Título. Análisis comparativo de algoritmos para recomendar documentos basados en filtrado colaborativo.

Title. Comparative analysis of algorithms for recommending documents based collaborative filtering.

Categoría. Desarrollo humano/ Inteligencia artificial, máquinas inteligentes y robótica

Autores. Viana De la Cruz Leyva[1], Rolando Bairon Santisteban García[2], Yamila Silva del Rosario[3]

1 Universidad de Granma, vleyva@udg.co.cu
2 Delegación Provincial de la Agricultura en Granma, bairon86@nauta.com
3 Universidad de las Tunas

RESUMEN

La investigación se desarrolla con el objetivo de comparar algoritmos para la recomendación de documentos a partir de las opiniones de los usuarios. Se estudian los principales algoritmos existentes, así como de las métricas utilizadas para evaluar la precisión de las recomendaciones formuladas por estos. Se propone como solución al problema, la implementación de los algoritmos basados en memoria. Durante la evaluación se usaron tres conjuntos de datos y el marco de trabajo Crab.

Palabras clave: algoritmos, filtrado colaborativo, marco de trabajo, métricas, sistemas de recomendación.

SUMMARY

The research is developed with the aim of comparing algorithms for the recommendation of documents based on the opinions of the users. The main existing algorithms are studied, as well as the metrics used to evaluate the accuracy of the recommendations made by them. It is proposed as a solution to the problem, the implementation of memory based algorithms. Three data sets and the Crab framework were used during the evaluation.

Keywords: algorithms, collaborative filtering, framework, metrics, recommendation systems.

INTRODUCCIÓN

En un contexto informático una recomendación intenta proporcionar sugerencias significativas para los recursos que los usuarios puedan encontrar interesantes y útiles. Estas sugerencias se fundamentan en las calificaciones que un usuario le da a los recursos a través de una votación o mediante el historial de navegación. Si un usuario no ha calificado ningún recurso o no cuenta con navegación alguna, el sistema no le podrá brindar una recomendación con la mayor calidad y precisión posible [1].

Hoy se dispone de grandes cantidades de información, por lo que se hace necesario saber identificar cuál es la que resuelve cada una de las necesidades de las personas. Además, no se dispone del tiempo suficiente para seleccionar lo más interesante, tornando la revisión bibliográfica en un proceso tedioso.

CARACTERIZACIÓN DE LOS SISTEMAS DE RECOMENDACIÓN DE FILTRADO COLABORATIVO

- Sistemas de recomendación de filtrado colaborativo

Los SR de filtrado colaborativo son aquellos en los que las recomendaciones se realizan basándose solamente en los términos de similitud entre los usuarios, o sea, recomiendan objetos que son del gusto de otros usuarios de intereses afines. Los objetos a recomendar se elegirán entre aquellos que han recibido mayor puntuación por parte de otros usuarios con similares gustos o intereses, es decir, un patrón de preferencias similar [2].

Su funcionamiento se basa en recopilar las votaciones de cada individuo para una serie de elementos en un dominio dado, para luego nivelar personas que comparten las mismas necesidades o gustos. El procedimiento general aplicado en estos sistemas se puede resumir de la siguiente manera [3]:

1. Los usuarios expresan sus valoraciones sobre elementos del sistema mediante una escala numérica.
2. Luego se intenta predecir el puntaje que daría el usuario que solicita la recomendación, a los elementos del sistema no conocidos hasta el momento por él.
3. Las predicciones calculadas seleccionan los elementos con valores más altos para realizar la recomendación.

- Selección de información para la creación de los perfiles

La recomendación en los sistemas basados en filtrado colaborativo, se hace a partir de las preferencias y valoraciones que otorgan los usuarios hacia los diferentes objetos. Primeramente, se obtiene la información para la creación de los perfiles de usuarios, que indican las preferencias y luego, con la información coleccionada, se conforman los perfiles que utilizará el sistema para clasificar los usuarios y generar las recomendaciones.

