MINIMIZACIÓN DEL RANGO DE UNA MATRIZ APLICADO EN LA OPTIMIZACIÓN DE UN SISTEMA DE RECOMENDACIÓN DE PRODUCTOS

TATIANA CAROLINA GÉLVEZ BARRERA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2016

MINIMIZACIÓN DEL RANGO DE UNA MATRIZ APLICADO EN LA OPTIMIZACIÓN DE UN SISTEMA DE RECOMENDACIÓN DE PRODUCTOS

TATIANA CAROLINA GÉLVEZ BARRERA

Trabajo de grado para optar al título de Ingeniero Industrial e Ingeniero de Sistemas

Director:
HENRY ARGUELLO FUENTES
PhD en Ingeniería eléctrica y computación

Codirector:
HENRY LAMOS DÍAZ
PhD en matemática-física

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2016

DEDICATORIA

A Dios, por haberme guiado a lo largo de este camino, por hacerme una mujer fuerte, valiente y no permitirme desfallecer en los momentos que parecían difíciles.

A mis padres, por su amor y apoyo incondicional, por brindarme la oportunidad de estudiar y culminar satisfactoriamente mis metas. Cada logro alcanzado es sólo el reflejo de su grandeza como padres.

A mis hermanos mayores, por cuidar desde lejos o cerca de mí, por ser un valioso ejemplo y enseñarme a ser luchadora y perseguir mis sueños.

AGRADECIMIENTOS

Quisiera agradecer a todas las personas que con su compañía, amistad y colaboración aportaron su granito de arena para que este trabajo y mi sueño de ser Ingeniera culminara con éxito.

Especialmente, agradezco a Dios porque si él nada sería posible.

Agradezco a mi mami CARMENZA BARRERA, quien con su amor y sabiduría me ha acompañado en todo momento, en todo lugar sin duda alguna, y quien me ha formado como una mujer de principios, valiente y que lucha por sus sueños.

Agradezco a mi papi J.R. GÉLVEZ quien me ha apoyado en cada proyecto emprendido y quien me ha enseñado a disfrutar y ver la vida de una bonita manera, a sonreír ante las adversidades.

A mis hermanos HARLEY GÉLVEZ y MAYRA GÉLVEZ por ser un ejemplo y guía.

A mi tía ROSA DELIA BARREA por ser una segunda mamá, amiga y compañera incondicional.

A toda mi familia, por enseñarme el valor de compartir y disfrutar juntos. Todo pasa, la familia permanece.

A mi director HENRY ARGUELLO FUENTES, por ser un ejemplo e inspiración para alcanzar grandes sueños. A mi codirector HENRY LAMOS por la confianza y el apoyo brindado. A mi tutor HOOVER RUEDA, por su paciencia, dedicación y apoyo.

A la familia PRADA REMOLINA, por ser unos amigos incondicionales, especialmente a la señora ANA REMOLINA por ser un ángel en mi vida.

Al grupo HDSP, porque me han enseñado el poder de la unión como equipo.

A todos mis amigos y compañeros por cada sonrisa y experiencia compartida.

CONTENIDO

Pág.

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE ANEXOS

RESUMEN

TÍTULO: “Minimización del rango de una matriz aplicado en la optimización de un sistema de recomendación de productos”*.

AUTORES: Tatiana Carolina Gélvez Barrera**

PALABRAS CLAVE: Sistema de recomendación, reducción dimensional, optimización matemática, análisis de datos, teoría de completar matrices.

DESCRIPCIÓN: Un sistema de recomendación (SR) es una herramienta software de mercadeo focalizado que relaciona a un usuario con los productos de su mayor interés. Amazon.com, TripAdvisor y Netflix son algunos ejemplos. Un SR predice el nivel de preferencia de un usuario hacia un producto con un puntaje en una escala de valoración. Estos puntajes son calculados mediante algoritmos que procesan datos del comportamiento de compra y características de los usuarios. Por ejemplo, la reducción dimensional es un enfoque que calcula una estructura latente para caracterizar el comportamiento de un usuario. Particularmente, la teoría de completar matrices se basa en la minimización del rango de una matriz de interacción usuario-producto para determinar tal estructura. Esta metodología es una de las más precisas y exactas. Por lo anterior, en este proyecto de grado se implementó un algoritmo bajo la metodología de completar matrices para mejorar el rendimiento de un sistema de recomendación de restaurantes. El algoritmo propuesto denominado DeMBaR subdivide la matriz de interacción en pequeña sub-matrices para estimar los puntajes. Los experimentos realizados muestran un incremento porcentual de hasta 5 % en la métrica de precisión y de hasta 53% para la métrica de exhaustividad respecto al enfoque semántico.

---

* Trabajo de grado

** Facultad de ingenierías físico-mecánicas. Director: PhD. Henry Arguello Fuentes. Codirector: PhD: Henry Lamos Díaz.

ABSTRACT

TITLE: "Commercial recommender system optimization via low-rank minimization."*

AUTHORS: Tatiana Carolina Gélvez Barrera**

KEY WORDS: Recommender system, dimensional reduction, mathematical optimization, data analytics, matrix completion.

DESCRIPTION: A recommender system (RS) is a software tool used as a target marketing tool that aims to provide suggestions for items to be of interest to an user. Amazon.com, TripAdvisor and Netflix are some examples. The interest of the user to an item is calculated as a score in a rating scale. RS make data processing to analyze the consumer behavior in the past in order to predict the ratings of the consumer behavior in the future. For instance, the dimensional reduction approach calculates a latent structure that determines the user behavior. Particularly, the matrix completion theory (MC) is based in the rank minimization of the user-item matrix to determine the latent structure. MC methodology has obtained high levels of performance in real applications. For this reason, in this final project it was implemented an algorithm under the MC methodology in order to improve the performance of a restaurants' recommender system. The proposed algorithm called DeMBaR divides the user-item matrix in smaller sub-matrices to estimate the missing ratings. Several realized experiments show an improvement up to 5 % in precision and up to 53 % in the recall metric

* Trabajo de grado

** Facultad de ingenierías físico-mecánicas. Director: PhD. Henry Arguello Fuentes. Codirector: PhD: Henry Lamos Díaz.

INTRODUCCIÓN

El auge de internet y las tecnologías de información y comunicación (TIC) han fortalecido la comunicación entre los usuarios* y las organizaciones. No obstante, la cantidad de información disponible es superior a la que un usuario puede consultar y procesar. Esta sobrecarga de información[1] hace difícil tomar una decisión cuando se va a realizar una compra. Así mismo, implica que las organizaciones deban almacenar, procesar y analizar grandes volúmenes de datos[2].

Para facilitar la toma de decisiones, los usuarios confían en recomendaciones de otros usuarios con alguna experiencia similar. El área de investigación conocida como sistemas de recomendación (SR) es una extensión de este fenómeno de confianza el cual, tiene como propósito generar recomendaciones y proporcionar información relevante a los usuarios mediante el análisis y procesamiento de datos[3].

Un SR es una herramienta computacional que analiza datos digitales con el propósito de recomendar productos** a un conjunto de usuarios[4]. De esta forma, un SR cumple dos funciones principales, en primer lugar, facilitar a los usuarios la búsqueda del producto apropiado y en segundo lugar, incrementar el beneficio de las organizaciones por medio del mercadeo focalizado[5]. Algunas aplicaciones comerciales de este tipo son Netflix, Amazon, TripAdvisor y MovieLens[6].

(*) El término usuario en éste documento hace referencia a usuarios, beneficiarios o compradores.

[1] HILBERT, Martin y LÓPEZ Priscila. The world's technological capacity to store, communicate, and compute information. En: Science. Abril, 2011. vol. 332. no. 60, p. 60-65.

[2] SCHAFER, Ben, KONSTAN, Joseph y RIEDL, John. E-Commerce recommendation applications. En: Data mining and knowledge discovery. Enero-Abril, 2001. vol. 5, p.115-153.

[3] SINGHAL, Amit. Modern information retrieval: A brief overview. En: IEEE Data Eng. Bull. Diciembre, 2001. vol 24. no. 4, p.35-43.

** El término producto en éste documento hace referencia a cualquier bien, servicio o contenido.

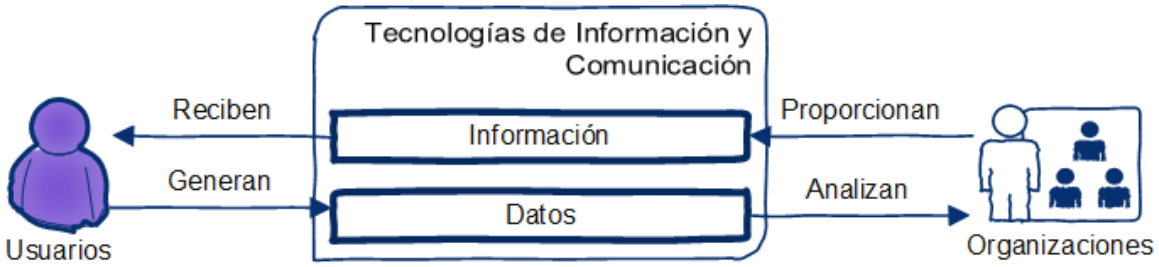
[4] RICCI, Francesco, ROKACH, Lior y BRACHA, Shapira. Introduction to recommender systems handbook. En: Recommender systems handbook. Springer. New York Dordrecht Heidelberg London, 2011, p. 1-35.

[5] MOCEAN, Loredana, y CIPRIAN Marcel. Marketing recommender systems: a new approach in digital economy. En: Informática Economíca. Octubre, 2012. vol. 16. no. 4, p.142-149.

[6] Ibíd. p. 4.

La Figura 1 muestra una vista conceptual de la interacción entre los usuarios y las organizaciones por medio de las TIC. Los datos generados por los usuarios son procesados y analizados por las organizaciones; como resultado, las organizaciones proporcionan información relevante que conduzca al usuario a tomar una decisión de compra acertada de manera eficiente.

Figura 1 Interacción Usuario-Organización



La efectividad de los SR depende principalmente del algoritmo para predecir el nivel de preferencia de un usuario por un producto[7]. Por lo tanto, mejorar la escalabilidad, precisión y rendimiento de los algoritmos es un problema de gran interés en el ámbito académico y comercial[8]. Los diferentes algoritmos se han clasificado en dos categorías principales: basados en contenido y de filtrado colaborativo (FC).

Los algoritmos basados en contenido realizan las predicciones analizando las características inherentes a los usuarios y/o a los productos. En contraste, los de FC se basan en un conjunto de puntajes explícitos que representan el grado de preferencia de los usuarios hacia los productos, estos puntajes se almacenan en una matriz, denominada la matriz de interacción usuario-producto (U-P)[9].

---

[7] GHOSHAL, Abhijeet, SUBODHA Kumar, y VIJAY Mookerjee. Impact of recommender system on competition between personalizing and non-personalizing firms. En: Journal of Management Information Systems. Enero, 2015. vol. 31. no. 4, p.243-277.

[8] ADOMAVICIUS, Gediminas y TUZHILIN, Alexander. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. En: IEEE transactions on knowledge and data engineering. Junio, 2005. vol.17, no. 6, p.734-749.

[9] Ibid. p.1

Algunas alternativas híbridas proponen explotar las características inherentes y los puntajes simultáneamente. Otros algoritmos incluyen información contextual, demográfica y realimentación mediante comentarios para mejorar el rendimiento[10]. Enfoques más recientes suponen que las decisiones de los usuarios están determinadas por un conjunto de factores latentes. Por lo tanto, se han empleado técnicas de reducción dimensional para abordar el problema de predicción[11].

En las técnicas de reducción dimensional se propone representar la matriz U-P mediante una matriz de bajo rango. El bajo rango indica que la matriz tiene solo unas filas o columnas independientes que determinan el comportamiento de toda la matriz. Ésta técnica ha obtenido niveles altos de precisión respecto a las técnicas tradicionales[12]. En consecuencia, en este proyecto de grado se modeló, implementó y evaluó un algoritmo de predicción bajo la metodología de reducción dimensional para generar recomendaciones y mejorar el rendimiento de un sistema de recomendación de restaurantes.

El algoritmo denominado DeMBaR explota la estructura de bajo rango de los datos analizados; sin embargo, a diferencia de la técnica tradicional, supone que la matriz U-P está compuesta por varias sub-matrices de bajo rango. De esta manera, los puntajes faltantes son estimados por medio de la técnica de completar matrices sobre un conjunto de sub-matrices de bajo rango que componen la matriz U-P original. Para evaluar el rendimiento del algoritmo DeMBaR se utilizó una base de datos (BD) de un SR de restaurantes. En los experimentos realizados se variaron los parámetros que podrían afectar el rendimiento del algoritmo, como el solucionador principal de minimización del rango de una matriz, la cantidad de sub-matrices estimadas y la forma de encontrar tales matrices. Los resultados obtenidos

[10] RICCI. Op. cit. p.14.
[11] KOREN, Yehuda, BELL, Robert y VOLINSKY, Chris. Matrix factorization techniques for recommender systems En: IEEE Computer Society. Agosto, 2009. vol. 8 no. 42, p. 30-37.
[12] Ibid. p.1

mostraron un incremento promedio en la precisión de hasta 5% y en la métrica de exhaustividad de hasta 53%. Adicionalmente comparado con la técnica anterior del SR, el sistema propuesto no necesita de información contextual para realizar las recomendaciones.

1. CUMPLIMIENTO DE OBJETIVOS

Objetivo general: Diseñar un algoritmo de optimización convexa para la minimización del rango de una matriz y aplicarlo en el mejoramiento de un sistema de recomendación de productos.

En la Tabla 1 se relacionan los objetivos específicos con el numeral que corresponde a su cumplimiento. De esta manera, se da por alcanzado el objetivo general planteado en el proyecto.

Tabla 1 Cumplimiento de objetivos

Objetivos Específicos	Cumplimiento
Realizar una revisión de literatura de la minimización del rango de una matriz y de los sistemas de recomendación.	Numeral 2.
Diseñar un algoritmo de optimización convexa para la minimización del rango de una matriz.	Numeral 6.
Implementar el algoritmo diseñado en un sistema de recomendación de productos dado.	Numerales 4,5.
Evaluar el rendimiento del sistema de recomendación, midiendo la capacidad de predicción del algoritmo implementado.	Numeral 7.
Elaborar un producto publicable en una revista indexada a partir de los resultados obtenidos en la investigación.	Anexo C

2. REVISIÓN DE LA LITERATURA

2.1.SISTEMAS DE RECOMENDACIÓN

En su revisión, SINGHAL[13] expone cómo la invención de las computadoras y el auge de la web han proporcionado a los usuarios una herramienta útil para almacenar grandes volúmenes de datos. Así mismo, analiza el problema que dicha alta capacidad de almacenamiento generó desde sus inicios: encontrar información de interés de manera rápida y efectiva. SINGHAL hace un recorrido en la evolución de la primera técnica de recuperación de información cuyo propósito es extraer información relevante de forma automática. No obstante, otras áreas de investigación como minería de datos, ciencia cognitiva, teoría de la aproximación, y recientemente los llamados sistemas de recomendación (SR) también han tratado el problema de extracción de información a partir de grandes volúmenes de datos.

Los SR surgen a mediados de los años 90 con la publicación de trabajos como el expuesto por GOLDBERG *et al.*[14], quienes desarrollaron Tapestry en 1994, el primer software considerado un SR. Los autores introdujeron el término filtrado colaborativo (FC) al estado del arte, este término hace referencia a la participación directa de las personas en el proceso de filtrado de información automática. El propósito de Tapestry era filtrar los correos recibidos electrónicamente, identificando aquellos relevantes para el usuario y aquellos que eran no deseados. Por medio de esta investigación los autores demostraron que incluir a las personas en el proceso de filtrado automático era una estrategia efectiva para extraer información relevante de manera eficiente.

---

[13] SINGHAL. Op. cit.

[14]GOLDBERG, David *et al.* Using collaborative filtering to weave an information Tapestry. En: Communications of the ACM. Diciembre, 1992. vol.35. no.12, p. 61-70.

En el mismo año, RESNICK *et al.*[15], desarrollaron GroupLens, una herramienta cuyo objetivo era encontrar los artículos más interesantes dentro de una enorme cantidad de artículos disponibles. Como aporte principal, estos autores incluyeron el término puntajes (del inglés *ratings)* al estado de arte. Un puntaje se define como la calificación impuesta directamente por un usuario a un producto que ha consumido en el pasado. Adicionalmente, los autores modelaron los datos como una estructura matricial la cual, relacionaba a cada usuario con cada producto mediante el puntaje dado. Es así como el problema de FC se establece formalmente como: estimar la información desconocida dentro de la matriz de puntajes a partir de los valores conocidos.

En este orden de ideas, RESINCK y VARIAN[16] realizan la primera comparación y evaluación de 5 herramientas de la época, en 1997. Los autores clasificaron la naturaleza del software, los datos de entrada y la presentación de las recomendaciones con el objetivo de proporcionar un marco de evaluación del rendimiento del sistema según su clasificación. La Tabla 2 resume la clasificación de las características relevantes de un SR considerada por tales autores. A partir de esta investigación, se establece que el factor más importante en la efectividad de un SR es el algoritmo para calcular las predicciones.

Tabla 2 Características relevantes en la evaluación de SR - 1997.

