos por el algoritmo. Las más utilizadas son: Error medio absoluto (MAE), *coverage*, precisión y *recall*.

3.3.1. Error medio absoluto (MAE)

El error medio absoluto, también denominado MAE (*Mean Absolute Error*), calcula la distancia absoluta existente entre las predicciones realizadas y la votación real del usuario.

Para su cálculo [10],[22] se utiliza la Ecuación 24, donde C_u es el conjunto de ítems valorados por el usuario u para los que ha sido posible obtener la predicción.

$$C_u = \{i \in I | r_{u,i} \neq * \wedge p_{u,i} \neq *\} \quad (24)$$

El MAE de un usuario es el valor medio de la diferencia absoluta entre las predicciones realizadas y las votaciones reales del usuario. Para su cálculo se utiliza la Ecuación 25.

$$MAE_u = \frac{1}{\#C_u} \cdot \sum_{i \in C_u} |r_{u,i} - p_{u,i}| \quad (25)$$

El MAE del sistema se define como el promedio del error medio absoluto cometido en cada usuario conforme la Ecuación 26.

$$MAE_s = \frac{1}{\#U} \cdot \sum_{u \in U} MAE_u \quad (26)$$

3.3.2. Coverage

Esta medida tiene como finalidad medir la capacidad de recomendación que tienen los k-vecinos de cada usuario [10]. Dicho con otras palabras, indica el porcentaje de ítems que han podido ser predichos respecto del total de los que podían haberlo sido. Esta métrica se calcula como se indica en la Ecuación 27.

$$coverage_u = 100 \cdot \frac{\#\{i \in I | r_{u,i} = * \wedge p_{u,i} \neq *\}}{\#\{i \in I | r_{u,i} = *\}} \quad (27)$$

Por su parte, el *coverage* del sistema es computado con la Ecuación 28 [10], como el valor promedio de la métrica para cada usuario (ver Ecuación 28):

$$coverage = \frac{1}{\#U} \cdot \sum_{u \in U} coverage_u \quad (28)$$

3.3.3. Precisión y recall

La calidad de las recomendaciones realizadas puede medirse mediante dos métricas que, habitualmente, aparecen unidas que son la precisión y el *recall*. La precisión busca comprobar cuantos de los ítems recomendados al usuario activo eran realmente relevantes para él. El *recall* se encarga de comprobar qué porcentaje de los elementos significativos para un usuario le fueron efectivamente recomendados. Para poder definir cuando un objeto es o no relevante se emplea un parámetro Θ que especifica la nota mínima que debe tener para ser considerado como interesante.

Sean X'_u el conjunto de ítems para los cuales el usuario no asignó una calificación y para los cuales pudo obtenerse una recomendación, y Z'_u un subconjunto de X'_u que contiene los elementos significativos para el usuario que fueron correctamente recomendados [10], estos se definen en la Ecuación 29.

$$\begin{aligned} X'_u &\subset I \wedge \forall i \in X'_u, r_{u,i} \neq *, p_{u,i} \neq * \\ Z'_u &\subseteq X'_u, \quad Z'_u = N, \forall i \in Z'_u, \forall j \in (X'_u - Z'_u): p_{u,i} \geq p_{u,j} \end{aligned} \quad (29)$$

Si se definen los siguientes conjuntos para el usuario u : $Y_u = \{i \in Z'_u | r_{u,i} \geq \theta\}$ para ítems recomendados y que son significativos; $N_u = \{i \in Z'_u | r_{u,i} < \theta\}$ para ítems recomendados que no son relevantes y $O_u = \{i \in (I_u - Z'_u) | r_{u,i} \geq \theta\}$ para ítems no recomendados y que son de interés, es posible entonces definir la precisión de un usuario u como el porcentaje de ítems relevantes entre los recomendados, como se muestra en la Ecuación 30.

$$precisión_u = \frac{\#Y_u}{\#Y_u + \#N_u} \quad (30)$$

La precisión del sistema se calcula como se indica en la Ecuación 31, resulta ser el valor promedio de la precisión para cada usuario.