Existen dos mecanismos para conseguir la información correspondiente a los gustos de los usuarios. De forma explícita, donde el usuario ofrece conscientemente su opinión sobre un documento, o de manera implícita, en las que esa opinión se extrae a partir de la interacción y el comportamiento del usuario en el sistema. En un SR con preferencias explícitas, el usuario se encarga de valorar los distintos documentos según su opinión. [4].

Por otra parte, en un SR con preferencias implícitas, la opinión del usuario se deduce a partir del uso que hace de la aplicación. En la práctica se han usado una gran variedad de eventos para la extracción

de información, como el tiempo que pasa leyendo un documento, los enlaces que visita y el número de veces que revisa un documento [4].

- Aplicación de los sistemas de recomendación de filtrado colaborativo

En la actualidad los SR basados en filtrado colaborativo han evolucionado y es posible encontrarlos en diversos ámbitos de aplicación como en el comercio electrónico, donde se han convertido en una herramienta fundamental para los proveedores en línea [5]. En cada dominio se presentan diferentes problemas a los que hay que dar soluciones diferentes.

- Algoritmos de recomendación de documentos basados en filtrado colaborativo

En los sistemas basados en filtrado colaborativo, el perfil de usuario es el conjunto de calificaciones otorgadas a los diferentes objetos. En general estas calificaciones, se representan como un valor unario (mostrando solo los elementos pertinentes), binario (que permite distinguir entre buenos y malos elementos) o comúnmente, como un valor numérico en una escala limitada.

Las valoraciones de los usuarios se almacenan en una tabla conocida como la matriz de calificación. Esta tabla se procesa con el fin de generar las recomendaciones. En dependencia de cómo son procesados los datos de la matriz de calificación, se clasifican los algoritmos de filtrado colaborativo. Existen dos tipos de algoritmos: basados en memoria y basados en modelos.

Los algoritmos basados en memoria utilizan medidas de similitud para seleccionar usuarios (u objetos) que son similares a la del usuario activo. La predicción se calcula a partir de las calificaciones de estos usuarios, por eso también se les llama basados en vecinos. Los algoritmos basados en vecinos, son de los más populares en la implementación de SR de filtrado colaborativo. Como su nombre lo indica se centran en buscar los vecinos de usuarios y objetos, es decir, otros usuarios y objetos con preferencias similares para generar las recomendaciones (ver figura 1). [6]