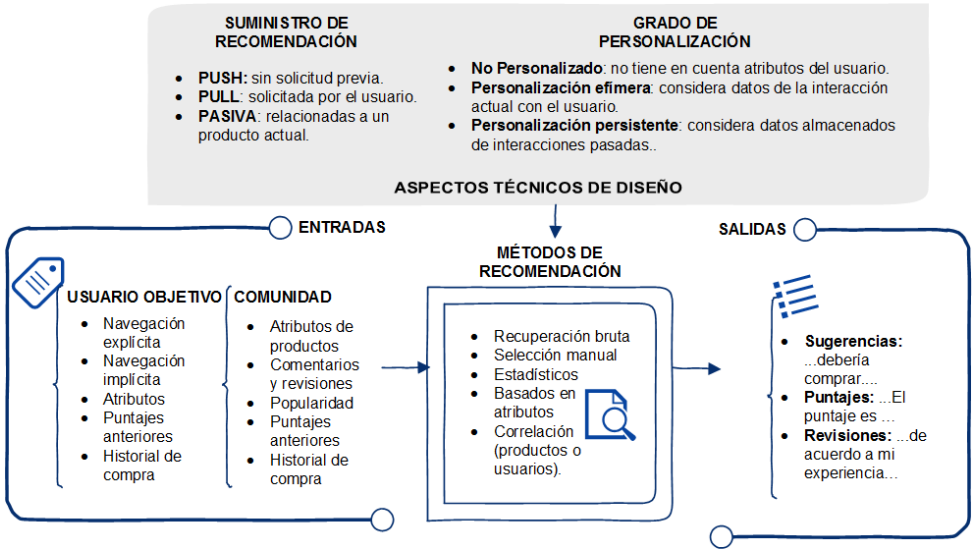
Software	Datos de entrada	Visualización
• Recomendaciones personalizadas en base al pasado. • Análisis del contenido.	• Implícitos: tiempo de lectura de una página web, referencias. • Explícitos: Puntajes dados dentro de una escala numérica.	• Filtrado por selección. • Listas organizados. • Consolidado de productos encontrados.

[15] RESNICK, Paul, *et al.* GroupLens: an open architecture for collaborative filtering of Netnews. En: Conference on Computer supported cooperative work (5: 22-26, octubre: Chapel Hill, North Carolina, USA). Proceedings of ACM conference on computer supported cooperative work. New York, NY, USA: ACM, 1994, p.175-186.
[16] RESINCK, Paul y VARIAN, Hal. Recommender systems. En: Communications of the ACM. Marzo, 1997. vol. 40. no. 3, p. 56-58.

En consecuencia, BREESE, HECKERMAN y KADIE[17] realizaron un estudio de los algoritmos desarrollados bajo el enfoque particular de FC en 1998. Allí, distinguieron dos categorías de algoritmos: basados en memoria y basados en modelos. En la primera categoría, la estimación se realiza con un promedio ponderado de los puntajes asignados por una vecindad de usuarios o productos. La segunda categoría es el enfoque probabilístico que estima los valores esperados para los puntajes desconocidos. Como conclusión se determina que dependiendo del objetivo del sistema, el rendimiento de los diferentes enfoques cambia.

Seguidamente, SCHAFER, KONSTAN y RIEDl[18] proponen una taxonomía propia de los SR, la cual estableció tres categorías generales: las entradas y salidas del sistema, los métodos de generación de recomendaciones y los aspectos técnicos de diseño. La Figura 2 muestra un esquema de la taxonomía propuesta.

Figura 2 Esquema de la taxonomía de los SR-2001.



[17] BREESE, John, HECKERMAN, David y KADIE, Carl. Empirical analysis of predictive algorithms for collaborative filtering. En: Conference on Uncertainty in Artificial Intelligence (UAI) (3: 24-26, julio: Madison, Wisconsin, USA). Proceedings of the 14th conference on uncertainty in artificial intelligence. San Francisco, CA, USA: UAI. 1998 p. 43-52.

[18] SCHAFER, Ben, KONSTAN, Joseph y RIEDL, John. E-Commerce recommendation applications. En: Data mining and knowledge discovery. Enero-Abril, 2001. vol. 5. no.1-2, p.115-153.

Años más tarde, HERLOCKER *et al*.[19] realizan una nueva revisión de las metodologías de evaluación de los SR en FC. Con su trabajo se establecen nuevos factores a los considerados por RESINCK y VARIAN[20] anteriormente, algunos de ellos son: la expectativa de los usuarios sobre el SR, la selección apropiada de las bases de datos a evaluar y analizar las fortalezas, debilidades e interpretación de las métricas de evaluación. Finalmente, plantean nuevas características de evaluación. Por ejemplo, ¿Qué tan sorprendente (del inglés *serendipity*) e inesperada es la recomendación ofrecida para el usuario?, ¿Cuál es la variabilidad de las recomendaciones? y ¿Qué tan satisfechos están los usuarios con los resultados ofrecidos?

En el año 2005 ADOMAVICIUS y TUZHILIN[21] presentan una revisión detallada del estado del arte de los SR. Los autores exponen dos importantes limitaciones: el problema de arranque en frio (del inglés *cold start*) y la escasez de datos (del inglés *sparsity*). En su investigación se describen algunos retos del área tales como: mejorar la representación de los usuarios y productos mediante modelos más realistas; incorporar información contextual para incrementar la precisión de los resultados, desarrollar herramientas eficientes que admitan puntajes basados en múltiples criterios y emplear estrategias menos intrusivas que generen confianza y aceptación de los usuarios hacia las recomendaciones dadas.

La generación de confianza es un aspecto que empezó a ser ampliamente estudiado por CRAMER, *et al.*[22] en el año 2008. Los autores analizaron el efecto que la transparencia en el proceso de predicción causaba sobre la aceptación,

---

[19]HERLOCKER, Jonathan *et al.* Evaluating collaborative filtering recommender systems En: ACM Transactions on information systems. Enero, 2004. vol. 22. no. 1, p. 5–53

[20] RESINCK y VARIAN. Op. cit.

[21]ADOMAVICIUS, Gediminas y TUZHILIN, Alexander. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. En: IEEE Transactions on knowledge and data engineering. Junio, 2005. vol. 17. no. 6, p.734-749.

[22] CRAMER, Henriette, *et al*. The effects of transparency on trust in and acceptance of a content-based art recommender. En: User modeling and user-adapted interaction. Noviembre, 2008. vol. 18, no. 5, p. 455-496.

confianza y satisfacción del usuario hacia el SR. En su trabajo los autores comparan 3 formas diferentes de presentar las recomendaciones. La primera corresponde a presentar una sugerencia directa visualizando el producto que obtuvo mayor puntaje. La segunda adiciona una explicación de por qué el usuario podría estar interesado en dicho producto. Finalmente, la tercera presenta no sólo el resultado sino explica cómo se realizó el cálculo para obtener la predicción y el producto sugerido. Los resultados mostraron que la confianza y aceptación eran directamente proporcionales a la información suministrada respecto al proceso de predicción. Es decir, los usuarios confían cuando conocen la razón de la recomendación recibida.

En el año 2009 la empresa Netflix[23] organizó un concurso que otorgaba un millón de dólares al grupo investigador que redujera el error de predicción de su SR un 10%. Este reto generó un alto interés dentro de la comunidad académica y surgieron importantes avances en el área de investigación. El principal aporte corresponde a la inclusión de los efectos temporales en los datos de entrada. Los investigadores notaron que los gustos de una persona suelen cambiar a lo largo del tiempo, por lo tanto, se debe incluir dicho efecto en los algoritmos diseñados. Al final del reto se establece la técnica de reducción dimensional como la técnica de mejor rendimiento para tratar el problema de predicción, siendo el algoritmo más escalable y exacto.

Más tarde, KOREN, BELL, y VOLINKSY[24] aplican la técnica de reducción dimensional mediante el enfoque de factorización de matrices. Esta técnica se fundamenta en que la matriz de puntajes, también denominada matriz de interacción usuario-producto U-P, es de bajo rango, esto es, las preferencias de los usuarios dependen de una pequeña cantidad de factores que determinan su elección sobre todos los productos disponibles. Los factores latentes se encuentran factorizando la

[23]BELL, Robert, KOREN, Yehuda, VOLINSKY, Chris. The belkor 2008 solution to the Netflix Prize. 2008. Statistics Research Department at AT&T Research, 2008.

[24] KOREN, Yehuda, BELL, Robert y VOLINSKY, Chris. Matrix factorization techniques for recommender systems. En: Computer. Agosto, 2009. vol. 42. no. 8, p. 30-37.

matriz de puntajes original en tres matrices diferentes: la primera que define un conjunto de factores relevantes; la segunda que relaciona a los usuarios con los factores encontrados, indicando qué tan importante es cada factor para cada usuario particular; y la última que relaciona a los productos con los factores, indicando qué tanto posee un producto de cada factor particular.

La factorización de la matriz se realiza típicamente mediante la técnica de descomposición de valores singulares (del inglés *singular value decomposition* SVD) y la teoría de completar matrices (del inglés *Matrix Completion MC)*. Diversos algoritmos como el propuesto por ZHOU *et al.*[25] han sido desarrollados para resolver tal problema. Sin embargo, la teoría de completar matrices está limitada por el problema de escasez de datos. La escasez de datos se refiere a que se conoce alrededor del 10% de los datos para predecir el 90% faltante. Para minimizar esta limitación, trabajos recientes se enfocan en la incorporación de información contextual. Por ejemplo, GOGNA y MAJUMDAR[26] proponen incorporar información geográfica y demográfica en el modelo del usuario.

Finalmente, enfoques recientes como el descrito por LI, CHEN y WANG[27] realizan un análisis avanzado de texto como realimentación usando los comentarios de los usuarios para obtener información adicional. Éstas técnicas se caracterizan por permitir la distinción de emociones, opiniones e intereses sobre tópicos específicos, los cuales incrementan la precisión en los resultados de los SR.

[25] ZHOU, Xun. SVD-based incremental approaches for recommender systems. En: Journal of computer and system sciences. Junio, 2015. vol. 81. no. 4, p.717-33.
[26] GOGNA, Anupriya, y MAJUMDAR, Angshul. Matrix completion incorporating auxiliary information for recommender system design. En: Expert systems with applications. Agosto, 2015. vol 42. no. 14, p. 5789.
[27] LI, Chen, CHEN, Guanliang y WANG, Feng. Recommender systems based on user reviews: the state of the art. En: User modeling and user-adapted interaction. Junio, 2015. vol. 25. no. 2, p.99-154.

2.2. MINIMIZACIÓN DEL RANGO DE UNA MATRIZ

El problema de minimizar el rango de una matriz es un problema de reducción dimensional que surgió en aplicaciones de análisis y diseño de sistemas de control. La complejidad computacional de éste problema es NP-completo[28], por esta razón, inicialmente fue abordado mediante heurísticas que buscaban una aproximación a la solución. El problema consiste en encontrar una representación de los datos en una estructura de menor dimensión que permita reducir la memoria requerida e incrementar la eficiencia en los cálculos realizados sobre los datos[29].

En el año 2001, FAZEL, HAITHAM y BOYD[30] proponen una heurística que reduce la complejidad computacional del problema. La heurística reemplaza la minimización del rango, por la minimización de la traza de la matriz. Los autores demuestran que la traza es una medida equivalente. Aunque el algoritmo demostró buenos resultados, su aplicación es limitada a matrices cuadradas y simétricas.

Más tarde, en el año 2008 CANDES y RECHT[31] aplican la técnica de minimizar el rango para abordar el problema de completar una matriz de datos (MC). Los autores demostraron matemáticamente que una matriz de bajo rango puede ser reconstruida o completada de forma exacta a partir de un conjunto de pocos datos observados. CANDES y RECHT plantean el problema mediante la minimización de la norma nuclear o suma de los valores singulares de la matriz. Su propuesta se caracteriza porque no está limitada a matrices simétricas y cuadradas.

---

[28] CANDES, Enmanuel, RECHT, Benjamin. Exact low-rank matrix completion via convex optimization. En: 46 Annual Allerton Conference on communication, control, and computing (4: 23-26, septie: Monticello, IL, USA). Communication, control, and computing 2008. Monticello, IL, USA: University of Illinois. 2008, p. 806–812.

[29] FAZEL, Maryam, HAITHAM Hindi, and BOYD, Stephen. A rank minimization heuristic with application to minimum order system approximation. En: American Control Conference (3:25-27, junio: Arlington, VA, USA). Proceedings of the 2001 American Control Conference. 2001. vol. 6, p. 4734-4739.

[30] Ibid.

[31] CANDES, *et al*. Op cit.

En el año 2010, CAI, CANDES y SHEN[32] proponen el primer algoritmo para encontrar la matriz de mínimo rango mediante la minimización de la norma nuclear. El algoritmo SVT (*Singular Value Thresholding*) es un algoritmo iterativo basado en la descomposición de valores singulares (SVD) con un operador de umbralización. A pesar de que obtuvo altos niveles de precisión, el SVT tiene una alta complejidad computacional. Por esta razón, HUI, *et al*.[33] proponen reemplazar la factorización SVD por la factorización de productos Kronecker; aunque la precisión es comparable, los algoritmos basados en SVD continúan siendo superiores.

Por otro lado, el método de solución propuesto por SHIQIAN, GOLDFAR y CHEN[34] se basa en el método de punto fijo para determinar la descomposición SVD. El algoritmo denominado FPC (del inglés *Fixed Point Continuation*) resuelve el problema mediante una aproximación de Monte Carlo. FPC es uno de los algoritmos propuestos más robustos para resolver el problema de completar matrices de bajo rango[35].

Las técnicas de agrupamiento, los métodos basados en factorización de matrices y reducción dimensional han minimizado las limitaciones de escalabilidad, escasez y sinonimia de los métodos tradicionales para resolver el problema de SR[36]. Sin embargo, tienen la desventaja de tener alta complejidad computacional. Es por esto que las computaciones se realizan fuera de línea (del inglés *offline)*, lo cual, hace que se ignore la naturaleza dinámica de los datos y el contexto de las recomendaciones. Para resolver este problema, se han presentado recientemente

---

[32] CAI, Jian, CANDES Emmanuel y SHEN Zuowei. A singular value thresholding algorithm for matrix completion. En: SIAM Journal on Optimization. Marzo, 2010. vol. 20. no. 4, p.1956-1982.
[33] HUI, Zhao, *et al*. A Scalable spectral relaxation approach to matrix completion via kronecker products. 2011.
[34] SHIGIAN, Ma, GOLDFARB, Donald y CHEN, Lifeng. Fixed point and Bregman iterative methods for matrix rank minimization. En: Mathematical Programming. Junio, 2001. vol. 128. no. 1-2, p. 321-353.
[35] MICHENKOVÁ, Marie. Numerical algorithms for low-rank matrix completion problems. 2011.
[36]SARWAR, Badrul, *et al*. Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering. En: Proceedings of the fifth international conference on computer and information technology. 2002, vol. 1.

esquemas incrementales el cual consiste en establecer un modelo base con información estática y actualizar únicamente los datos dinámicos. LUO, *et al.*[37] proponen una metodología incremental en la cual combinan un modelo estático y un modelo incremental que considera información dinámica. Los resultados presentan una exactitud comparable con las demás metodologías mejorando el rendimiento computacional.

Finalmente, MICHENKOVÁ[38] realiza una evaluación y comparación de 9 solucionadores disponibles gratuitamente para minimizar el rango de una matriz. La autora distingue dos enfoques de solución. El primero corresponde a los algoritmos de optimización convexa basados en la minimización de la norma nuclear. Dentro de esta categoría el algoritmo LMaFit[39] presenta el mejor rendimiento, exactitud y escalabilidad. El segundo enfoque corresponde a la minimización de la versión Lagragiana del problema. Dentro de éste enfoque se destaca el algoritmo FPC descrito anteriormente[40].

[37]LUO, Xiaohua *et al*. An incremental-and-static-combined scheme for matrix-factorization-based collaborative filtering. En: IEEE Transactions On Automation Science & Engineering [En línea]. Enero, 2016. Vol. 13. no. 1, p. 333-343.
[38] MICHENKOVÁ. Op. Cit.
[39] ZAIWEN, Wen, YIN, Wotao, ZHANG, Yin. Solving a low-rank factorization model for matrix completion by a nonlinear successive overrelaxation algorithm. En: Mathematical Programming Computation. Diciembre, 2012. vol. 4. no. 4, p.333–361
[40] SHIGIAN. Op. Cit.

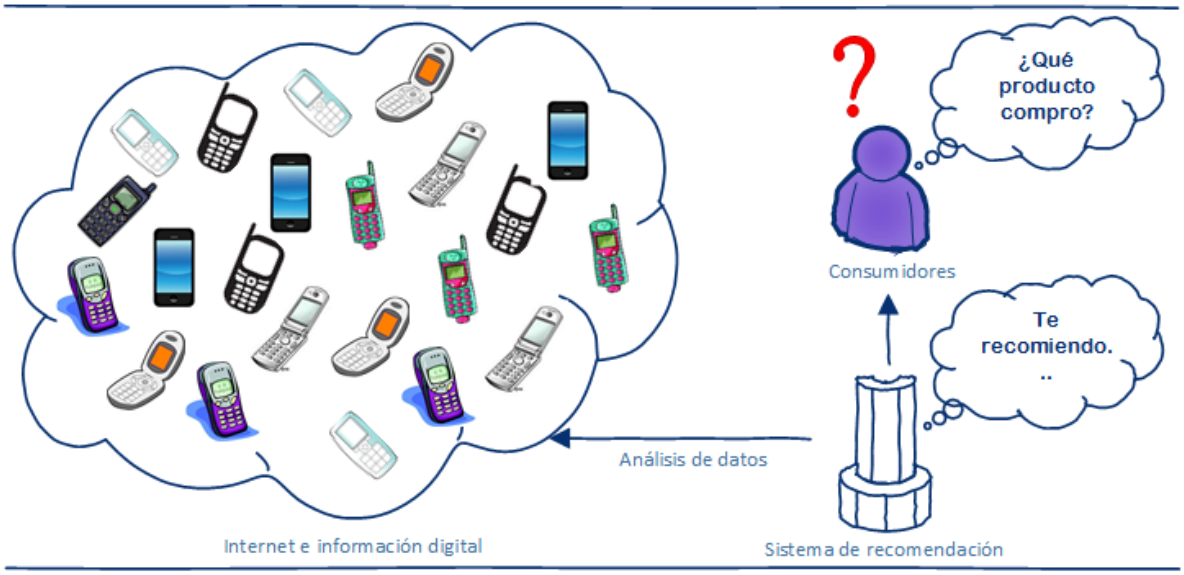
3. MARCO TEÓRICO

3.1.SISTEMAS DE RECOMENDACIÓN[41]

Un sistema de recomendación (SR) es una herramienta computacional para solucionar el problema de sobrecarga de información. El objetivo de un SR es recomendar el producto más apropiado para un usuario en un contexto específico. Los algoritmos de SR estiman un puntaje que mide el grado de satisfacción de los usuarios con un producto a partir del análisis de un conjunto de datos almacenados.