$$precisión = \frac{1}{\#U} \cdot \sum_{u \in U} precisión_u \quad (31)$$

El *recall* de un usuario u será el porcentaje de ítems relevantes recomendados respecto del total de ellos (ver Ecuación 32)

$$recall_u = \frac{\#Y_u}{\#Y_u + \#O_u} \quad (32)$$

Finalmente, la métrica general del sistema puede calcularse como el valor promedio del *recall* de cada usuario, como se muestra en la Ecuación 33.

$$recall = \frac{1}{\#U} \cdot \sum_{u \in U} recall_u \quad (33)$$

4. Metodología y resultados

El conjunto de datos utilizado en la realización de las pruebas de FC basado en ítems fue obtenido del *dataset* de calificaciones de chistes del sitio web *Jester* [23]. Se seleccionaron aleatoriamente dos subconjuntos con las siguientes características: para el primero se escogieron 100 registros con una calificación de -10 a 10 para cada chiste leído con la característica de ser denso, es decir, todos los ítems han sido evaluados por al menos un usuario y todos éstos han calificado más de un chiste. El segundo tiene la misma dimensión y rango de puntajes, pero es poco denso.

Se procedió a construir módulos en Python para las métricas de similitud: Pearson, Coseno y NHSM. Se usaron como entradas los conjuntos de previamente definidos y se evaluaron los resultados obtenidos. Asimismo, se obtuvieron recomendaciones utilizando las tres técnicas generadoras mencionadas en la sección anterior. Para las corridas, se solicitaron diferentes cantidades de vecinos cercanos y se consideró como calificación mínima para recomendación la mediana del rango de *ratings*.

Realizando el análisis de las medidas de evaluación con respecto a los vecinos cercanos considerados, la Fig. 4 muestra que la precisión de cada algoritmo es dependiente del número de éstos últimos y que en los métodos de Coseno y NHMS, la participación del incremento de los mencionados en el valor final de la métrica es marginal. La Fig. 5 muestra que el *coverage* de los sistemas es proporcional al número de vecinos, lo cual es un comportamiento esperado. Sin embargo, la Fig. 6 muestra que el incremento hace que el error del sistema aumente. Esto significa que las calificaciones predichas son más inexactas.

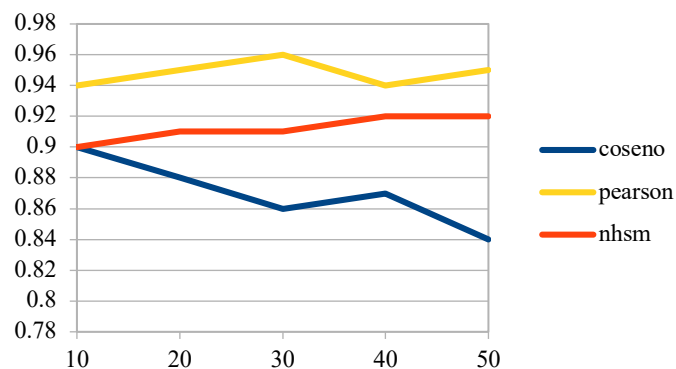


Figura 4. Comparativa Precisión / Número de vecinos para datos poco dispersos.

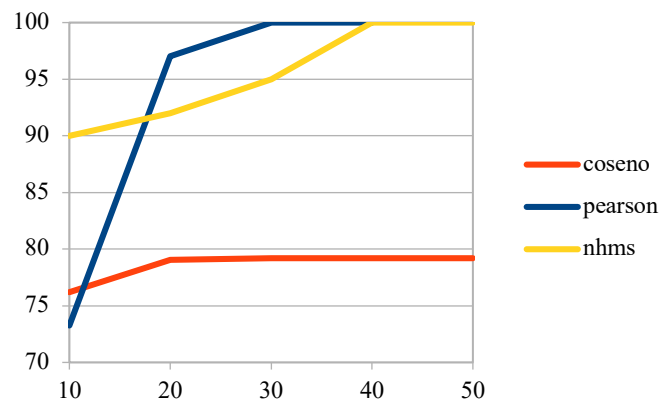


Figura 5. Comparativa Coverage / Número de vecinos para datos poco dispersos.