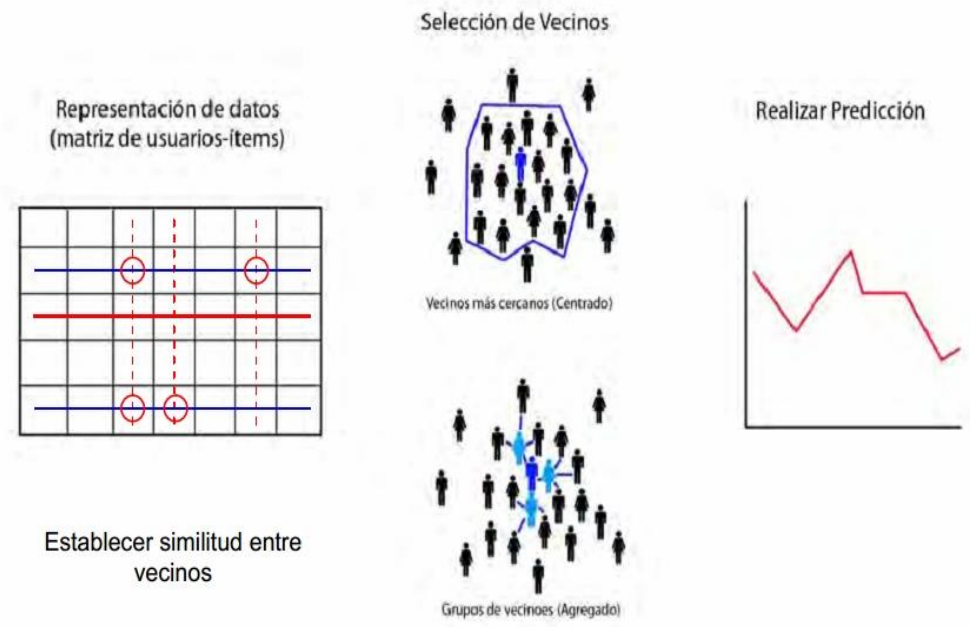

Figura 1: Funcionamiento de los algoritmos basados en memoria. Fuente: [7]

La mayoría de estos algoritmos pueden ser clasificados como los algoritmos basados en usuarios o algoritmos basados en objetos dependiendo de si el proceso de obtención de los vecinos se centra en la búsqueda de usuarios similares o en los objetos.

Los algoritmos basados en modelos o elementos, primero construyen un modelo para representar el comportamiento de los usuarios y por lo tanto, para predecir sus calificaciones. Los parámetros del modelo se estiman en línea con los datos de la matriz de calificación. Existen diferentes enfoques más relacionados con la máquina de aprendizaje en función de los métodos de álgebra lineal; el análisis de factores, la agrupación, las redes neuronales y los gráficos o métodos probabilísticos, como las redes de Bayes o modelos de clase latente [6].

En general, los algoritmos basados en memoria o basados en usuarios son más simples y se obtienen resultados razonablemente precisos. Sin embargo, presentan problemas de escalabilidad que hacen que el algoritmo tenga que procesar todos los datos para calcular una predicción, con un gran número de usuarios o elementos.

- Algoritmos basados en modelos

Los algoritmos de clasificación de la puntuación se orientan a la tarea de predicción. La abordan como un problema de clasificación, donde cada una de las posibles puntuaciones es vista como una clase y se intenta predecir a cuál de las posibles "clases" pertenece realmente la puntuación que el usuario actual habría dado a cierto objeto. A partir de un determinado modelo, el algoritmo estima la probabilidad de pertenencia a cada clase y la puntuación más probable es devuelta como predicción. Cada algoritmo utilizará un modelo diferente, el cual se entrena previamente a partir de la información en la matriz de puntuaciones.

Los algoritmos de agrupamiento o clustering consisten en clasificar usuarios y/u objetos en grupos, de acuerdo a las puntuaciones que les fueron otorgadas. Asumen que hay grupos de usuarios u objetos con similares características y por tanto, una vez que un usuario fue asignado a un grupo la predicción o recomendación se obtiene a partir de las puntuaciones concedidas por miembros del mismo grupo. Entre las técnicas de agrupamiento existentes una de la más sencilla es K-Means, en la que se utiliza un proceso iterativo para minimizar la distancia de los usuarios con el perfil medio del grupo al que pertenecen. En esta técnica generalmente, solo se agrupan los usuarios y cada usuario pertenece a un solo grupo [7].

- Algoritmos basados en memoria

Los algoritmos basados en usuario seleccionan el vecindario a partir de usuarios similares al usuario actual. La similitud se calcula comparando sus perfiles, es decir, sus puntuaciones. La idea es comparar las puntuaciones de aquellos objetos que ambos usuarios han puntuado. Si dos usuarios tienen un perfil de las puntuaciones similar en los objetos, se asume que el resto de las puntuaciones también van a ser parecidas y por tanto, se puede recomendar a un usuario otros objetos bien puntuados por sus vecinos. Los algoritmos basados en usuarios, también conocidos como basados en el vecindario, son una de las estrategias más populares de filtrado colaborativo. Siguen un proceso de tres pasos:

1. Calcular la similitud entre el usuario activo y el resto de los usuarios.
2. Seleccionar un subconjunto de usuarios de acuerdo a su similitud con el usuario activo.
3. Calcular la predicción utilizando las votaciones de los vecinos.