Las entidades que participan en un SR de productos son: los usuarios, los productos y el sistema en sí mismo. En la Figura 3 se observa una vista conceptual de la participación de un SR como un puente de comunicación y filtrado entre un usuario que busca un producto y una plataforma que ofrece una gran cantidad de posibilidades.

Figura 3 Vista conceptual de los SR



[41] RICCI, Francesco. Op. cit.

Teniendo en cuenta que un SR es un sistema de procesamiento de información que analiza datos pasados para generar las recomendaciones, la calidad de los datos de entrada es fundamental. Los datos de entrada son generalmente: las características de los productos que van a ser sugeridos, las características de los usuarios que recibirán las recomendaciones y las relaciones que se generan durante la interacción usuario-producto expresadas como un puntaje.

Los SR se clasifican en dos amplias categorías según el nivel de personalización. Aquellas herramientas que tienen en cuenta las características, gustos y preferencias individuales de los usuarios se denominan sistemas de recomendación personalizados. Por otro lado, aquellas que hacen estimaciones teniendo en cuenta únicamente preferencias generales se denominan no-personalizados.

3.1.1 Sistemas de recomendación no personalizados Los SR no personalizados calculan promedios de todos los puntajes almacenados para generar las recomendaciones. Estos sistemas no modelan a cada usuario particular ni consideran información contextual que atienda gustos particulares. Las aplicaciones típicas incluyen recomendación de revistas, noticias y páginas web. Tipicamente, la presentación de las sugerencias es una lista de tendencias para mostrar los n productos más populares de la categoría de interés.

3.1.2 Sistemas de recomendación personalizados Un SR personalizado es aquél que almacena y utiliza datos de los gustos, preferencias y estilo de vida del usuario particular para recomendar los productos apropiados. Bajo este enfoque el usuario es modelado mediante un perfil que contiene diferentes propiedades, ya sean de tipo demográfica (edad, género, nacionalidad) o de comportamiento (páginas que visita, actividades cotidianas que realiza, etc). Un SR personalizado se caracteriza por mostrar un resultado diferente a cada usuario aun cuando se esté buscando una misma categoría de productos. Este tipo de herramientas ha sido ampliamente

abordado por ser el más empleado en los sitios de comercio electrónico como Amazon.com, Netflix y MovieLens[42].

Un ejemplo que contrasta los SR personalizados con los no-personalizados es el sitio web de Amazon.com[43]. Los usuarios no registrados son recomendados con los artículos de tendencia, los más vendidos o más importantes del día, lo cual es un enfoque no personalizado. Sin embargo, si el usuario ingresa a su cuenta personal y realiza una búsqueda, las recomendaciones generadas son personalizadas y resultan de una estimación basada en el perfil almacenado (historial de compra, información demográfica, puntajes, etc.).

3.2.CLASIFICACIÓN DE DATOS DE ENTRADA

Los datos empleados para realizar las estimaciones de las preferencias se clasifican en dos categorías según la forma en que son adquiridos: datos explícitos o implícitos.

3.2.1 Datos explícitos Los datos explícitos son los puntajes proporcionados directamente por el usuario. Comúnmente se adquieren mediante preguntas directas que hace el SR al usuario. Por ejemplo, los sistemas que solicitan un puntaje en una escala numérica son sistemas que emplean datos explícitos como fuente de datos principal. Estos datos son almacenados tradicionalmente en una estructura matricial denominada matriz de interacción usuario-producto U-P.

3.2.2 Datos Implícitos Son datos inferidos a partir del seguimiento de la actividad y comportamiento de un usuario. Por ejemplo, las páginas web que visita, el tiempo de navegación, la ubicación geográfica, los comentarios u opiniones que

[42] MOCEAN, Loredana. Op. Cit.
[43]LINDEN, Greg, SMITH, Brent y YORK, Jeremy. Amazon.com recommendations: Item-to-item collaborative filtering. En: IEEE Internet Computing. Vol. 7, No.1 (2003) p.76-80.

proporciona. Usualmente, el usuario no es consciente que su comportamiento de navegación en la web o de compra define sus preferencias y tendencias, por tal razón los datos implícitos resultan ser muy valiosos para definir el perfil de usuario.

3.3.ENFOQUES DE RECOMENDACIÓN

Los SR han sido clasificados según el método que emplean para generar las recomendaciones. Por un lado, se encuentra el enfoque basado en contenido, el cual analiza características inherentes de los productos para encontrar las similitudes. Por otro lado, el enfoque basado en filtrado colaborativo emplea los puntajes almacenados en la matriz de interacción U-P o datos explícitos para hacer las estimaciones.

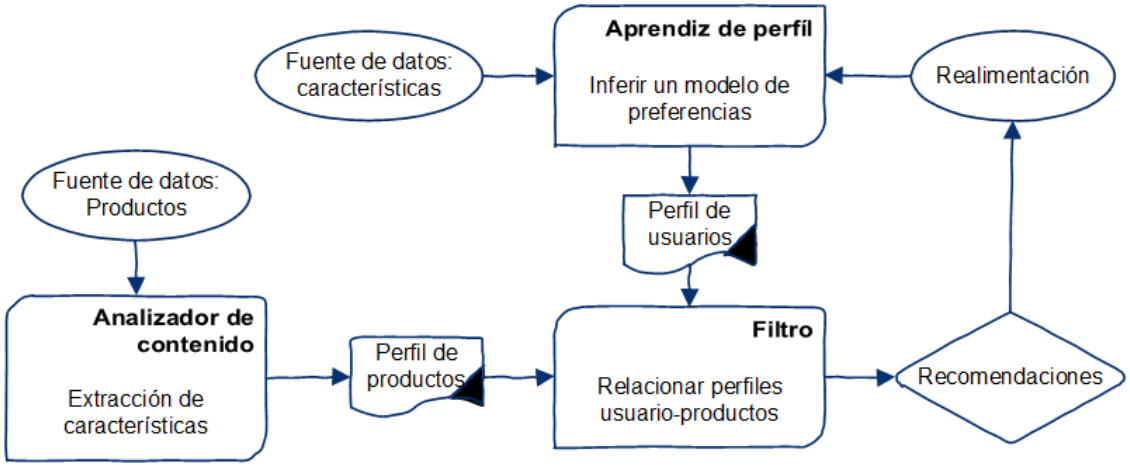
3.3.1 Filtrado basado en contenido El filtrado basado en contenido recomienda a los usuarios los productos más similares a aquellos que han consumido en el pasado o recomienda los productos consumidos recientemente por los usuarios más similares. Para determinar la similitud se construye un perfil de usuario y un perfil del producto.

El perfil del usuario describe la importancia que tiene para un usuario la presencia de una característica determinada. El perfil del producto describe el nivel de presencia de una característica en el producto evaluado. Finalmente, los perfiles construidos se comparan y se relacionan los que son más similares.

La métrica de similitud se determina mediante el análisis de las características inherentes y propias de los productos: descripciones, funciones, marca, categoría, precio, etc. Cuando los perfiles contienen etiquetas se suelen representar en escalas numéricas para realizar la comparación.

La Figura 4 muestra el esquema general de un SR basado en contenido. El esquema contiene tres elementos principales. En primer lugar, el analizador de contenido corresponde a las técnicas que extraen características de los productos a ser recomendados para definir sus perfiles. Seguidamente, el aprendiz de perfil corresponde a las técnicas empleadas para construir el perfil de usuario en base a características previas y a la realimentación generada de la interacción con el sistema. Finalmente, el filtro es la herramienta que relaciona los perfiles de usuario-productos para generar las recomendaciones.

Figura 4 Esquema de un SR basado en contenido



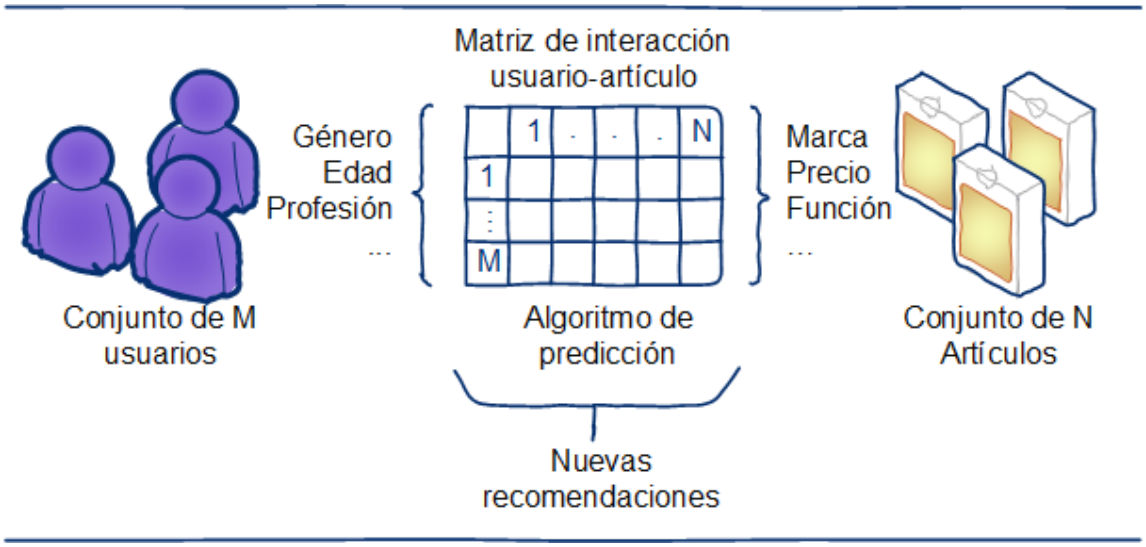
Modificado de: RICCI, Francesco, *et al.* 2011

3.3.2 Filtrado colaborativo El filtrado colaborativo (FC) es la técnica de predicción más empleada y exitosa en ambientes de comercio electrónico, como los casos de Netflix, eBay y Amazon[44]. Esta técnica está basada en las relaciones de preferencia o que existen entre un usuario y un producto. Tradicionalmente, las relaciones de preferencia se codifican en una matriz denominada matriz de interacción usuario-producto (U-P) donde el valor de relación es el puntaje otorgado por el usuario al producto.

[44] YUE, Shi, LARSON, Matha y HANJALIC, Alan. Collaborative filtering beyond the user-item matrix, a survey of the state of the art and future challenges. En: ACM computer survey. Julio, 2014, vol. 47, no. 1, p. 1–45.

La técnica de FC inicialmente fue diseñada para emplear solo datos explícitos contenidos en la matriz U-P. Sin embargo, han surgido modificaciones que permiten emplear información contextual, o etiquetas para obtener un modelo más realista de los usuarios y de los productos. En la Figura 5 se ilustra una vista conceptual de los SR bajo el enfoque de FC.

Figura 5 Vista conceptual del enfoque de FC



Sea $\mathrm{C} = \{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_M\}$ un conjunto de M usuarios, $\mathrm{P} = \{\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_N\}$ un conjunto de N productos y sea $\mathbf{R} \in \mathbb{R}^{M \times N}$ la matriz U-P cuyos valores conocidos se almacenan en una matriz $\mathbf{M} \in \mathbb{R}^{M \times N}$; el problema de FC consiste en recomendar a cada usuario $c_i \in \mathrm{C}$ una lista de sus n productos preferidos $p_1, \ldots, p_n \in \mathrm{P}$. Para generar las recomendaciones se calcula una matriz estimada $\widehat{\mathbf{R}} \in \mathbb{R}^{M \times N}$ de $\mathbf{R}$ a partir de los datos conocidos en $\mathbf{M}$. De esta manera, se recomiendan aquellas relaciones estimadas en la matriz $\widehat{\mathbf{R}}$ que obtuvieron los puntajes más altos.

3.4.CLASIFICACIÓN DE TÉCNICAS DE FC

3.4.1 Algoritmos basados en memoria El objetivo de estos algoritmos es encontrar similitudes con los puntajes contenidos en **R.** Se llaman basados en memoria porque emplean las decisiones que el usuario haya tomado en el pasado para determinar las similitudes y estimar los puntajes faltantes. Existen dos enfoques de análisis de la matriz: encontrar similitudes de producto-producto (ver figura 6) en el cual se recomiendan los productos más similares a aquellos que el usuario adquirió en el pasado, o encontrar similitudes de usuario-usuario (ver figura 7) en el cual se recomiendan los productos de acuerdo a los gustos de los usuarios más similares al usuario que va a ser recomendado[45].

Figura 6 Similitud producto-producto

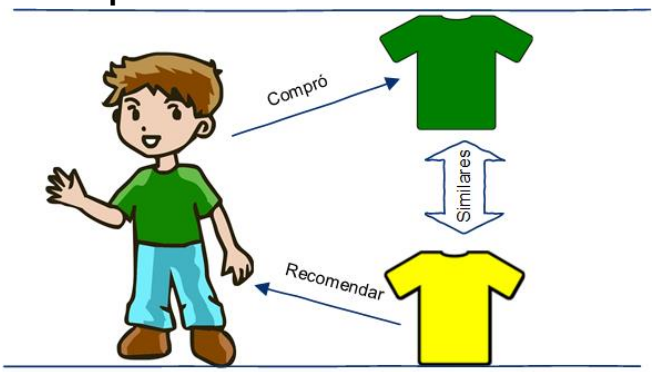
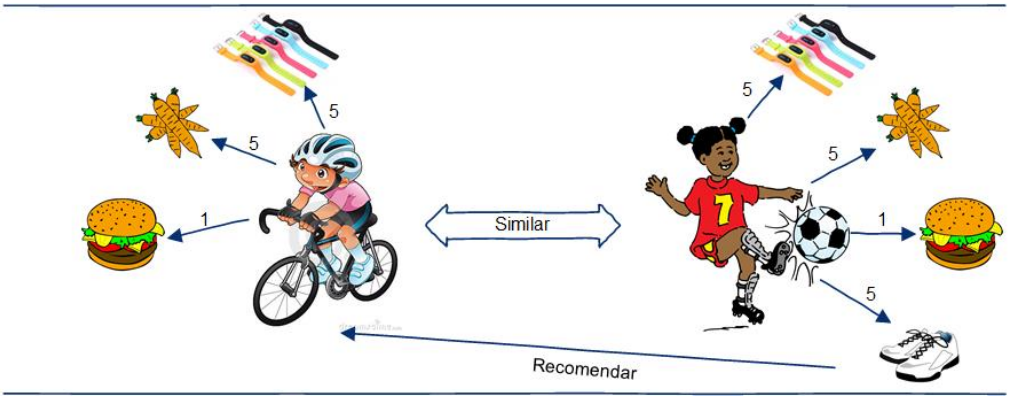


Figura 7 Similitud usuario-usuario



[45] ADOMAVICIUS. Op. cit.

La formulación matemática del enfoque de similitud usuario-usuario para obtener la matriz estimada $\widehat{\mathbf{R}}$ consiste en estimar cada puntaje $\hat{\mathrm{r}}_{ij}$ como un promedio ponderado de los puntajes m_{kj} otorgados por una vecindad; la ponderación está dada teniendo en cuenta la similitud entre el par de usuarios evaluados. En (1) se describe la formulación matemática.

$$\hat{\mathrm{r}}_{ij} = \frac{1}{\alpha} \sum_{k \in Z_i} sim(i,k)\, m_{kj}, \tag{1}$$

donde $\hat{\mathrm{r}}_{ij}$ es la estimación del puntaje asignado por el usuario i al producto j, α es una constante de normalización, Z_i es el conjunto de K vecinos del usuario i y $sim(i,k)$ representa la similitud entre el usuario i y el usuario k. En el enfoque producto-producto las estimaciones se reemplazan de la misma manera, reemplazando los índices que indican usuarios por productos.

Por otro lado, la métrica para calcular la similitud es principalmente la métrica de correlación de Pearson (2) o la distancia del coseno (3).

$$sim(i,k) = \frac{\sum_{s \in \mathrm{S}_{ik}} (m_{is} - \bar{m}_i)(m_{ks} - \bar{m}_k)}{\sqrt{\sum_{s \in \mathrm{S}_{ik}} (m_{is} - \bar{m}_i)^2 \; \sum_{s \in \mathrm{S}_{ik}} (m_{ks} - \bar{m}_k)^2}}, \tag{2}$$

donde S_{ik} es el conjunto de los productos calificados por el usuario i y el usuario k, r_{is} es el puntaje asignado por el usuario i al producto s y $\bar{r}_i$ es el promedio de los puntajes proporcionados por el usuario i.

$$sim(l,k) = \cos(\vec{l}, \vec{k}) = \frac{\vec{l} \cdot \vec{k}}{\|\vec{l}\|_2 \times \|\vec{k}\|_2}, \tag{3}$$

donde $(\cdot)$ denota el producto punto entre dos vectores y $\| \|_2$ indica la norma euclidiana.

Los algoritmos basados en memoria tienen la desventaja de que la complejidad computacional es doblemente exponencial, además la precisión de los resultados depende de la métrica para calcular la similitud entre usuarios y/o productos[46].

3.4.2 Algoritmos basados en modelos Este enfoque de algoritmos se basa en entrenar un modelo de predicción a partir de la información contenida en la matriz $\mathbf{R}$. Para entrenar el modelo, es necesario tener como datos de entrada un conjunto de datos correspondientes a ciertos parámetros del usuario a ser recomendado y de los productos. Una vez el modelo está entrenado se calculan los puntajes faltantes mediante una función de predicción. Matemáticamente, éste enfoque se formula como se describe en (4).

$$f(g_i, q_j) \longrightarrow \hat{r}_{ij}, i = 1, \dots, M\ , j = 1, \dots, N, \qquad (4)$$

donde g_i, q_j son el conjunto de parámetros para el usuario i y el producto j y $f(g_i, q_j)$ es la función de predicción para obtener la matriz $\widehat{\mathbf{R}}$. Algunos métodos desarrollados para este enfoque emplean redes bayesianas, modelos de semántica latente y modelos basados en factorización de matrices (FM) y reducción dimensional (RD). Los modelos de RD se caracterizan por la alta precisión, eficiencia y escalabilidad que han alcanzado[47].