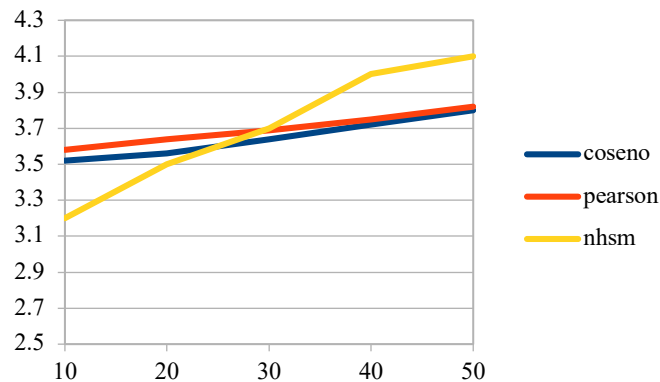


Figura 6. Comparativa MAE / Número de vecinos para datos poco dispersos.

Para el caso de los datos dispersos, los algoritmos no pueden en la mayoría de los casos terminar con el proceso de obtención de similitudes. Esto sucede debido a que existen ítems que no han sido valorados por ningún usuario o bien, casos de estos últimos que no han valorado ningún otro elemento en común con otros. Demostrando que el problema del inicio en frío es una condición crítica para cualquiera de los algoritmos de similitud [24]. En este caso es importante hacer notar que esto debe ser subsanado con alguna otra técnica para asegurarse que el elemento sea recomendado para empezar a obtener ponderaciones [25]. Netflix, por ejemplo, utiliza una lista de novedades para presentarla a los usuarios sin importar sus preferencias, de esta manera se asegura que haya algunos que comiencen a calificarlas.

5. Conclusiones

La validez de las recomendaciones obtenidas a partir de un SR basado en FC depende de dos aspectos importantes: el dominio de los contenidos del SR y la función de similitud utilizada para generar los vecinos cercanos. Para el primer punto es necesario identificar a priori si las características de los ítems serán tomadas en cuenta o no. Si las características no interesan y solo se recomendarán objetos similares en calificación, el enfoque colaborativo basado en ítems será una buena propuesta de solución si se tiene la matriz de ratings correspondiente. Si se desea que las características de los elementos sean tomadas en cuenta, entonces debe considerarse un SR basado en contenido en vez del FC.

Para el segundo punto, la medida de similitud dependerá de la naturaleza de la matriz de valoraciones. Si esta es dispersa entonces procesos como Pearson o coseno darán resultados poco precisos pues puede ser que no pueda

establecerse similitud entre los ítems/usuarios. NHSM puede ofrecer resultados más exactos en este caso, sin embargo, es un proceso lento debido a todos los cálculos previos que tiene que realizar antes de poder obtener la similitud. Si es densa o poco dispersa, entonces Coseno o Pearson podrán dar resultados mejores en menos tiempo.

Otro aspecto a considerar son los *ratings* que pueden utilizarse: NHSM es efectivo cuando las calificaciones incluyen números negativos, lo mismo que Pearson ponderado o PIP. Si solo incluyen números positivos entonces Coseno o su versión ajustada son mejores opciones.

Al evaluar la precisión debe considerarse que el número de vecinos a seleccionar es directamente proporcional al *coverage* del sistema. Sin embargo, también es directamente proporcional al error del sistema e inversamente proporcional a la precisión. Por lo que seleccionar el número correcto de vecinos debe ser una de las primeras decisiones.