Desde las primeras propuestas de los algoritmos basados en usuarios, se han estudiado diferentes técnicas para hacer frente a cada uno de estos pasos. Por lo tanto, el enfoque basado en el usuario, se considera una familia de algoritmos en lugar de un único algoritmo. Cada uno combina diferentes estrategias para cada paso.

- Algoritmos basados en objetos

Los algoritmos basados en objetos son similares a la aproximación basada en usuarios descrita anteriormente, pero en lugar de buscar usuarios análogos al actual, se basan en examinar objetos similares a los que el usuario activo ha puntuado. Los pasos que realizan son los siguientes:

1. Calcular la similitud entre los objetos puntuados por el usuario activo y el resto de los objetos existentes.
2. Seleccionar un subconjunto de objetos de acuerdo a la similitud con los calificados por el usuario activo.
3. Calcular la predicción utilizando las votaciones a los objetos semejantes.

TENDENCIAS ACTUALES

El algoritmo basado en tendencias funciona de forma diferente a los algoritmos de filtrado colaborativo tradicionales. En lugar de buscar las relaciones entre individuos o elementos, se basa en las diferencias entre ellos. Es decir, en las variaciones de cada usuario a la hora de puntuar un determinado elemento. Las variaciones no solo se deben a las diferencias entre los gustos u opiniones entre usuarios, sino a otros aspectos como la apreciación o manera de puntuar de cada individuo. Existen usuarios que acostumbran evaluar positivamente, utilizando puntuaciones negativas para elementos realmente malos. Sin embargo, otros reservan las puntuaciones más altas para los mejores productos [10].

La tendencia se refiere a si un usuario es propenso a puntuar positivamente los pro-ductos o a hacerlo negativamente. Se define la tendencia de un usuario, como la diferencia media de sus puntuaciones respecto a la media de los elementos [10].

Los algoritmos k-NN son una de las técnicas de recomendación basadas en filtrado colaborativo más empleadas en la actualidad, debido a la calidad elevada de las recomendaciones y que exhiben buenos resultados en multitud de dominios y condiciones.

- Marcos de trabajo para evaluar algoritmos basados en filtrado colaborativo

Existen varias métricas para comparar algoritmos, más a la hora de aplicarlas el proceso de comparación puede tornarse lento y bastante tedioso, es por ello que surge la necesidad de utilizar frameworks. A continuación se mencionan algunos:

Apache Mahout: Proyecto de Apache Lucene cuyo propósito es construir librerías escalables de máquinas de aprendizaje y minería de datos. Implementa algoritmos basados en filtrado colaborativo y basados en contenido, utilizados para desarrollar SR. Permite evaluar algoritmos de recomendación, a través de diversas métricas que permiten conocer algunos aspectos en las recomendaciones como la precisión, la cobertura, la exhaustividad y la ganancia acumulada con reducción normalizada [8].

MyMediaLite: Biblioteca rápida que posee disímiles funciones para la creación de SR basados en filtrado colaborativo y realiza evaluaciones a nuevos algoritmos desarrollados. Entre los aspectos que permite evaluar se encuentran los errores que comenten los algoritmos al predecir la puntuación hacia los objetos, la ganancia acumulativa con descuento normalizado y la precisión promedio de decisión [9]. Basa su funcionamiento en dos tareas principales: la predicción de las votaciones y la generación de la lista con las recomendaciones.

Crab: Herramienta de código abierto distribuida bajo la licencia BSD. Framework escrito en Python, publicado en el año 2011, para brindar la posibilidad de evaluar los algoritmos de recomendación. Posee un potente módulo con las métricas más utilizadas para analizar los resultados de

un algoritmo de recomendación [10]. Actualmente permite evaluar únicamente los algoritmos basados en usuarios y los basados en objetos, brinda la posibilidad de medir varios aspectos como: la precisión, retentiva, así como conocer la medida del error que comete el algoritmo al realizar cada una de las predicciones. Además, permite visualizar los resultados de las evaluaciones mediante gráficas y adaptarlos a diferentes dominios de información.