3.5.ALGORITMOS BASADOS EN FACTORIZACIÓN DE MATRICES (FM)

El fundamento de éste enfoque es encontrar una representación de la matriz original compuesta por la multiplicación de dos o más sub-matrices. Dada la matriz $\mathbf{M} \in \mathbb{R}^{M \times N}$ con valores conocidos en el conjunto de posiciones $\Omega = \{(i,j) \in \mathbb{N}^2 \mid m_{ij}\ es\ conocido\ \}$, éste problema se resuelve típicamente mediante el problema de optimización en (5).

[46] YUE, Shi, Op. cit.
[47] BELL, Robert. op cit.

$$\mathbf{U}^*, \mathbf{V}^* = \underset{\mathbf{U},\mathbf{V}}{\operatorname{argmin}} \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{ij}(m_{ij} - U_i^{\mathrm{T}}V_j)^2 + \frac{\lambda_{\mathbf{U}}}{2}\|\mathbf{U}\|_{\mathbf{F}}^2 + \frac{\lambda_{\mathbf{V}}}{2}\|\mathbf{V}\|_{\mathbf{F}}^2 \qquad (5)$$

donde $\mathbf{U} \in \mathbb{R}^{M\times r}$ es la matriz que relaciona r factores latentes con los M usuarios y $\mathbf{V} \in \mathbb{R}^{N\times r}$ es la matriz que relaciona r factores latentes con los N productos. $\lambda_{\mathbf{U}}$ y $\lambda_{\mathbf{V}}$ son parámetros de regularización para evitar el sobreajuste de los datos; finalmente I_{ij} representa una función indicador que toma el valor de 1 si $(i,j) \in \Omega$ y 0 en otro caso.

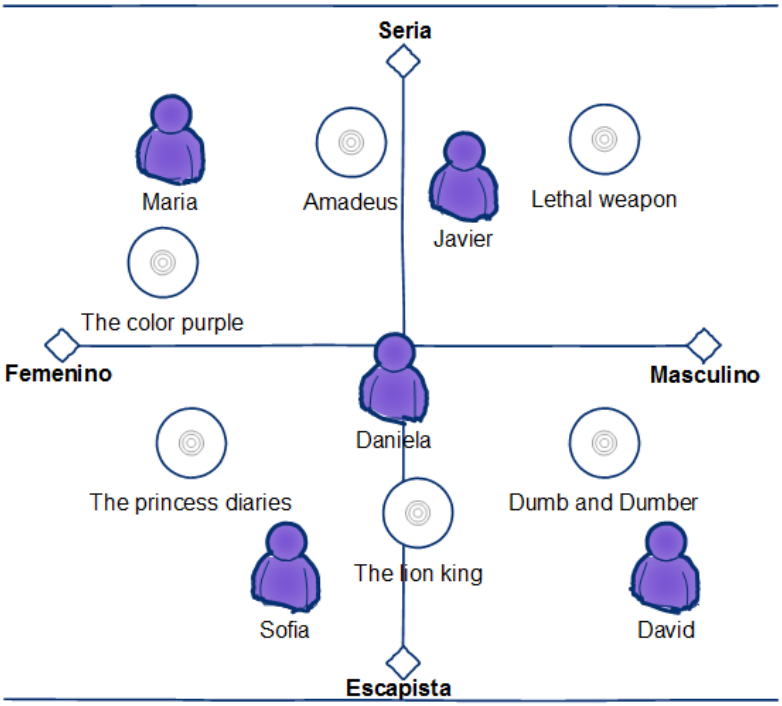
Estos métodos tienen la ventaja de permitir la incorporación de información adicional como realimentación implícita, efectos temporales y niveles de confianza con el objetivo de mejorar la precisión y escalabilidad del algoritmo.

3.6.REDUCCIÓN DIMENSIONAL (RD)

La RD consiste en la representación de la matriz $\mathbf{R} \in \mathbb{R}^{M\times N}$ mediante una matriz de bajo rango, es decir una representación de menor dimensión. El rango de una matriz corresponde a la cantidad de filas o columnas independientes de la matriz. Asumiendo que las preferencias de los usuarios y características de los productos están determinadas por un pequeño conjunto de factores, la matriz $\mathbf{R}$ debe ser de bajo rango, esto es $\mathrm{rank}(\mathbf{R}) \ll \min(M,N)$, donde $\mathrm{rank}(\cdot)$ indica el rango de la matriz.

La Figura 6 ilustra un ejemplo de un SR de películas el cual se caracteriza mediante dos factores latentes: categoría del contenido de la película y género de la persona. Lo anterior permite representar toda la información que antes era de dimensión $\min(M,N)$ en un espacio bidimensional. De esta manera, las estimaciones de los puntajes se obtienen calculando la distancia entre los usuarios y productos en el nuevo espacio de baja dimensión, donde la posición de cada usuario y producto corresponde a una coordenada en el espacio definido.

Figura 6 Modelo latente de un SR de películas

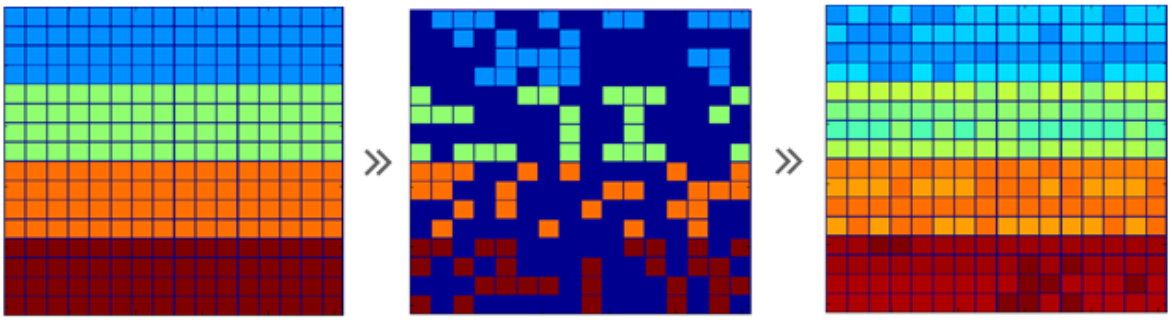


Modificado de: KOREN, Yehuda, *et al.,* 2009.

3.7.TEORÍA DE COMPLETAR MATRICES DE BAJO RANGO

La teoría de completar matrices propone que una matriz de bajo rango puede ser recuperada a partir de un pequeño conjunto de datos conocidos. La Figura 7 ilustra la teoría de completar matrices de bajo rango en donde la matriz desconocida de bajo rango $\mathbf{R}$ tiene un conjunto de pocos valores conocidos almacenados en la matriz $\mathbf{M}$; a partir de estos datos conocidos se estima una aproximación $\hat{\mathbf{R}}$ de la matriz original.

Figura 7 Teoría de completar matrices de bajo rango



Matemáticamente el problema se plantea como: Sea $\mathbf{R} \in \mathbb{R}^{M \times N}$ una matriz de bajo rango, $\text{rank}(\mathbf{R}) \ll \min(M, N)$ y $\mathbf{M} \in \mathbb{R}^{M \times N}$ una matriz que contiene los valores conocidos de la matriz $\mathbf{R}$ en el conjunto de posiciones $\Omega = \{(i,j) \in \mathbb{N}^2 \mid m_{ij}\ es\ conocido\}$, encontrar la estimación $\hat{\mathbf{R}}$ resolviendo el problema de optimización en (6).

$$\begin{aligned} &\underset{\hat{\mathbf{R}}}{\text{minimizar}}\ \text{rank}(\hat{\mathbf{R}}) \\ &\text{sujeto a}\ \hat{r}_{i,j} = m_{i,j,}\ (i,j)\ \in\ \Omega \end{aligned} \tag{6}$$

El problema en (6) se caracteriza por su complejidad computacional NP-completo[48]. Sin embargo, se demostró que bajo ciertas condiciones éste problema se puede resolver eficientemente[49]. En primer lugar, los datos observados deben ser suficientes para determinar la estructura de bajo rango. En segundo lugar, la muestra debe ser uniforme de tal forma que cada fila y columna tengan al menos un dato observado. En términos del problema específico de los SR, las condiciones expuestas por la teoría de completar matrices indican dos factores importantes. Primero, cada usuario debe haber proporcionado al menos un puntaje y cada producto debe haber sido calificado al menos una vez. Segundo, la cantidad de puntajes conocidos dada por $|\Omega| = m$ debe ser lo suficientemente grande para poder estimar los puntajes restantes.

[48] CANDES y RECHT, Op. cit.
[49] Ibid.

Para resolver el problema en (6) se han propuesto dos enfoques que reducen la complejidad del problema original. El primer enfoque consiste en reemplazar la minimización del rango por la minimización de la norma nuclear. Este enfoque corresponde a la relajación convexa del problema. El algoritmo *Fixed Point Continuation* (FPC)[50] es un solucionador que resuelve el problema en (6) mediante esta propuesta. El segundo enfoque se basa en la minimización de la norma Frobenius y corresponde a la solución de la versión Lagragiana del problema original. El algoritmo *Low-rank Matrix Fitting* (LmaFit)[51] es uno de los solucionadores más eficientes bajo este enfoque.

3.8. LIMITACIONES DE LOS SR

En la literatura se han enunciado tres grandes retos para los SR:

- Mejorar el rendimiento computacional, es decir, aumentar el número de recomendaciones calculadas por segundo.
- Incrementar la precisión de las sugerencias calculadas, es decir, acercarse más al puntaje real que un usuario le otorgaría a un producto.
- Obtener gran cobertura a pesar de tener datos dispersos. La cobertura se refiere a la cantidad de productos y usuarios que pueden ser recomendados empleando los pocos datos conocidos.

Los dos primeros retos resultan ser conflictivos entre sí, observe que al aumentar la velocidad de procesamiento se pierde precisión en los resultados, adicionalmente

[50] SHIGIAN, Ma. Op. Cit.
[51] ZAIWEN, Wen, YIN, Wotao, ZHANG, Yin. Solving a low-rank factorization model for matrix completion by a nonlinear successive overrelaxation algorithm. En: Mathematical Programming Computation. Diciembre, 2012. vol. 4. no. 4, p.333–361.

este problema se complica considerando la naturaleza dinámica de los datos y la gran cantidad de usuarios y productos disponibles.

3.9.EVALUACIÓN DE LOS SR

La evaluación de los SR considera diferentes factores según el propósito para el cual fue desarrollado. Inicialmente solo se evaluaba la capacidad de predicción, es decir, la exactitud y precisión de las puntuaciones calculadas. Sin embargo, estos no son los únicos factores que influyen en la calidad de un SR. Actualmente, existen 3 categorías de métricas de medición: métricas de exactitud, métricas de soporte a decisiones y métricas organizacionales.

3.9.1 Métricas de Exactitud Se usan para evaluar la diferencia entre la predicción realizada y el valor real del puntaje del usuario sobre los productos. Siendo p_i el valor de la predicción y r_i el valor real, hay 3 métricas de exactitud descritas en (7), (8) y (9).

$$\text{EAM} = \text{Error Absoluto Medio} = \frac{\sum_{puntuaciones} |p_i - r_i|}{\#\ de\ puntuaciones} \tag{7}$$

$$\text{ECM} = \text{Error Cuadrático Medio} = \frac{\sum_{puntuaciones} (p_i - r_i)^2}{\#\ de\ puntuaciones} \tag{8}$$

$$\text{RECM} = \text{Raíz del Error Cuadrático Medio} = \sqrt{\frac{\sum_{puntuaciones} (p_i - r_i)^2}{\#\ de\ puntuaciones}}, \tag{9}$$

3.9.2 Métricas de soporte de decisiones Este enfoque busca medir la capacidad del SR para guiar a los usuarios a tomar buenas decisiones. Esto es, elegir los “buenos” productos y evitar los “malos”. Las principales métricas empleadas son:

$$\text{Precisión} = \frac{N_{rs}}{N_s}, \tag{10}$$

donde N_{rs} es el número de elementos relevantes recuperados y N_s es la cantidad de elementos recuperados. Es decir, la precisión es el porcentaje de los elementos recuperados que son “relevantes”. Con esta métrica se mide la capacidad del algoritmo para hacer buenas elecciones.

$$\text{Exhaustividad} = \frac{N_{rs}}{N_r}, \tag{11}$$

donde N_r es el número de elementos relevantes. Es decir, la exhaustividad (del inglés *recall*) es el porcentaje de elementos relevantes que fueron recuperados. Con esta métrica se mide la capacidad del algoritmo para seleccionar todos los elementos relevantes.

3.9.3 Métricas organizacionales Este enfoque está interesado en medir los SR de acuerdo a los objetivos perseguidos por las organizaciones. Se evalúan tres aspectos principalmente:

- Cobertura: Es la cantidad de productos para los cuales el SR es efectivo. Se desea un SR que haga buenas predicciones para todo el portafolio de una organización que uno que funciona en un conjunto reducido de productos.
- Diversidad: mide la diferencia entre los elementos que son recomendados. Se pretende evitar que recomiende varios productos muy similares a los que ya recomendó en el pasado.
- Novedad: Es la capacidad de sorprender al usuario con las recomendaciones realizadas y satisfacerlo con resultados no esperados.

4. METODOLOGÍA

4.1. DELIMITACIÓN Y CARACTERÍSTICAS DEL PROBLEMA

El propósito de este proyecto es aplicar la técnica de minimización del rango de una matriz para obtener las estimaciones de un SR de productos y mejorar su rendimiento. De acuerdo a esto, el problema está delimitado por las características que se presentan en la Tabla 3. Estas características delimitan el dominio de aplicación para la selección de la BD, diseño, modelamiento, implementación y evaluación del algoritmo propuesto.

Tabla 3 Características para delimitar el problema

Clasificación	SR personalizado basado en las preferencias de los usuarios.
Datos de entrada	Datos explícitos en escala numérica. Los datos deben estar representados en una matriz U-P.
Algoritmo de predicción	Algoritmo propuesto basado en optimización convexa y teoría de completar matrices de bajo rango.
Datos de salida	Recomendaciones de tipo lista top-n para cada usuario del sistema.

Adicional a los requisitos expresados en la delimitación del problema (Tabla 3), se consideran los siguientes factores para seleccionar adecuadamente la BD sobre la cual se evaluará el algoritmo propuesto:

- Deben existir publicaciones de trabajos de investigación que hayan empleado la BD de tal forma que se puedan comparar resultados.
- El problema presentado debe ser uno en el que sea relevante y deseable aplicar la técnica de reducción dimensional.

4.2.SELECCIÓN DE LA BASE DE DATOS (BD)

En la Tabla 4 se presentan las BD del área de SR que se encuentran disponibles gratuitamente para uso académico. Los datos presentados corresponden a 4 características relevantes en la selección de la BD como objeto de estudio en este proyecto. Las características comparadas corresponden a la cantidad de productos, la cantidad de usuarios, el dominio de aplicación y el tipo de datos que almacena (etiquetas o puntajes explícitos).

Tabla 4 Consulta de BD disponibles

No.	BD	Cantidad Productos	Cantidad Usuarios	Dominio	Tipo de datos		% puntajes
					Etiquetas	Explícitos	
1	*MovieLens*	1682	943	Películas		X	5.9 %
2	*Delicious Bookmarks*	105000	1867	Sitios Web	X		NA
3	*Last.fm*	92800	1892	Música	X		NA
4	*IMDb/ Rotten Tomatoes*	10197	2113	Películas	X		NA
5	*BookCrossing*	271379	278858	Libros		X	1.52 %
6	*Jester*	100	73496	Chistes		X	55.84 %
7	*EachMovie*	1628	72916	Películas		X	2.37 %
8	*Restaurant & consumer*	130	138	Comida		X	6.5 %

La elección de la BD se realizó mediante la evaluación de los siguientes criterios:

1. De acuerdo a la delimitación presentada en la Tabla 3, los datos de entrada deben ser de tipo explicito, por lo tanto las BD 2,3 y 4 son descartadas del conjunto de elegibles.

2. De acuerdo a las condiciones de la teoría de completar matrices, el conjunto de datos observados debe ser lo suficientemente grande para encontrar su estructura de bajo rango. Por lo tanto, se verificó mediante simulaciones preliminares la convergencia de los solucionadores evaluados para cada una de las BD. Como resultado se obtuvo que el porcentaje de puntajes proporcionados por las BD 1, 5 y 7 no son suficientes y quedaron descartadas del conjunto de elegibles.

3. Como criterio final se analizó el contexto del problema abordado por las BD restantes. “Jester” es una BD que almacena puntajes proporcionados anónimamente de un SR de bromas y chistes. Por su parte “Restaurant & Consumer” es una BD obtenida mediante un prototipo de SR de restaurantes denominado “SURFEOUS”, cuyo objetivo es ayudar a los diferentes usuarios a encontrar el lugar apropiado para ir a comer.

   Considerando que este proyecto se realiza en un contexto que permita integrar las herramientas computacionales con objetivos empresariales, por ejemplo aplicar técnicas de mercadeo focalizado. La BD elegida para evaluar el rendimiento del algoritmo propuesto corresponde a la BD No. 8, cuyo objetivo resulta ser más interesante y relevante en el ámbito comercial.

4.3.SELECCIÓN DE MÉTRICAS DE EVALUACIÓN

Con el objetivo de comparar los resultados del algoritmo propuesto, se escogen las métricas de soporte a decisión, precisión (10) y exhaustividad (11) que fueron empleadas para comparar el algoritmo propuesto con los resultados presentados en un trabajo previo empleando la misma BD. Adicionalmente se presentan resultados en términos de exactitud con la métrica de EAM para proporcionar un marco que permita comparaciones con trabajos futuros. Finalmente se presenta un análisis del tiempo de cómputo del algoritmo propuesto.

4.4.DESCRIPCIÓN DE LA BD

La Tabla 5 presenta la información general de la BD "Restaurant & Consumer" la cual se encuentra disponible en el repositorio de aprendizaje automático UCI[52]. Aunque la BD únicamente contiene un 6,4716 % de valores conocidos, simulaciones preliminares permitieron verificar el cumplimiento de las condiciones de la teoría de completar matrices mediante la minimización de rango, la cual es el fundamento del algoritmo propuesto.