6. Referencias

- [1] Dietmar, J., Zanker, M., Felfernig, A., Friedrich, G. (2011). *Recommender Systems*. New York: Cambridge University Press.
- [2] Facebook. (2019). <https://www.facebook.com>
- [3] Twitter. (2019). <https://twitter.com>
- [4] Pinterest. (2019). <https://www.pinterest.com.mx/>
- [5] Amazon. (2019). <https://www.amazon.com.mx>
- [6] Ebay. (2019). <https://www.ebay.es>
- [7] Netflix. (2019). <https://www.netflix.com/browse>
- [8] YouTube. (2019). <https://www.youtube.com/>
- [9] Moya García, R. (2013). *SVD aplicado a sistemas de recomendación basados en filtrado colaborativo* (Tesis Master). Universidad Politécnica De Madrid, España.
- [10] Sarwar, B., Karypis, G., Konstan, J., Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. Presentado en el *Tenth International Conference on World Wide Web - WWW*, Hong Kong. doi: <https://doi.org/10.1145/371920.372071>
- [11] Gunawardana, A., Shani, G. (2009). A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research*, 10, 2935-2962. Recuperado de: <https://dl.acm.org/citation.cfm?id=1755883>
- [12] Owen S., Robin A. (2009). *Mahout in Action*. Greenwich: Manning Publications Publications Co.
- [13] Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56, 156–166. doi: <https://doi.org/10.1016/j.knosys.2013.11.006>
- [14] Castellano, E. J., Martínez, L., Barranco, M., Pérez Cordon, L. G. (2007). Recomendación de perfiles académicos mediante algoritmos colaborativos basados en el expediente. Presentado en la *Conferência IADIS Ibero-Americana WWW/Internet*, Villa Real, Portugal. Recuperado de: https://sinbad2.ujaen.es/sites/default/files/publications/Castellano2007_IADIS.pdf
- [15] Rojas Castellanos, Y. (2014). Sistema de recomendación por filtrado colaborativo para el sistema de publicación de contenido multimedia—VideoWeb 1.0. *International Journal of Innovation and Applied Studies*, 6 (3), 326-334. Recuperado de: <http://www.ijias.issr-journals.org/abstract.php?article=IJIAS-14-098-04>
- [16] Del Pino, J., Salazar, G., Cedeño, V. (2011). Adaptación de un Recomendador de Filtro Colaborativo Basado en el Usuario para la Creación de un Recomendador de Materias de Pregrado Basado en el Historial Académico de los Estudiantes. *Revista Tecnológica - ESPOL*, 24 (2), 29-34. Recuperado de: <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/85>
- [17] Zeng, W., Zhu, Y.-X., Lü, L., Zhou, T. (2011). Negative ratings play a positive role in information filtering. *Physica A: Statistical Mechanics and its Applications*, 390 (23-24), 4486–4493. doi: <https://doi.org/10.1016/j.physa.2011.07.005>

- [18] Taghavi, M., Bentahar, J., Bakhtiyari, K., Hanachi, C. (2018). New Insights Towards Developing Recommender Systems. *The Computer Journal*, 61 (3), 319–348, doi: <https://doi.org/10.1093/comjnl/bxx056>
- [19] Papachristoudis, G. (2019). *Popular evaluation metrics in recommender systems explained*. Recuperado de: <https://medium.com/qloo/popular-evaluation-metrics-in-recommender-systems-explained-324ff2fb427d>
- [20] Pinto da Costa, J. F. (2011) Weighted Correlation. En: M. Lovric (eds), *International Encyclopedia of Statistical Science*. Springer-Verlag Berlin Heidelberg. doi: https://doi.org/10.1007/978-3-642-04898-2_612
- [21] Ramírez Morales, C. A. (2018). Algoritmo SVD aplicado a los sistemas de recomendación en el comercio. *Tecnología Investigación y Academia*, 6 (1), 18-27. Recuperado de <https://revistas.udistrital.edu.co/index.php/tia/article/view/11827>
- [22] Sadegui Moghaddam, M. H., Mustapha, N., Mustapha, A., Sharef, N. S., Elahian, A. (2015). Rust metrics in recommender systems: a survey. *Advanced Computational Intelligence: An International Journal*, 2 (3), 1-12. doi: <https://doi.org/10.5121/acii.2015.2301>
- [23] Goldberg, K., Roeder, T., Gupta, D., Perkins, C. (2000). Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval*, 4 (2), 133-151. doi: <https://doi.org/10.1023/A:1011419012209>
- [24] Graham, D. (2019). *Limitations of Collaborative Recommender Systems*. Recuperado de: <https://towardsdatascience.com/limitations-of-collaborative-recommender-systems-9801036941b3>
- [25] Papagelis, M., Plexousakis, D., Kutsuras, T. (2005). Alleviating the sparsity problem of collaborative. En: Herrmann P., Issarny V., Shiu S. (eds), *Trust Management. iTrust 2005. Lecture Notes in Computer Science, vol 3477*. Springer, Berlin, Heidelberg. doi: https://doi.org/10.1007/11429760_16