COMPARACIÓN DE LOS ALGORITMOS

Para obtener un algoritmo de calidad es necesario escoger de forma adecuada las tecnologías a utilizar para el desarrollo de una solución robusta y lo más eficiente posible.

- Marco de trabajo y lenguaje de programación

Se escogió como marco de trabajo para efectuar la evaluación de los algoritmos de recomendación Crab en su versión 0.1, pues es una herramienta de código abierto y de libre utilización. Además, permite evaluar diferentes aspectos en los algoritmos de recomendación como son: la exhaustividad, la precisión de las predicciones y de las recomendaciones. Otra razón por la cual fue seleccionado Crab para probar los algoritmos es la sencillez de su uso, lo que evita emplear tiempo en el estudio de otra herramienta con una curva elevada de aprendizaje. Teniendo en cuenta que es un framework desarrollado en Python, este es el lenguaje de programación empleado en la implementación de los algoritmos a comparar. Y como lenguaje se selecciona Python porque brinda legibilidad y elegancia en la sintaxis, por lo que es imposible escribir un código ofuscado. Además, existe un número elevado de librerías escritas para este lenguaje que servirán de apoyo para el desarrollo de la solución.

ALGORITMO SELECCIONADO

- Algoritmo k-NN

El algoritmo k-NN (k vecinos más cercanos) es un tipo de clasificador basado en instancias. Estos clasificadores, trabajan directamente sobre los datos sin construir ningún tipo de modelo sobre la base de datos, están basados en aprendizaje por analogía encuadrándose dentro del paradigma del aprendizaje perezoso [11]. A pesar de que los algoritmos k-NN son una de las técnicas de recomendación más antiguas, siguen siendo hoy en día una de las más populares. De hecho, este tipo de algoritmos todavía presentan importantes ventajas sobre las técnicas desarrolladas en los últimos años, mucho más complejas y elaboradas [6]:

Simplicidad: convierte a estos algoritmos en una técnica fácil de entender y por tanto de implementar.

Justificación de resultados: el funcionamiento de estos algoritmos es muy intuitivo y permite derivar fácilmente el motivo del porqué un objeto ha recomendado, lo que es muy útil de cara a justificar al usuario las razones de una recomendación concreta.

Eficiencia: a pesar de que en sistemas comerciales se añaden continuamente nuevos objetos y usuarios, estos algoritmos no necesitan volver a entrenar complejos modelos, sino simplemente actualizar ciertas estructuras como el vecindario, lo que repercute también en una mayor eficiencia.

Tradicionalmente, los algoritmos k-NN, al igual que la mayor parte de algoritmos de filtrado colaborativo, se habían desarrollado con la tarea de predicción en mente.

Su uso en la tarea de recomendación, requiere de pequeños cambios que permitan aprovechar sus numerosas ventajas. A continuación en la figura 2 se muestra el funcionamiento de los algoritmos k-

NN [6]. El funcionamiento general de un algoritmo k-NN en la tarea de recomendación es bastante sencillo.

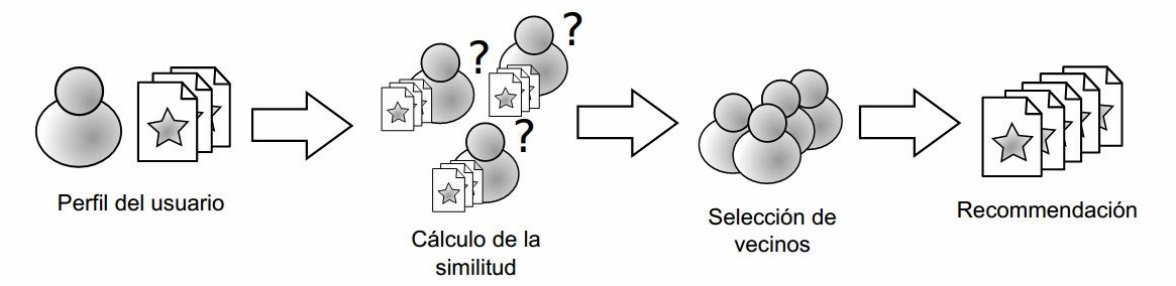

Figura 2: Funcionamiento de los algoritmos k-NN en la tarea de recomendación. Fuente: [7].