Tabla 5 Información base de datos *"Restaurant and consumer"*

Nombre	Restaurant & consumer data
Número de instancias	138 usuarios – 130 restaurantes
% de puntajes	6.4716 %
Fecha	04 de Agosto de 2012
Fuente	Rafael Ponce y Juan Gabriel González. Department of Computer Science. National Center for Research and Technological Development CENIDET, México.
Trabajos previos	Effects of relevant contextual features in the performance of a restaurant recommender system[53].

4.5.PLANTEAMIENTO DEL PROBLEMA SR "SURFEOUS"

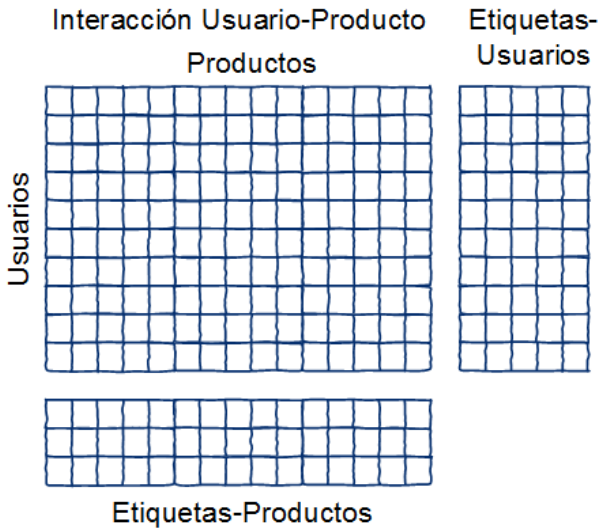
Típicamente, los usuarios acuden por una orientación para elegir un restaurante donde comer en una fecha o evento especial. "*Restaurant & Consumer data*" es la BD del sistema de recomendación "SURFEOUS: ¿Dónde comer?"[54]. SURFEUOUS

[52]UCI. Machine Learning Repository. Center for Machine Learning and Intelligent Systems. [En línea]. Disponible en: https://archive.ics.uci.edu/ml/datasets/Restaurant+%26+consumer+data [18/01/16].
[53] VARGAS, Blanca, GONZÁLEZ, Gabriel y PONCE, Rafael. Effects of relevant contextual features in the performance of a restaurant recommender system. En: ACM Conference on Recommender Systems (5:23-27, octubre: Chicago, IL, USA). Proceedings of the 5th ACM RecSys. New York, NY, USA: ACM, 2011, p.85-92.
[54] Ibíd. p.1

es un prototipo de SR diseñado para emplear etiquetas e información contextual en el proceso de recomendación. Su propósito es recomendar restaurantes de acuerdo a las características de los usuarios y al contexto específico del evento.

El algoritmo de predicción de SURFEOUS corresponde a un enfoque de FC basado en leyes semánticas, que emplea tres matrices de datos: la matriz de usuario-etiquetas, la matriz de producto-etiquetas y la matriz de interacción usuario-producto (U-P). Las predicciones de los puntajes faltantes se obtienen fusionando los resultados del análisis semántico de cada matriz por separado y se presentan en una lista top-n[55]. La Figura 8 ilustra el esquema de la estructura de datos empleada por VARGAS, Blanca *et al*[56]. para el SR SURFEOUS. Esta estructura fue planteada por TSO-SUTTER, Karen, *et al*.[57] para SR conscientes del contexto.

Figura 8 Esquema de la estructura de datos



Modificado de: TSO-SUTTER, Karen, *et al*., 2008

[55] TSO-SUTTER, Karen, MARINHO, Leandro y SCHMDIT, Lars. Tag-aware recommender systems by fusion of collaborative filtering algorithms. En: Proceedings of the 2008 ACM symposium on applied computing (5: 16-20 Marzo: Fortaleza, Ceara, Brazil). ACM. New York, NY, USA. 2008. p. 1995-1999.
[56] VARGAS, Blanca. Op. Cit.
[57] TSO-SUTTER, Karen, *et al*. Op. cit.

La matriz de etiquetas de productos incluye características de los restaurantes disponibles, como la ubicación o especialidad de la casa. La matriz de etiquetas de usuarios almacena información demográfica y geográfica de las personas. Por lo tanto, las recomendaciones se realizan mediante el análisis semántico de las características del usuario con las de los restaurantes. Por ejemplo, si un usuario expresa que le gusta fumar, este usuario será relacionado con los restaurantes que permitan fumar dentro del establecimiento.

Una desventaja del método actual es su carácter intrusivo. Es decir, cuando un usuario nuevo se registra en la BD del sistema, debe llenar un formulario que solicita 15 características que indagan sobre su personalidad. Así mismo, cada restaurante se caracteriza con 23 atributos empleados para construir el perfil del restaurante. Adicionalmente, es necesario que los usuarios proporcionen una calificación después de haber visitado el restaurante. Por lo anterior, es deseable una metodología que permita calcular los puntajes restantes sin contar con información adicional a la contenida en la matriz U-P. El objetivo del proyecto entonces se delimita a modelar, implementar y evaluar un algoritmo de predicción para el SR SURFEOUS usando el enfoque de filtrado colaborativo basado en modelos, particularmente bajo el enfoque de reducción dimensional y descomposición de matrices.

El algoritmo planteado debe encontrar una representación de menor dimensión de la matriz U-P. Esta representación debe reflejar los factores latentes que determinan las decisiones de los usuarios cuando eligen un restaurante. De esta manera, las predicciones se calculan considerando únicamente el modelo latente del sistema.

4.6.CARACTERIZACIÓN DEL PROBLEMA

La Tabla 6 presenta un resumen de los parámetros que caracterizan el problema específico planteado para la evaluación del algoritmo propuesto en éste proyecto.

Tabla 6 Parámetros del problema

Dominio del problema	SR de restaurantes
Dimensiones	138 usuarios $\times$ 130 productos
Datos de entrada	Explícitos - 93.5 % desconocido
Datos de Salida	Recomendaciones top-5
Enfoque del problema	Filtrado colaborativo basado en modelos
Algoritmo de predicción	Algoritmo propuesto DeMBaR basado en la teoría de completar matrices y en solucionadores de problemas de optimización convexa.
Métricas de evaluación	Precisión, exhaustividad, error absoluto medio, y tiempo de cómputo

5. MODELAMIENTO MATEMÁTICO DEL PROBLEMA

5.1. ANÁLISIS DEL PROBLEMA PARTICULAR

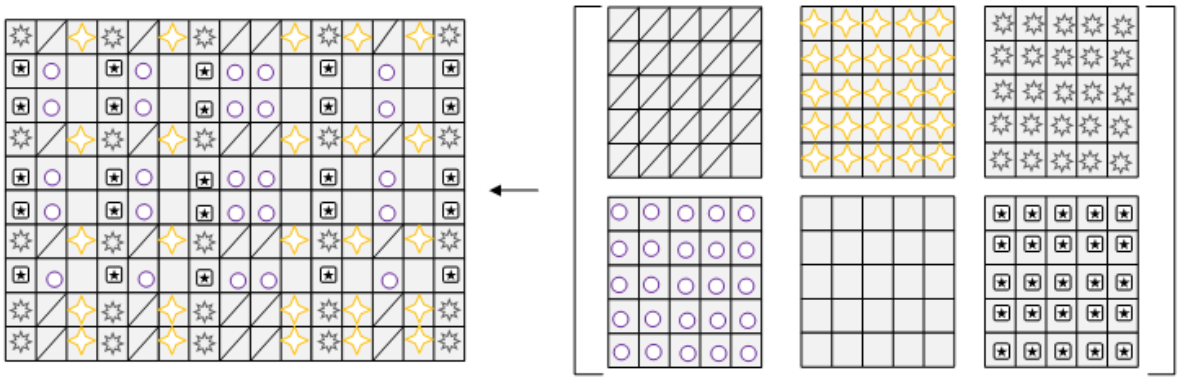
Suponiendo que las decisiones de los usuarios dependen de unos pocos factores latentes, el problema del SR SURFEOUS puede ser reformulado como un problema de completar matrices, lo que evita emplear información adicional del usuario y/o de los lugares disponibles a ser recomendados. Bajo este enfoque se debe completar la matriz de interacción U-P a partir de los pocos puntajes conocidos. Usualmente, se considera que la matriz en su totalidad es una matriz de bajo rango. Sin embargo, en este proyecto se planteó un enfoque que asume que la matriz U-P está compuesta por varias matrices de bajo rango.

El planteamiento supone que el conjunto de usuarios y/o productos se puede separar en subconjuntos o categorías diferentes. Por ejemplo, dentro del conjunto de usuarios se pueden diferenciar sub-conjuntos de acuerdo a la edad o el género. Así mismo, los restaurantes se clasifican en diferentes categorías de acuerdo a sus atributos como tipo de cocina u horarios de atención. La combinación de las categorías encontradas de usuarios y productos genera un conjunto de sub-matrices de interacción U-P, las cuales tienen factores latentes diferenciados. Por lo tanto, la estimación de la matriz U-P original se hace minimizando el rango de un conjunto de sub-matrices determinadas por las combinaciones de los subconjuntos de usuarios y/o productos.

La Figura 9 ilustra el concepto de descomposición en varias matrices de bajo rango analizado en este proyecto. Observe como la matriz de interacción U-P se descompone en 6 sub-matrices diferentes. Esta descomposición está determinada por una partición del conjunto de usuarios y una partición del conjunto de productos. En el contexto del problema, estas particiones pueden ser analizadas como la clasificación de los usuarios y productos en diversas categorías. Adicionalmente,

observe que la matriz original puede ser reordenada de ésta manera sin pérdida de estructura. Puesto que el índice original está determinado por el tiempo de registro del usuario o producto y no porque mantenga alguna relación con sus vecinos.

Figura 9 Descomposición en matrices de bajo rango



Matriz de interacción U-P

Matrices de bajo rango que componen la matriz de interacción U-P

5.2.Formulación matemática

Sea $\mathrm{C} = \{c_1, c_2, \dots, c_M\}$ un conjunto de M usuarios y $\mathrm{P} = \{p_1, p_2, \dots, p_N\}$ un conjunto de N productos; un SR debe recomendar a cada usuario $c_i \in \mathrm{C}$ una lista de sus n productos preferidos $p_1, \dots, p_n \in \mathrm{P}$, donde n está determinado por el tamaño de la lista recomendación, generalmente. $n \ll N$**.** Sea $\mathbf{R} \in \mathbb{R}^{M \times N}$ una matriz que relaciona a los usuarios en C con los productos en P, la relación r_{ij} es el puntaje asignado por el usuario $c_i \in \mathrm{C}$ al producto $p_j \in \mathrm{P}$. Idealmente, se desea conocer todas las relaciones presentes en $\mathbf{R}$ para realizar las recomendaciones. Sin embargo, $\mathbf{R}$ comúnmente es una matriz desconocida.

Sea $\mathbf{M} \in \mathbb{R}^{M \times N}$ una matriz que contiene $m \ll (M \times N)$ relaciones conocidas y sea $\Omega = \{(i,j) \in \mathbb{N}^2\}$ el conjunto de posiciones (i,j) donde el valor m_{ij} es conocido, la matriz $\mathbf{M}$ se emplea para calcular $\widehat{\mathbf{R}} \in \mathbb{R}^{M \times N}$ la cual representa la estimación de la

matriz **R.** Este planteamiento se puede describir como un proceso de predicción en función de la matriz **M** como se observa en (12).

$$\widehat{\mathbf{R}} \leftarrow \text{estimación}(\mathbf{M}), \text{donde } m_{ij}, (i,j) \in \Omega \text{ es conocido} \tag{12}$$

Como se describió en el análisis del problema, en este planteamiento se supone que las decisiones de los usuarios dependen de un pequeño conjunto de factores latentes, es decir, la matriz **R** se caracteriza por ser de bajo rango, donde $r = \text{rank}(\mathbf{R}) \ll \min(M, N)$. Por lo tanto, la estimación $\widehat{\mathbf{R}}$ se encuentra empleando la teoría de completar matrices*. Esta teoría indica que una matriz de bajo rango puede ser completada bajo dos condiciones. En primer lugar, la cantidad de valores conocidos m debe ser lo suficientemente grande, y en segundo lugar, el conjunto de posiciones conocidas Ω debe ser uniformemente distribuido de tal forma que cada usuario $c_i \in \mathrm{C}$ haya calificado al menos un producto y cada producto $p_j \in \mathrm{P}$ haya sido calificado al menos una vez[58].

Sea H una partición de C con K elementos, $\mathrm{H} = \{\mathrm{C}_k : k = 1, \dots, K\}$, y F una partición de P con L elementos, $\mathrm{F} = \{\mathrm{P}_\ell : \ell = 1, \dots, L\}$,. La matriz **R** se descompone en KL sub-matrices de la forma $\mathbf{R}_{k+(\ell-1)K} = (r_{ij})$ para las parejas ordenadas $(i,j) \in \mathrm{C}_k \times \mathrm{P}_\ell$ con $\mathrm{C}_k \in \mathrm{H}$ y $\mathrm{P}_\ell \in \mathrm{F}$ para $k = 1,2, \dots, K$ y $\ell = 1,2, \dots, L$. Las KL sub-matrices componen la matriz original **R** como se muestra a continuación:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \cdots & \mathbf{R}_{1+(L-1)K} \\ \vdots & \ddots & \vdots \\ \mathbf{R}_K & \cdots & \mathbf{R}_{KL} \end{bmatrix}. \tag{13}$$

De manera equivalente, la matriz **M** se descompone en KL sub-matrices de la forma:

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \cdots & \mathbf{M}_{1+(L-1)K} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_K & \cdots & \mathbf{M}_{KL} \end{bmatrix}, \tag{14}$$

* Ver numeral 3.7. Teoría de completar matrices de bajo rango

[58] CANDES, Enmanuel, RECHT, Benjamin. Op. Cit.

donde, $\mathbf{M}_{k+(\ell-1)K} = (m_{ij})$ para las parejas ordenadas $(i,j) \in \mathrm{C}_k \times \mathrm{P}_\ell$ con $\mathrm{C}_k \in \mathrm{H}$ y $\mathrm{P}_\ell \in \mathrm{F}$, $k = 1,2,\ldots,K$ y $\ell = 1,2,\ldots,L$.

El conjunto de sub-matrices $\mathbf{R}_i$ en (13) se suponen todas de bajo rango, así, las estimaciones $\widehat{\mathbf{R}}_i$ de $\mathbf{R}_i$ se calculan resolviendo el problema de completar matrices sobre el conjunto de sub-matrices $\mathbf{M}_i$ en (14). De esta manera la estimación $\widehat{\mathbf{R}}$ se obtiene como sigue estimando de manera independiente cada matriz $\widehat{\mathbf{R}}_i$ como se observa en (15).

$$\widehat{\mathbf{R}} = \begin{bmatrix} \widehat{\mathbf{R}}_1 & \cdots & \widehat{\mathbf{R}}_{1+(L-1)K} \\ \vdots & \ddots & \vdots \\ \widehat{\mathbf{R}}_K & \cdots & \widehat{\mathbf{R}}_{KL} \end{bmatrix}, \tag{15}$$

6. DISEÑO DEL ALGORITMO DE SOLUCIÓN

El algoritmo de solución propuesto se denomina DeMBaR (Descomposición en Matrices de Bajo Rango). A continuación se describe de manera detallada los pasos realizados por el algoritmo DeMBaR para resolver el problema de recomendación de restaurantes según el contexto del problema del SR SOURFEOUS.

6.1.PARÁMETROS DE ENTRADA

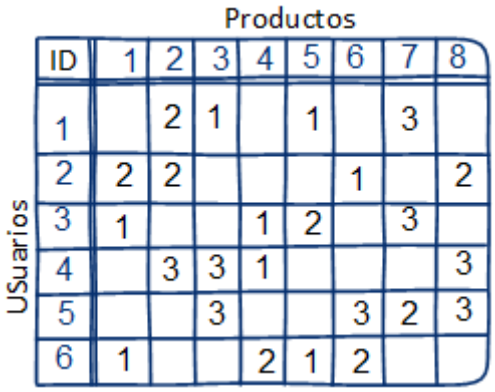
Los datos de entrada para el algoritmo son:

- $\mathrm{C} = \{c_1, c_2, \dots, c_M\}$ el conjunto de M usuarios.
- $\mathrm{P} = \{p_1, p_2, \dots, p_N\}$ el conjunto de N productos.
- $\mathbf{M} \in \mathbb{R}^{M \times N}$ con m puntajes conocidos en las posiciones $(i, j) \in \Omega$.
- K: cantidad de particiones del conjunto de usuarios.
- L: cantidad de particiones del conjunto de productos.
- $OptSub$: Este parámetro corresponde a la metodología para determinar las K particiones de usuarios y las L particiones de productos. Las tres alternativas empleadas fueron, selección aleatoria, selección basada en la correlación de Pearson y selección basada en la descomposición en valores singulares (SVD).
- $OptSol$: Este parámetro corresponde al solucionador elegido para completar las KL sub-matrices de bajo rango. Los dos solucionadores evaluados son el FPC y LmaFit.
- n: Parámetro que determina la cantidad de productos a recomendar en la lista top$-n$ de recomendación.

Como ejemplo ilustrativo a continuación se muestra la metodología de solución en el proceso de recomendación cuando se tiene un conjunto de 6 usuarios sobre un conjunto de 8 productos.

La Figura 10 muestra el ejemplo de la matriz de valores conocidos $\mathbf{M} \in \mathbb{R}^{6\times 8}$**,** en la cual, la primera columna corresponde al identificador del usuario y la primera fila corresponde al identificador de los productos. Observe que la matriz **M** tiene en este ejemplo un total de $6 * 8 = 48$ posiciones, de las cuales se conocen únicamente 24, es decir, el algoritmo debe predecir el 50% de datos faltantes.

Figura 10 Ejemplo de matriz de puntajes conocidos



El primer paso que debe realizar el algoritmo consiste en dividir el conjunto de usuarios y productos en diferentes subconjuntos y determinar así la partición H de usuarios y la partición F de productos.