- Experimentos y conjuntos de datos

Existen diferentes tipos de experimentos que se pueden realizar con los algoritmos de recomendación. De manera general se clasifican en experimentos offline, en los cuales no intervienen usuarios que soliciten información al sistema en tiempo real, sino que se emplean conjuntos de datos previamente almacenados, y los experimentos online, donde sí se cuenta con usuarios que interactúan con el sistema [7].

Se utilizó un experimento offline, ya que no se cuenta con un sistema donde los usuarios puedan efectuar sus votaciones con respecto a productos reales. La principal ventaja es la comodidad y el coste, pues permiten comparar un gran número de técnicas y opciones sin salir del laboratorio. Además, pueden repetirse en cualquier momento, lo que simplifica la comparación entre distintos algoritmos. La evaluación parte de un conjunto de datos previamente almacenados, que recoge información acerca de las preferencias u opiniones de un cierto número de usuarios sobre un conjunto de productos. Generalmente, se utilizan conjuntos de datos seleccionados a partir de sistemas reales, que son públicos y por tanto pueden ser usados por distintos grupos de investigación, lo que facilita la comparación de resultados.

En el ámbito de los sistemas de recomendación y especialmente la técnica de filtrado colaborativo, son muy populares tres conjuntos de datos pertenecientes a sistemas de recomendación de películas, música y libros respectivamente.

El principal problema de la evaluación offline es que solo permite valorar ciertos aspectos de las técnicas de recomendación. Además, la información disponible sobre cada usuario es limitada, lo que lleva a asumir que el comportamiento del usuario es el reflejado por los datos que se disponen, sin posibilidad de considerar la influencia del sistema de recomendación o la calidad de productos que el usuario no había valorado originalmente.

- Descripción de los conjuntos de datos

MovieLens es un conjunto de datos coleccionado por GroupLens Research Project de la Universidad de Minnesota. La información que contiene son datos reales correspondientes a las calificaciones de películas, recopiladas a través del sitio web MovieLens durante un periodo de 7 meses. De manera general contiene 100 000 calificaciones de 943 usuarios, realizadas a 1682 archivos de películas. Los

creadores del conjunto eliminaron los usuarios con menos de 20 calificaciones, con el fin de resaltar las películas suficientemente evaluadas y con ello reducir el tamaño del conjunto [12].

Book-Crossing es un grupo de información acerca de libros, recopilada por Cai-Nicolas Ziegler. Los datos que recoge se obtuvieron de la Comunidad Book-Crossing en un plazo de cuatro se-manas. En total almacena 1149780 votaciones en un rango de 1-10 hechas por 278858 usuarios a 271379 libros.

Sample Songs es un conjunto de datos coleccionados para programadores de algoritmos de recomendación. La información que almacena está relacionada con las votaciones de usuarios a varios archivos de música. Consiste en un fichero llamado "sample_songs.csv" que contiene 50 votaciones hechas por 8 usuarios a 8 archivos de música.

EVALUACIÓN DE LOS ALGORITMOS

La evaluación de los algoritmos de recomendación es una tarea compleja, pues la calidad de una recomendación depende de un gran número de factores, que incluye la persona que la recibe, el dominio y el objetivo del sistema, al contrario que otro tipo de algoritmo donde está relativamente claro que parámetros se deben mejorar.