6.2.IDENTIFICAR PARTICIONES DE USUARIOS Y PRODUCTOS

En la Figura 11 se muestra un ejemplo de identificación de los usuarios y productos que pertenecen a cada partición cuando se tienen como parámetros de entrada $K = 2$ y $L = 2$. Observe que estos parámetros indican que tanto la partición H como la partición F deben tener dos elementos. En la figura 13 se ilustra como el conjunto de usuarios C de 6 elementos se debe descomponer en dos subconjuntos C_1y C_2 de tal forma que estos conformen la partición H de C, es decir, cada subconjunto de la partición H contiene $\frac{6}{2} = 3$ elementos. Así mismo el conjunto de productos P con 8

elementos se debe descomponer en dos subconjuntos P_1y P_2 de tal forma que estos sean una partición F de P, es decir, cada subconjunto de la partición F contiene $\frac{8}{2} = 4$ elementos. En el ejemplo ilustrado los usuarios y productos fueron ubicados de la siguiente manera: $\mathrm{C}_1 = \{6,3,5\}$, $\mathrm{C}_2 = \{1,2,4\}$, $\mathrm{P}_1 = \{5,1,8,2\}$ y $\mathrm{P}_2 = \{7,3,4,6\}$.

Figura 11 Ejemplo de selección de subconjuntos

$\mathrm{C} = \{1,2,3,4,5,6\}$	$\mathrm{P} = \{1,2,3,4,5,6,7,8\}$
$\mathrm{H} = \{\mathrm{C}_1 , \mathrm{C}_2\}$	$\mathrm{F} = \{\mathrm{P}_1 , \mathrm{P}_2\}$
$\mathrm{H} = \{\{6,3,5\}, \{1,2,4\}\}$	$\mathrm{F} = \{\{5,1,8,2\}, \{7,3,4,6\}\}$

El método para determinar la partición H de K subconjuntos del conjunto de usuarios y la partición F de L subconjuntos del conjunto de productos se define en el parámetro de entrada $OptSub$; este parámetro toma una de tres alternativas: selección aleatoria, selección por correlación de Pearson y selección de acuerdo a la descomposición en valores singulares SVD. A continuación, se describe el proceso realizado según cada alternativa planteada:

6.2.1 Selección aleatoria Esta opción consiste en elegir aleatoriamente una cantidad de $\lceil M/K \rceil$ usuarios $c_i \in \mathrm{C}$ y ubicarlos en una categoría dentro de los K elementos de la partición H del conjunto de usuarios siguiendo una distribución de probabilidad uniforme. De forma equivalente, cada una de las L categorías de productos que componen la partición F se conforma mediante la elección aleatoria de $\lceil N/L \rceil$ productos $p_i \in \mathrm{P}$ siguiendo una distribución de probabilidad uniforme.

6.2.2 Selección por correlación de Pearson En esta opción se usa la métrica de correlación de Pearson para definir los usuarios y productos que pertenecen a cada elemento de las particiones H y L. Para esto, se debe calcular la matriz de

coeficientes de correlación de Pearson a partir de la matriz de valores conocidos $\mathbf{M}$. En el caso de la partición H del conjunto de usuarios, se debe calcular la matriz de correlación entre usuarios, es decir entre filas de la matriz original para obtener la matriz $\mathbf{P}_C \in \mathbb{R}^{M\times M}$, la cual es una matriz simétrica donde $p_{C_{ij}}$ corresponde a la correlación entre el usuario i y el usuario j que pertenecen al conjunto de usuarios C. Una vez calculada la matriz de correlación, cada elemento de la partición se conforma eligiendo los $\lceil M/K \rceil$ usuarios que tienen mayor correlación.

De manera similar, la partición F del conjunto de usuarios se determina a partir de la matriz de coeficientes de correlación de Pearson entre productos, es decir entre las columnas de la matriz original para obtener la matriz $\mathbf{P}_P \in \mathbb{R}^{N\times N}$, la cual es una matriz simétrica donde $p_{p_{ij}}$ corresponde a la correlación entre el producto i y el producto j que pertenecen al conjunto de productos P. Una vez calculada la matriz de correlación cada categoría o elemento de la partición se conforma de los $\lceil N/L \rceil$ productos que tienen mayor correlación.

6.2.3 Selección con descomposición SVD Esta tercera alternativa los usuarios y productos que pertenecen a cada elemento de las particiones se eligen considerando la correlación entre usuarios y productos en un dominio diferente al original. El primer paso consiste en calcular una estimación preliminar de la matriz $\widehat{\mathbf{R}}$ usando un método de completar matrices a partir de la matriz de valores conocidos $\mathbf{M}$. Seguidamente la matriz estimada $\widehat{\mathbf{R}}$ se factoriza según la descomposición en valores singulares $\widehat{\mathbf{R}} = \widehat{\mathbf{U}}\widehat{\mathbf{S}}\widehat{\mathbf{V}}$, donde $\widehat{\mathbf{U}} \in \mathbb{R}^{M\times r}$, $\widehat{\mathbf{S}} \in \mathbb{R}^{r\times r}$ y $\widehat{\mathbf{V}} \in \mathbb{R}^{N\times r}$. La matriz $\widehat{\mathbf{U}}$ representa la relación de los usuarios con r factores latentes. La matriz $\widehat{\mathbf{V}}$ representa la relación de los productos con los r factores latentes y la matriz $\widehat{\mathbf{S}}$ contiene una ponderación indicando cuáles son los factores más importantes.

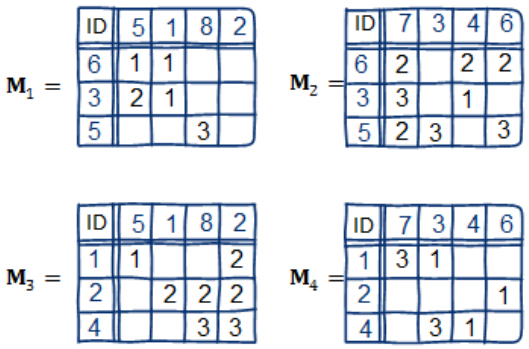
Finalmente, los usuarios y productos de cada elemento en las particiones se elige aplicando la metodología de selección por correlación Pearson, pero a diferencia de

lo descrito en la sección 6.2.2., las matrices de correlación se calculan a partir de la matriz $\widehat{\mathbf{U}}$ para los usuarios y a partir de la matriz $\widehat{\mathbf{V}}$ para los productos.

6.3.DETERMINAR LAS SUB-MATRICES DE BAJO RANGO

Una vez se tienen definidos los subconjuntos en cada partición, el siguiente paso corresponde a reordenar e identificar las sub-matrices que se generan a partir de la combinación de los subconjuntos definidos. Observe que los índices (i, j) correspondientes a cada sub-matriz $\mathbf{M}_{k+(\ell-1)K} = (m_{ij})$ se obtienen realizando el producto cruz $\mathrm{C}_k \times \mathrm{P}_\ell$ entre cada $\mathrm{C}_k \in \mathrm{H}$ con cada $\mathrm{P}_\ell \in \mathrm{F}$ para $k = 1,2,\dots,K$ y $\ell = 1,2,\dots,L$. La Figura 12 muestra el ejemplo de obtener 4 sub-matrices a partir de las particiones H y F obtenidas anteriormente e ilustradas en la figura 13.

Figura 12 Ejemplo de obtención de sub-matrices



Observe en la Figura 12 que la matriz $\mathbf{M}_1 = (m_{ij})$ contiene los valores determinados por los índices (i, j) obtenidos mediante el producto cruz $\mathrm{C}_1 \times \mathrm{P}_1 = \{6,3,5\} \times \{5,1,8,2\}$, donde $\mathrm{C}_1 \in \mathrm{H}$ y $\mathrm{P}_1 \in \mathrm{F}$. Es decir, el conjunto de índices para $\mathbf{M}_1 = (i, j) = \{(6,5), (6,1), (6,8), (6,2), (3,5), (3,1), (3,8), (3,2), (5,5), (5,1), (5,8), (5,2)\}$. Estos índices almacenan los valores {1, 1, ∅, ∅, 2, 1, ∅, ∅, ∅, ∅, 3, ∅}.

6.4.VERIFICAR CONDICIONES TEORÍA DE COMPLETAR MATRICES

Una vez se han obtenido las sub-matrices, el algoritmo debe verificar que cada sub-matriz cumpla las condiciones de la teoría de completar matrices. Por lo tanto, en este paso se verifica que cada sub-matriz $\mathbf{M}_i$ contenga al menos un valor conocido en cada fila y en cada columna. Si se encuentra que alguna matriz no cumple la condición, se realiza un intercambio de un elemento de un subconjunto $\mathrm{C}_k \in \mathrm{H}$ o de un elemento de un subconjunto $\mathrm{P}_\ell \in \mathrm{F}$ según corresponda.

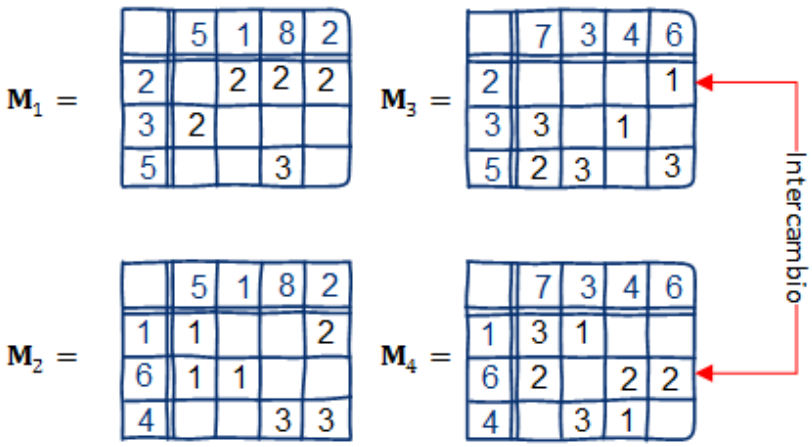
En el ejemplo mostrado en la Figura 12 se observa que la sub-matriz $\mathbf{M}_1$ tiene una columna identificada con la etiqueta del producto ID=2 completamente desconocida. Para solucionar este problema se realiza un paso de intercambio. El paso de intercambio selecciona aleatoriamente un elemento del subconjunto C_1 y lo reemplaza con un elemento de cualquier otro subconjunto $\mathrm{C}_i \in \mathrm{H}, i \neq 1$.

En el ejemplo ilustrativo se tienen 2 subconjuntos, por lo tanto la única opción es realizar el intercambio con un elemento del subconjunto C_2. El elemento seleccionado para realizar el intercambio puede ser cualquiera, elegido aleatoriamente, con la condición de que solucione el problema presentado en M_1. Es decir, en este ejemplo el usuario que será intercambiado debe haber calificado el producto con ID=2 para así solucionar el problema. Como se observa en la matriz original U-P en la Figura 10, hay tres usuarios que cumplen esta condición: 1,2 y 4.

En la Figura 13 se muestran las sub-matrices obtenidas después de realizar el paso de intercambio. Las etiquetas intercambiadas corresponden al usuario 6 que pertenece a C_1 con el usuario 2 que pertenece a C_2. De esta manera, los nuevos índices para la matriz $\mathbf{M}_1 = (m_{ij})$ corresponden a las parejas ordenadas (i, j) obtenidos mediante el producto cruz $\mathrm{C}_1 \times \mathrm{P}_1 = \{2,3,5\} \times \{5,1,8,2\}$. Esto es, $(i, j) = \{(2,5), (2,1), (2,8), (2,2), (3,5), (3,1), (3,8), (3,2), (5,5), (5,1), (5,8), (5,2)\}$. Observe que ahora ninguna sub-matriz tiene columnas o filas completamente desconocidas.

Si el problema presentado es una fila completamente desconocida, el intercambio se realiza entre elementos de los subconjuntos de la partición F, es decir, no se hace intercambio entre usuarios sino entre productos.

Figura 13 Ejemplo del paso de intercambio



Después de realizar el paso de intercambio entre elementos de los subconjuntos de la partición H, se recalculan las sub-matrices $\mathbf{M}_i$.

6.5.MINIMIZACIÓN DEL RANGO DE CADA SUB-MATRIZ

En este paso se realiza la estimación $\widehat{\mathbf{R}}_i, i = 1, .., KL$ a partir de la sub-matriz correspondiente $\mathbf{M}_i, i = 1, .., KL$. El proceso de estimación se realiza empleando el solucionador especificado en el parámetro $OptSol$. Los solucionadores se eligieron teniendo en cuenta la revisión de la literatura[59] y considerando los resultados de un trabajo de investigación previo* empleando la teoría de completar matrices en la recuperación de imágenes[60]. El parámetro $OptSol$ puede tomar los valores:

[59] MICHENKOVÁ. Op. Cit.

* El algoritmo LMaFit fue inicialmente evaluado en el dominio de reconstrucción de imágenes multi-espectrales. Los resultados fueron presentados en la conferencia STSIVA, 2015

[60] GELVEZ, Tatiana, RUEDA, Hoover y ARGUELLO, Henry. Coded aperture design for hyper-spectral image recovery via Matrix Completion. En: XX Simposio de Tratamiento de Señales, Imágenes y Visión Artificial (STSIVA) (3:2-4 Septiembre: Bogotá, DC, COLOMBIA). Pontificia Universidad Javeriana. 2015, p.1-7.

1. **Algoritmo FPC:** Algoritmo iterativo que emplea el enfoque de minimización de la norma nuclear para resolver el problema. Ver Anexo A.
2. **Algoritmo LMaFit:** Algoritmo que resuelve la versión Lagragiana del problema. Ver Anexo B.

La Figura 14 muestra un ejemplo de las estimaciones obtenidas $\widehat{\mathbf{R}}_i$ resolviendo el problema de completar matrices a partir de cada $\mathbf{M}_i$ definida en el punto anterior.

Figura 14 Ejemplo de las sub-matrices estimadas

$\widehat{\mathbf{R}}_1 =$

ID	5	1	8	2
2	3	2	2	2
3	2	1	1	3
5	1	2	3	3

$\widehat{\mathbf{R}}_3 =$

ID	7	3	4	6
2	3	2	1	1
3	3	1	1	2
5	2	3	3	3

$\widehat{\mathbf{R}}_2 =$

ID	5	1	8	2
1	1	1	2	2
6	1	1	1	1
4	3	2	3	3

$\widehat{\mathbf{R}}_4 =$

ID	7	3	4	6
1	3	1	3	3
6	2	3	2	2
4	1	3	1	1

Cada matriz $\widehat{\mathbf{R}}_i$ se obtiene resolviendo el problema de minimización del rango para completar matrices usando un algoritmo de completar matrices.

6.6.OBTENCIÓN DE LA MATRIZ DE INTERACCIÓN

El siguiente paso consiste en combinar los resultados de la estimación de cada una de las sub-matrices para obtener la estimación de la matriz original. La Figura 15 muestra el ejemplo de obtener la matriz $\widehat{\mathbf{R}}$ integrando los resultados obtenidos en el punto anterior.

Figura 15 Ejemplo de la matriz de interacción estimada

$\hat{\mathbf{R}} =$

ID	5	1	8	2	7	3	4	6
2	3	2	2	2	3	2	1	1
3	2	1	1	3	3	1	1	2
5	1	2	3	3	2	3	3	3
1	1	1	2	2	3	1	3	3
6	1	1	1	1	2	3	2	2
4	3	2	3	3	1	3	1	1

La matriz estimada $\hat{\mathbf{R}}$ integra las estimaciones individuales de cada sub-matriz $\hat{\mathbf{R}}_i$ conservando la estructura según los índices de cada sub-matriz.

6.7.GENERAR LISTA DE RECOMENDACIÓN TOP-n.

Finalmente las recomendaciones se proporcionan en un formato de lista top-n. Para cada usuario $c_i \in \mathrm{C}$, se eligen n elementos determinados por los n puntajes más altos a lo largo de los valores $\hat{\mathbf{r}}_i = [\hat{r}_{i1}, \hat{r}_{i2}, \ldots, \hat{r}_{iN}]$. En la Figura 16 se muestra el ejemplo de la lista top-2 para el usuario c_2.

Figura 16 Ejemplo de lista top-n

Puntajes más altos

$\hat{\mathbf{r}}_2 =$

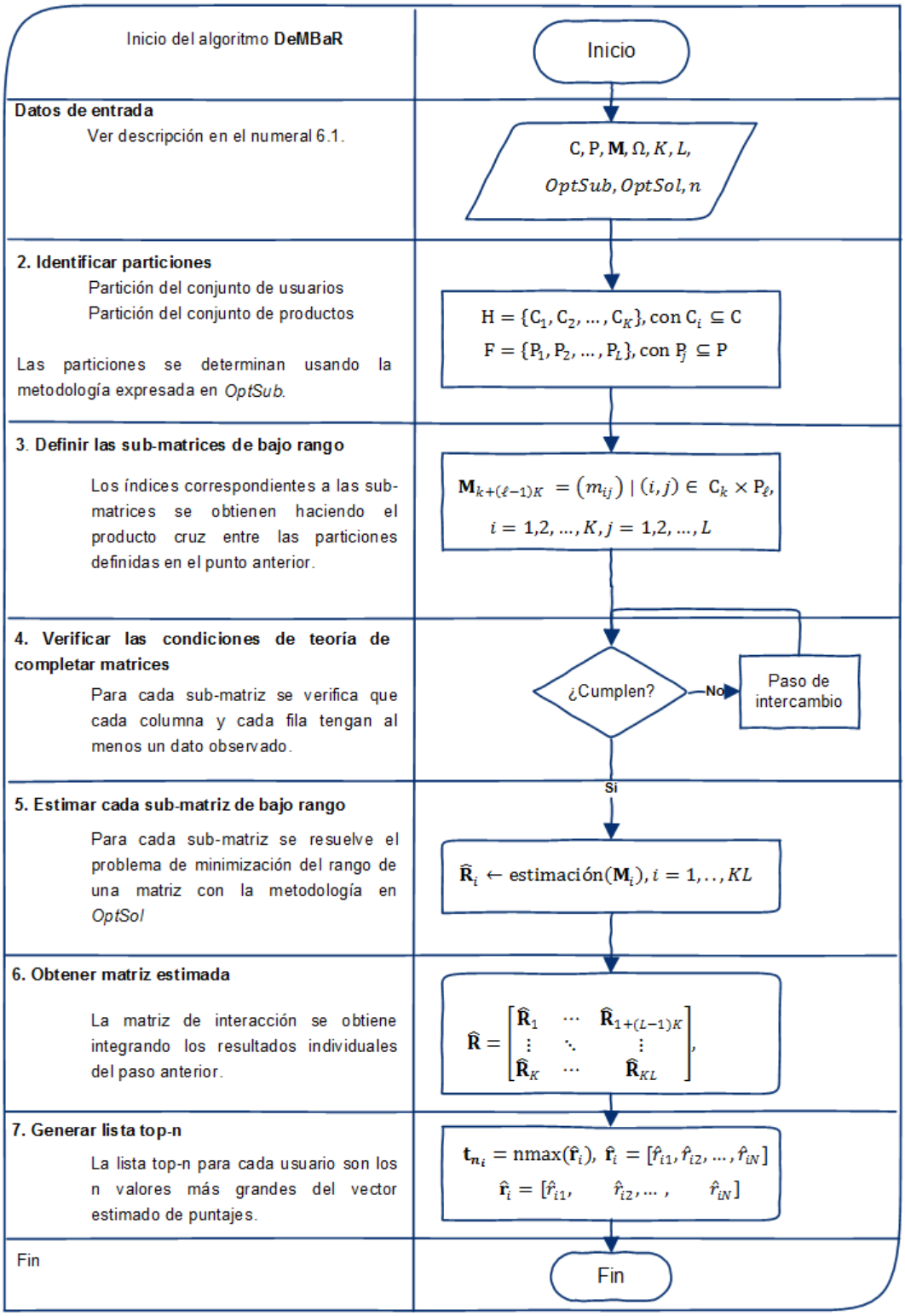
	5	1	8	2	7	3	4	6
2	3	2	2	2	3	2	1	1

Lista top-2 = | 5 | 7 |

A partir de las estimaciones contenidas en $\hat{\mathbf{r}}_2$ se eligen los 2 productos $(p_5\ y\ p_7)$ con los puntajes más altos. En este caso se recomiendan al usuario 2 los productos $p_5\ y\ p_7$.