Los resultados que a continuación se exponen fueron alcanzados luego de aplicar las métricas seleccionadas para evaluar los algoritmos. Cada una de las tablas recoge los valores arrojados por el marco de trabajo empleado para la evaluación. Es válido resaltar que el proceso de pruebas realizado mediante Crab se torna un poco complejo y el consumo de recursos del ordenador es bastante elevado. Las características del computador usado en el transcurso de la evaluación son: procesador Intel Core i5-3470 con 3.20 GHz de velocidad en el procesamiento de datos, memoria instalada de 4GB y el sistema operativo de 64 bit.

Las tablas siguientes recopilan los resultados de la evaluación al utilizar los conjuntos de datos Sample Songs, MovieLens y Book-Crossing respectivamente:

Métrica	Algoritmo basado en usuarios	Algoritmo basado en objetos
RMSE	1.27	1.14
MAE	1.13	0.87
precisión	0.72	0.78
exhaustividad	0.72	0.78
promedio	0.72	0.78

Métrica	Algoritmo basado en usuarios	Algoritmo basado en objetos
RMSE	1.12	1.00
MAE	1.5	0.74
precisión	0.74	0.80
exhaustividad	0.74	0.80
promedio	0.74	0.80

Métrica	Algoritmo basado en usuarios	Algoritmo basado en objetos
RMSE	1.04	0.96
MAE	0.99	0.92
precisión	0.80	0.83
exhaustividad	0.79	0.81
promedio	0.79	0.82

Los dos algoritmos mejoran sus resultados a medida que aumenta el porcentaje de puntuaciones en el conjunto de datos. Este resultado es el esperado y prueba que el error cometido por un algoritmo depende no tanto de la técnica empleada sino de la cantidad de información que dispone. En condiciones de densidad relativamente elevadas ambas soluciones presentan resultados similares. Se percibe cómo el algoritmo basado en objetos es más factible que el basado en usuarios y posee mayor precisión en las recomendaciones que efectúa, donde los elementos relevantes generalmente, están presentes en la lista. Además, el algoritmo basado en objetos tiende a cometer menos errores al predecir las puntuaciones de los usuarios con respecto a objetos no calificados

El algoritmo normaliza los datos correspondientes a los usuarios, objetos y puntuaciones. Organiza la información de forma tal que solo necesita los identificadores de usuarios y objetos, crea una colección constituida por cada usuario y el listado de los objetos de su preferencia con la respectiva calificación. Esta característica permite adaptar el algoritmo a diferentes dominios de información.

Para el correcto funcionamiento del algoritmo es imprescindible surtirle los datos de los usuarios, objetos y puntuaciones indispensables para la elaboración de las recomendaciones. Luego se debe seleccionar la medida que se empleará para el cálculo de la similitud entre los objetos. Por último, especificar la cantidad de elementos que conformarán la lista de la recomendación.

REFERENCIAS BIBLIOGRÁFICAS

[1] joseph a konstan, LGT, john TR, and jonathan LH. “Evaluating Collaborative Filtering Recommender Systems”. {ACM} Trans. Inf. Syst.2004.

[2] miguel SF and carles F. “Upcommons”. 2004.

[3] paolo M and paolo A. “Trust-aware Collaborative Filtering for Recommender Systems”. 2004.

[4] zan H, daniel Z, and hinchun C. “A comparison of collaborative-filtering recommendation algorithms for e-commerce”. 2007.

[5] riedl SK. “Sistemas de Recomendación”.

[6] jonathan LH, joseph AK, and john TR. “An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms”. 2004.

[7] marlin B. “Collaborative filtering: A machine learning perspective”. 2004.

[8] sebastian S. “Collaborative Filtering with Apache Mahout”. 2013.

[9] rendle SG, Zeno and christoph F. “MyMediaLite: A Free Recommender System Library”. 2011.

[10] marcel, M BC and caspirro R. “Crab: A Recommendation Framework for Python”. 2011.

[11] concha B and pedro L. “Vecinos más cercanos”. 2011.

[12] GroupLens Research Project. “MovieLens Dataset”. 2013.