Para facilitar el entendimiento del algoritmo propuesto, el Diagrama 1 muestra la secuencia y explica de manera general los pasos detallados anteriormente.

Diagrama 1 Diagrama de flujo del algoritmo DeMBaR



6.8.PSEUDOCÓDIGO DEL ALGORITMO DEMBAR

Con fines de implementación, a continuación se presenta el pseudocódigo del algoritmo propuesto DeMBaR.

Algoritmo 1 DeMBaR (Descomposición en Matrices de Bajo Rango)

— Entrada: $\mathbf{C}, \mathbf{P}, \mathbf{M}, \boldsymbol{\Omega}, \boldsymbol{K}, \boldsymbol{L}, \boldsymbol{OptSub}, \boldsymbol{OptSol}, \boldsymbol{n}$

— $[\mathbf{H}, \mathbf{F}] \leftarrow \wp(\mathbf{C}, \boldsymbol{K}, \mathbf{P}, \boldsymbol{L}, \boldsymbol{OptSub})$ Determinar las particiones $\mathbf{H}$ y $\mathbf{F}$.

— **Para** $\boldsymbol{k} = \mathbf{1}, \dots, \boldsymbol{K}$ **hacer**

  — **Para** $\boldsymbol{\ell} = \mathbf{1}, \dots, \boldsymbol{L}$ **hacer**

  — $\mathbf{M}_{\boldsymbol{k}+(\boldsymbol{\ell}-\mathbf{1})\boldsymbol{K}} \leftarrow \left(\boldsymbol{m}_{\boldsymbol{ij}}\right) \mid (\boldsymbol{i}, \boldsymbol{j}) \in \mathbf{C}_{\boldsymbol{k}} \times \mathbf{P}_{\boldsymbol{\ell}}, \boldsymbol{m}_{\boldsymbol{ij}} = \mathbf{0}\ \boldsymbol{si}\ (\boldsymbol{i}, \boldsymbol{j}) \notin \boldsymbol{\Omega}$

  — **Fin Para**

— **Fin Para**

— **Mientras** NO CUMPLA condiciones de teoría de completar matrices **hacer**

  — $\left[\mathbf{M}_{\boldsymbol{k}+(\boldsymbol{\ell}-\mathbf{1})\boldsymbol{K}}, \mathbf{H}, \mathbf{F}\right] \leftarrow \mathbf{intercambio}\left(\mathbf{M}_{\boldsymbol{k}+(\boldsymbol{\ell}-\mathbf{1})\boldsymbol{K}}, \mathbf{H}, \mathbf{F}\right), \forall_{\boldsymbol{k}}, \forall_{\boldsymbol{\ell}}$

— **Fin mientras**

— **Para** $\boldsymbol{p} = \mathbf{1}, \mathbf{2}, \dots, \boldsymbol{KL}$ **hacer**

  — $\widehat{\mathbf{R}}_{\boldsymbol{p}} = \mathbf{argmin}\ \mathbf{rank}\left(\widehat{\mathbf{R}}_{\boldsymbol{p}}\right)\ \mathbf{sujeto\ a}\ \hat{\boldsymbol{r}}_{\boldsymbol{p}_{ij}} = \boldsymbol{m}_{\boldsymbol{p}_{ij}}, (\boldsymbol{i}, \boldsymbol{j}) \in \mathbf{C}_{\boldsymbol{k}} \times \mathbf{P}_{\boldsymbol{\ell}}$ con método $\boldsymbol{OptSol}$.

— **Fin para**

— $\widehat{\mathbf{R}} \leftarrow \begin{bmatrix} \widehat{\mathbf{R}}_{\mathbf{1}} & \cdots & \widehat{\mathbf{R}}_{\mathbf{1}+(\boldsymbol{L}-\mathbf{1})\boldsymbol{K}} \\ \vdots & \ddots & \vdots \\ \widehat{\mathbf{R}}_{\boldsymbol{K}} & \cdots & \widehat{\mathbf{R}}_{\boldsymbol{KL}} \end{bmatrix}$

— **Para** $\boldsymbol{i} = \mathbf{1}, \mathbf{2}, \dots, \boldsymbol{M}$ **hacer**

  — Calcular $\mathbf{t}_{\boldsymbol{n_i}} = \mathbf{nmax}(\hat{\mathbf{r}}_{\boldsymbol{i}}),\ \hat{\mathbf{r}}_{\boldsymbol{i}} = [\hat{\boldsymbol{r}}_{\boldsymbol{i1}}, \hat{\boldsymbol{r}}_{\boldsymbol{i2}}, \dots, \hat{\boldsymbol{r}}_{\boldsymbol{iN}}]$

— **Fin para**

— Salida: $\mathbf{t}_{\boldsymbol{n_i}}, \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{M}$

---

7. EVALUACIÓN Y EXPERIMENTOS REALIZADOS

Para verificar el rendimiento del algoritmo DeMBaR se realizaron diversas simulaciones sobre la BD "*Restaurant & Consumer*". Esta base de datos contiene 1161 puntajes conocidos de 138 usuarios sobre 130 productos. Esto corresponde al 6,4716 % del total de puntajes posibles.

7.1.MÉTRICAS DE EVALUACIÓN

El rendimiento del algoritmo se evaluó bajo tres diferentes enfoques: exactitud, rendimiento computacional y soporte de decisión. Los resultados de las métricas de soporte de decisión fueron comparados con los presentados en el trabajo realizado por VARGAS, Blanca, *et al.*[61], donde el algoritmo principal se basa en leyes semánticas y emplea información contextual para determinar las predicciones. En la Tabla 7 se presentan las métricas computadas para evaluar cada uno de los enfoques.

Tabla 7 Enfoques de evaluación

Enfoque	Métrica	Objetivo
Exactitud	Error Absoluto Medio Ver sección 3.9.1.	Evaluar en términos del error numérico las estimaciones calculadas.
Rendimiento computacional	Tiempo de cómputo.	Evaluar en términos de eficiencia computacional el algoritmo propuestos
Soporte de decisión	Precisión y exhaustividad. Ver sección 3.9.2.	Evaluar qué tan útil es la herramienta para orientar a los usuarios a tomar buenas decisiones.

[61] VARGAS, Blanca. Op. Cit.

7.2. ESQUEMA DE VALIDACIÓN

El esquema empleado para validar estadísticamente los resultados corresponde a la técnica de validación cruzada. Ésta técnica permite validar la independencia de los resultados obtenidos respecto a la partición entre datos de entrenamiento y datos de prueba[62]. La técnica fue elegida teniendo en cuenta que los resultados de comparación fueron obtenidos empleando este mismo enfoque[63].

Específicamente, se realizó validación cruzada dejando uno fuera (del inglés *leave-one out cross validation*). Bajo esta metodología, la partición entre datos de entrenamiento y datos de prueba se hace dejando $n - 1$ datos de entrenamiento y solo 1 de prueba. De esta manera, los datos de entrenamiento se definen extrayendo un puntaje por cada usuario del sistema. Es decir, se eliminaron 138 puntajes de los 1161 conocidos. Lo anterior corresponde a $11{,}89\ \%$ datos de prueba y $88{,}11\ \%$ datos de entrenamiento. El puntaje eliminado para cada usuario fue elegido aleatoriamente siguiendo una distribución de probabilidad uniforme.

La técnica de validación cruzada indica que el resultado de una réplica se obtiene calculando la media aritmética de los resultados obtenidos para cada instancia. De esta manera, los resultados de una réplica es la media aritmética del resultado obtenido para cada uno de los 138 usuarios. Así mismo, para cada experimento se realizaron 100 réplicas empleando una partición entre datos de entrenamiento y datos de prueba diferente. Estas particiones fueron generadas inicialmente y se emplearon para todos los experimentos. Por lo anterior, los resultados que se muestran corresponden a la media aritmética de las 100 réplicas.

---

[62] GUTIÉRREZ, Ricardo. Intelligent Sensor System. Leave-one-out Cross Validation. Wright State University. [En línea].
[63] VARGAS, Blanca. Op. Cit.

7.3. DESCRIPCIÓN DE EXPERIMENTOS

Los experimentos realizados en este proyecto tienen como objetivo analizar el rendimiento de algoritmo DeMBaR variando los parámetros que podrían afectar su desempeño. Estos parámetros corresponden a:

1. $OptSub$: Éste parámetro corresponde al método empleado para encontrar las partición de usuarios H y la partición de productos F. Este parámetro tiene tres alternativas de selección, de manera aleatoria, basada en la correlación de Pearson y basado en la descomposición SVD. Ver sección 6.2.

2. $OptSol$: Éste parámetro corresponde al solucionador empleado para resolver la minimización del rango de una matriz para cada una de las sub-matrices $\widehat{\mathbf{R}}_i$. Éste parámetro tiene dos posibilidades, el algoritmo FPC y el algoritmo LMaFit. Ver sección 6.5.

3. L: Cantidad de elementos en la partición F.

El parámetro L hace referencia a la cantidad de subconjuntos en la que se divide el conjunto original de productos. La definición de los niveles de éste parámetro se obtuvo considerando las limitaciones de la teoría de completar matrices.

La teoría de completar matrices dicta que para reconstruir una matriz cada fila y cada columna debe tener al menos un valor conocido. Observe que al dividir el conjunto de productos, la probabilidad de que una sub-matriz quede con filas completamente desconocidas es superior. Suponga un caso en el que un usuario solo ha calificado 3 productos y además se desea realizar una partición de 4 subconjuntos, en este ejemplo de ninguna manera será posible satisfacer las condiciones de completar matrices puesto que sólo existen tres elementos y son necesarios al menos 4 valores conocidos. Por lo tanto, la cantidad de subconjuntos

L en la partición F siempre está limitada por la cantidad de productos que ha calificado algún usuario $c_i \in \mathrm{C}$. Es decir, sea n_i la cantidad de productos que ha calificado el usuario c_i se debe cumplir que $L \leq n_i \, \forall_i, i = 1, \dots M$, es decir, la cantidad de particiones del conjunto de productos siempre debe ser menor a la cantidad de productos que ha calificado cada uno de los usuarios.

Suponga que β es la mínima cantidad de productos que han sido calificados por algún usuario. Además asuma que $L \leq \beta$ y que los L subconjuntos de F son todos de igual tamaño; la probabilidad $\Pr_i(e > 0)$ de que la fila $i, i = 1, \dots, M$ de cualquier sub-matriz tenga al menos un elemento conocido está determinado por $1 - \Pr_i(e = 0)$. A su vez, la probabilidad $\Pr_i(e = 0)$ de que una fila esté completamente vacía está determinada por la proporción entre la cantidad de combinaciones posibles en la que todos los elementos del mismo subconjunto son desconocidos y la totalidad de combinaciones posibles de los productos entre los diferentes subconjuntos. Es decir, sea β la mínima cantidad de productos calificados, N la cantidad de productos y L la cantidad de particiones, la probabilidad de que una fila i de cualquier sub-matriz esté vacía se determina según la ecuación en (16).

$$\Pr_i(e = 0) = \frac{combinaciones\ que\ generan\ una\ fila\ vacía}{total\ de\ combinaciones\ posibles} = \left(\frac{\binom{N-\beta}{N/L}}{\binom{N}{N/L}} \right) \qquad (16)$$

En la BD del SR SURFEOUS el peor caso corresponde a un usuario que solo calificó 3 productos diferentes. Además, teniendo en cuenta que la técnica de validación extrae un dato para prueba, claramente la cantidad de particiones L para los productos no puede ser mayor a 2.

4. K: Cantidad de elementos en la partición H.

K hace referencia a la cantidad de subconjuntos en la que se divide el conjunto original de usuarios. Observe que al dividir el conjunto de usuarios, la probabilidad de que una sub-matriz quede con columnas completamente desconocidas es superior. Suponga un caso en el que un producto ha sido calificado únicamente por

3 usuarios y además se desea realizar una partición de 4 subconjuntos, en este ejemplo de ninguna manera será posible satisfacer las condiciones de completar matrices puesto que sólo existen tres elementos y son necesarios al menos 4 valores conocidos. Por lo tanto, la cantidad de subconjuntos K en la partición H siempre está limitada por la cantidad de usuarios que han calificado un producto $p_j \in \mathrm{P}$. Es decir, sea n_j la cantidad de usuarios que han calificado el producto p_j se debe cumplir que $K \leq n_j \, \forall_j, j = 1, \dots N$, es decir, la cantidad de particiones del conjunto de usuarios siempre debe ser menor a la cantidad de usuarios que han calificado un producto.

Suponga que γ es la mínima cantidad de usuarios que han calificado algún producto. Además asuma que $K \leq \gamma$ y que los K subconjuntos de H son todos de igual tamaño; la probabilidad $\mathrm{Pr}_j(e > 0)$ de que la columna $j, j = 1, \dots, N$ de cualquier sub-matriz tenga al menos un elemento conocido está determinado por $1 - \mathrm{Pr}_j(e = 0)$. A su vez, la probabilidad $\mathrm{Pr}_j(e = 0)$ de que una columna esté completamente vacía está determinada por la proporción entre la cantidad de combinaciones posibles en la que todos los elementos del mismo subconjunto son desconocidos y la totalidad de combinaciones posibles de los usuarios entre los diferentes subconjuntos. Es decir, sea γ la mínima cantidad de usuarios que han calificado un producto, M la cantidad de usuarios y K la cantidad de particiones, la probabilidad de que una columna j de cualquier sub-matriz esté vacía se determina según la ecuación en (17).

$$\mathrm{Pr}_j(e = 0) = \frac{combinaciones\ que\ generan\ una\ columna\ vacía}{total\ de\ combinaciones\ posibles} = \left(\frac{\binom{M-\gamma}{M/K}}{\binom{M}{M/K}}\right) \qquad (17)$$

En la BD empleada el peor caso corresponde a un producto que solo fue calificado por 5 usuarios. Adicionalmente, para el rendimiento de los algoritmos se desea que la probabilidad $\mathrm{Pr}_j(e = 0)$ de encontrar una columna completamente vacía sea inferior al 10%. Por lo anterior, se establece que la cantidad de particiones de

usuarios K no puede ser superior a 2, obteniendo así una probabilidad de encontrar una columna vacía de $2{,}9\ \%$.

En Tabla 8 se muestra cada uno de los factores considerados en los experimentos con sus respectivos niveles. La combinación de estos factores genera 24 tratamientos o experimentos diferentes. Finalmente, se realizaron 100 réplicas de cada experimento para determinar los resultados promedio. Es decir, se llevaron a cabo 2400 simulaciones en total.

Tabla 8 Factores y niveles de los experimentos

Factor	**Niveles**		
Método de partición	Aleatorio	Pearson	SVD
Solucionador	FPC	LmaFit	
K	$\|K\| = 1$	$\|K\| = 2$	
L	$\|L\| = 1$	$\|L\| = 2$	

La Tabla 9 muestra de manera resumida los parámetros de los experimentos realizados para validar el rendimiento del algoritmo DeMBaR.

Tabla 9 Parámetros de los experimentos

Técnica de validación	Validación cruzada dejando uno fuera.
Datos de entrenamiento	$88{,}11\ \%$
Datos de prueba	$11{,}89\ \%$
Experimentos	24
Réplicas por experimento	100

El agoritmo DeMBaR fue implementado en la herramienta software Matlab R2015a. La Tabla 10 muestra las características físicas del equipo de cómputo en el cual fueron realizadas las simulaciones.

Tabla 10 Características físicas del equipo computacional

Procesador	Intel(R) Xeon(R) CPU ES-2697 v3 @ 2.60GHz
Memoria RAM	192 GB
Sistema Operativo	Windows 10 Pro
Núcleos	28

8. ANÁLISIS DE RESULTADOS

En esta sección se presentan y analizan los resultados numéricos obtenidos teniendo en cuenta los tres enfoques de evaluación propuestos en la metodología: soporte de decisión, exactitud y rendimiento computacional. En cada caso se compararon los resultados estadísticamente considerando cada tratamiento descrito en el diseño experimental.

8.1.SOPORTE DE DECISIÓN

8.1.1 Precisión La Tabla 11 muestra la comparación de la métrica precisión descrita en (10) entre cada tratamiento considerado. Las etiquetas de $K_i L_j$ indican la combinación de la partición de usuarios de tamaño $K = i$ y la partición de productos de tamaño $L = j$. Las etiquetas Aleatorio, Similitud y Svd indican el método empleado para determinar los elementos en cada partición y las etiquetas LmaFit y FPC indican el algoritmo para completar las matrices de bajo rango.

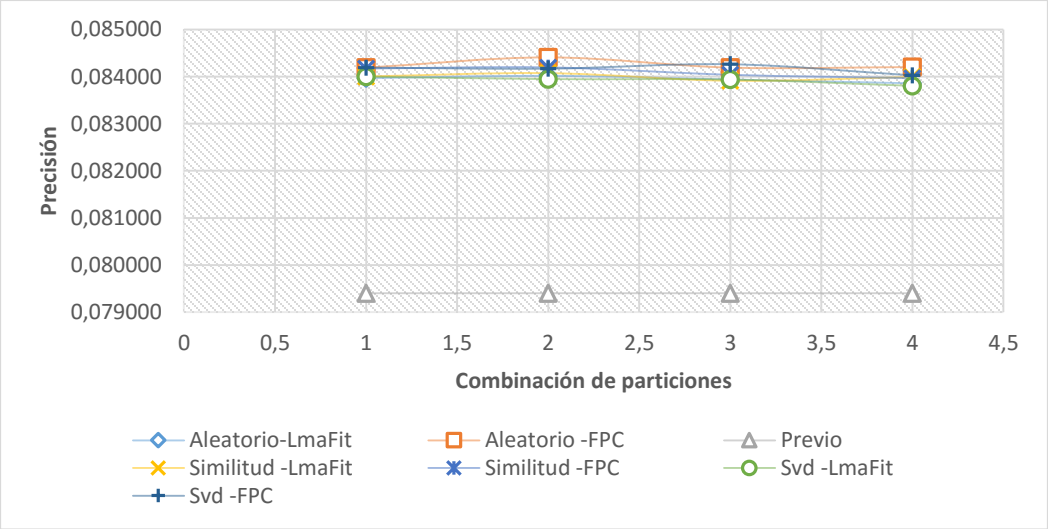
Tabla 11 Comparación métrica de precisión del algoritmo DeMBaR

Grupos	Aleatorio		Similitud		Svd	
	LmaFit	FPC	LmaFit	FPC	LmaFit	FPC
K_1L_1	0,083967	0,084193	0,084003	0,084169	0,084003	0,084193
K_2L_1	0,084015	0,084407	0,084073	0,084193	0,083948	0,084169
K_1L_2	0,083939	0,084193	0,083911	0,084039	0,083937	0,084264
K_2L_2	0,083861	0,084205	0,083990	0,083967	0,083800	0,084027

Teniendo en cuenta el objetivo planteado en este proyecto, el rendimiento del algoritmo DeMBaR fue comparado con el rendimiento del algoritmo empleado en el estado de arte sobre el sistema de recomendación bajo estudio. El algoritmo para generar las recomendaciones actualmente corresponde a un algoritmo de enfoque semántico. Como lo ilustra la Figura 17 el algoritmo DeMBaR supera al algoritmo

semántico usado actualmente y propuesto en[64] teniendo como referencia la métrica precisión. En la Figura 17 se muestra la precisión obtenida para cada tratamiento evaluado sobre el algoritmo DeMBaR. En el eje x las etiquetas 1, 2, 3 y 4 corresponden a las combinaciones K_1L_1, K_2L_1, K_1L_2 y K_2L_2. Observe que en promedio se obtuvo una mejora porcentual de hasta 5% en la métrica de precisión cuando se usó el algoritmo propuesto DeMBaR que sigue el enfoque de minimización del rango de una matriz comparado con el enfoque semántico del estado del arte.

Figura 17 Comparación de precisión DeMBaR – Enfoque semántico



El algoritmo propuesto supera hasta en un 5% al algoritmo previo

Por otro lado, se realizó un análisis de varianza sobre los datos presentados en la Tabla 11 con el objetivo de evaluar si existe diferencia significativa entre cada uno de los tratamientos evaluados. Observe en la Tabla 12 el resumen de la información estadística para la métrica de precisión.

Tabla 12 Resumen información estadística métrica Precisión

[64] VARGAS, Blanca. Op cit.

RESUMEN	Cuenta	Suma	Promedio	Varianza
K_1L_1	6	0,504528	0,084088	1,15595E-08
K_2L_1	6	0,504805	0,084134	2,63696E-08
K_1L_2	6	0,504284	0,084047	2,21564E-08
K_2L_2	6	0,503849	0,083975	1,99306E-08
Aleatorio LmaFit	4	0,335782	0,083945	4,17352E-09
Aleatorio FPC	4	0,336998	0,084250	1,10624E-08
Similitud LmaFit	4	0,335977	0,083994	4,39264E-09
Similitud FPC	4	0,336368	0,084092	1,15454E-08
Svd LmaFit	4	0,335688	0,083922	7,462E-09
Svd FPC	4	0,336653	0,084163	9,94317E-09

Teniendo en cuenta la información presentada en la Tabla 12 se observa que la combinación $K = 2\ y\ L = 1$ obtuvo una mayor precisión respecto a las demás y la combinación de selección aleatoria de los elementos en las particiones con el solucionador FPC obtuvo una mayor precisión.

Sin embargo, para verificar si existe diferencia significativa entre los tratamientos del experimento se realizó un análisis de varianza empleando la prueba de Fisher F con un nivel de significancia del $0{,}05$.

- Planteamiento de hipótesis:

$$H_0 = La\ media\ entre\ tratamientos\ es\ igual$$
$$H_1 = La\ media\ entre\ tratamientos\ no\ es\ igual$$

- Obtención de estadístico de prueba F de Fisher:

Observe en la Tabla 13 los valores calculados para determinar el estadístico de prueba F de Fisher.

Tabla 13 Análisis de varianza métrica Precisión

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Filas	8,21608E-08	3	2,73869E-08	6,461569573	0,005045532	3,287382105
Columnas	3,36504E-07	5	6,73008E-08	15,87870008	1,58197E-05	2,901294536
Error	6,35765E-08	15	4,23843E-09			
Total	4,82241E-07	23				

- Conclusión:

De acuerdo a los resultados de la Tabla 13 donde el estadístico $F = 6{,}4615 > 3{,}2874$ se rechaza la hipótesis nula con un nivel de significancia del $0{,}05$. Es decir, existe diferencia significativa entre las filas de los datos analizados. Por lo tanto, la precisión media obtenida entre los tratamientos correspondientes a las diferentes combinaciones entre la cantidad de subconjuntos en la partición del conjunto de usuarios K y la cantidad de subconjuntos en la partición del conjunto de productos L no se considera igual, por lo tanto éste parámetro genera varianza en el resultado. Como se mencionó anteriormente la mejor combinación fue $K = 2, L = 1$.

De acuerdo a los resultados de la Tabla 13 donde el estadístico $F = 15{,}8787 > 2{,}9013$ se rechaza la hipótesis nula con un nivel de significancia del $0{,}05$. Es decir, existe diferencia significativa entre las columnas de los datos analizados. Por lo tanto, la precisión media obtenida entre los tratamientos correspondientes a las diferentes combinaciones entre el método de completar matrices y el método para determinar los elementos en las particiones no se considera igual. Como se mencionó anteriormente la mejor combinación resultó de la interacción *OptSub* = Aleatorio y *OptSol* = FPC.

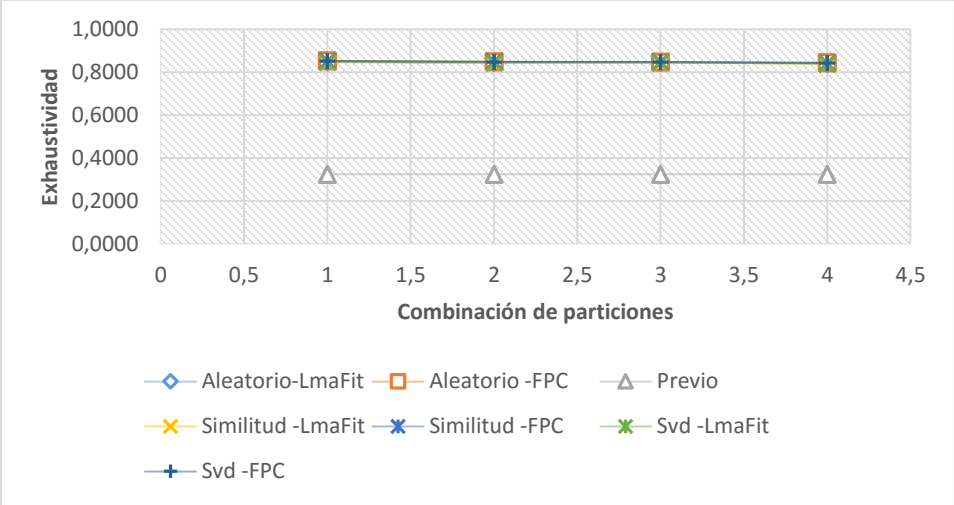
8.1.2 Exhaustividad La Tabla 14 muestra los resultados en términos de la métrica exhaustividad descrita en (11) para cada uno de los tratamientos en experimento.

Tabla 14 Comparación métrica de exhaustividad del algoritmo DeMBaR

Grupos	Aleatorio		Similitud		Svd	
	LmaFit	FPC	LmaFit	FPC	LmaFit	FPC
$K1L1$	0,8493	0,8540	0,847	0,8530	0,8485	0,8525
$K2L1$	0,8469	0,8496	0,8437	0,8488	0,8445	0,8487
$K1L2$	0,8459	0,8476	0,8430	0,8474	0,8460	0,8480
$K2L2$	0,8422	0,8452	0,8403	0,8424	0,8397	0,8434

De manera similar, los resultados obtenidos con el algoritmo DeMBaR en exhaustividad fueron comparados con los resultados alcanzados en el estado del arte para el sistema de recomendación bajo estudio. En la Figura 18 se observa que todos los tratamientos analizados en el diseño experimental superan los resultados obtenidos bajo el enfoque semántico actual. Note que en promedio se obtuvo una mejora de hasta 53% en la métrica de exhaustividad lo que indica que el algoritmo propuesto mejoró considerablemente la capacidad de recuperar los resultados relevantes para el usuario en el proceso de recomendación.

Figura 18 Comparación de exhaustividad DeMBaR – Enfoque semántico



El algoritmo propuesto superó en un 53% la exhaustividad respecto al algoritmo previo.

Por otro lado, observe en la Tabla 15 el resumen de datos estadísticos calculados para la métrica exhaustividad.

Tabla 15 Resumen estadístico métrica exhaustividad

RESUMEN	Cuenta	Suma	Promedio	Varianza
K_1L_1	6	5,10430	0,85072	7,98167E-06
K_2L_1	6	5,08221	0,84704	5,99372E-06
K_1L_2	6	5,07791	0,84632	3,37544E-06
K_2L_2	6	5,05315	0,84219	4,11425E-06
Aleatorio LmaFit	4	3,38429	0,84607	8,72467E-06
Aleatorio FPC	4	3,39640	0,84910	1,39067E-05
Similitud LmaFit	4	3,37401	0,84350	7,59311E-06
Similitud FPC	4	3,39160	0,84790	1,91067E-05
Svd LmaFit	4	3,37867	0,84467	1,38801E-05
Svd FPC	4	3,39260	0,84815	1,39367E-05

Teniendo en cuenta la información presentada en la Tabla 15 se observa que la combinación $K = 1\ y\ L = 1$ obtuvo una mayor exhaustividad respecto a las demás y la combinación de selección aleatoria de los elementos en las particiones con el solucionador FPC obtuvo una mayor exhaustividad.

Sin embargo, para verificar si existe diferencia significativa entre los tratamientos del experimento se realizó un análisis de varianza empleando la prueba de Fisher F con un nivel de significancia del $0{,}05$.

- Planteamiento de hipótesis:

$$H_0 = La\ media\ entre\ tratamientos\ es\ igual$$
$$H_1 = La\ media\ entre\ tratamientos\ no\ es\ igual$$

- Obtención de estadístico de prueba F de Fisher:

Observe en la Tabla 16 los datos calculados para determinar el estadístico de prueba F.

Tabla 16 Análisis de varianza métrica exhaustividad

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Filas	0,00021988	3	7,3294E-05	95,09226987	5,49873E-10	3,2873821
Columnas	9,5764E-05	5	1,91528E-05	24,84896218	9,25201E-07	2,90129454
Error	1,1562E-05	15	7,70768E-07			
Total	0,00032721	23				

- Conclusión:

Teniendo en cuenta los resultados de la Tabla 16 donde el estadístico obtenido $F = 95{,}0923 > 3{,}2874$ se rechaza la hipótesis nula con un nivel de significancia de $0{,}05$. Es decir, existe diferencia significativa entre las filas de los datos analizados. Por lo tanto, al menos una combinación en la cantidad de elementos en las particiones tiene media diferente a las demás. Este resultado indica que los parámetros K y L no solo afectan la precisión sino también la exhaustividad obtenida en los resultados.

Teniendo en cuenta los resultados de la Tabla 16 donde el estadístico obtenido $F = 24{,}8490 > 2{,}9013$ se rechaza la hipótesis nula con un nivel de significancia de $0{,}05$. Es decir, existe diferencia significativa entre las columnas de los datos analizados. Por lo tanto, al menos un tratamiento correspondiente a la combinación de algoritmos para seleccionar los elementos de los subconjuntos en la partición del conjunto de usuarios y de productos con los algoritmos de solución tiene media diferente a las demás.

8.2.EXACTITUD

Para evaluar la capacidad del algoritmo DeMBaR de predecir de manera exacta el puntaje otorgado por un usuario a un producto se calculó el Error Absoluto Medio descrito en (7). Observe en la Tabla 17 los resultados obtenidos para cada tratamiento descrito en el diseño de experimentos.

Tabla 17 Error Absoluto Medio para los 24 tratamientos

Grupos	Aleatorio		Similitud		Svd	
	LmaFit	FPC	LmaFit	FPC	LmaFit	FPC
K_1L_1	0,1308	0,1019	0,1442	0,1004	0,1325	0,1024
K_2L_1	0,1462	0,1239	0,1532	0,1253	0,1549	0,1241
K_1L_2	0,1477	0,1245	0,1619	0,1239	0,1489	0,1252
K_2L_2	0,1689	0,1455	0,1723	0,1452	0,1749	0,1448

Como se observa en la Tabla 17 el menor error absoluto medio fue obtenido mediante la combinación: $K = 1, L = 1, OptSub = Similitud$ y $OptSol = FPC$ obteniendo un error de 0,1004 en una escala de 1 a 3.

Tomando como referencia la métrica Error Absoluto Medio (EAM) se realizó un análisis de varianza para determinar si había diferencia significativa entre los niveles del parámetro *OptSol,* es decir entre los algoritmos de completar matrices LmaFit y FPC. Para ello, se utilizó la prueba t de student con un nivel de significancia de 0,05.

- Planteamiento de hipótesis:

$$H_0 = La\ media\ entre\ los\ dos\ solucionadores\ es\ igual$$
$$H_1 = La\ media\ entre\ los\ dos\ solucionadores\ no\ es\ igual$$

- Obtención de estadístico de prueba t de student:

Observe en la Tabla 18 los cálculos realizados para determinar el estadístico de prueba t de student.

- Conclusión:

Teniendo en cuenta los resultados de la Tabla 18 donde $t = 4{,}6697 > 2{,}0739$ se rechaza la hipótesis nula para un nivel de significancia del $0{,}05$. Es decir, existe diferencia significativa entre la media del error medio absoluto cuando se emplea el

algoritmo LmaFit y la media del algoritmo FPC. Observe que el algoritmo FPC obtuvo un EAM en promedio 0,03 puntos más pequeño que el obtenido con el LmaFit.

Tabla 18 Estadístico de prueba t de student parámetro OptSol

	LmaFit	FPC
Media	0,153032	0,123925
Varianza	0,000206	0,00026
Observaciones	12	12
Diferencia hipotética de las medias	0	
Grados de libertad	22	
Estadístico t	4,669652	
P(T<=t) dos colas	0,000118	
Valor crítico de t (dos colas)	2,073873	

Adicionalmente se analizó si existía diferencia significativa entre los niveles del parámetro *OptSub,* es decir, entre la metodología empleada para seleccionar los elementos de los subconjuntos en la partición del conjunto de usuarios y en la partición del conjunto de productos. En este caso se realizó un análisis de varianza empleando el estadístico F de Fisher considerando los niveles: enfoque aleatorio, enfoque por correlación de Pearson y enfoque por descomposición en valores singulares SVD.

- Planteamiento de hipótesis:

$$H_0 = La\ media\ entre\ los\ tres\ enfoques\ es\ igual$$
$$H_1 = La\ media\ entre\ los\ tres\ enfoques\ no\ es\ igual$$

- Obtención de estadístico de prueba F de Fisher:

Observe en la Tabla 19 los cálculos realizados para determinar el estadístico F de Fisher.

Tabla 19 Estadístico de prueba F de Fisher parámetro OptSub

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	8,582E-05	2	4,2912E-05	0,088992	0,915195917	3,466800112
Dentro de los grupos	0,0101262	21	0,0004822			
Total	0,010212	23				

- Conclusión:

Teniendo en cuenta los datos presentados en la Tabla 19 donde $F = 0{,}08899 < 3{,}4668$ no hay evidencia para rechazar la hipótesis nula con un nivel de significancia de $0{,}05$. Es decir, se acepta que no existe diferencia significativa entre la media del error medio absoluto cuando se emplea el enfoque aleatorio, el enfoque por correlación de Pearson o el enfoque por SVD para determinar los elementos que pertenecen a cada subconjunto de la partición de usuarios y de productos.

Finalmente se realizó un análisis de varianza de los parámetros K= cantidad de subconjuntos en la partición de usuarios y $L =$ cantidad de subconjuntos en la partición de productos. Para ello en la Tabla 20 se muestra un resumen de los resultados obtenidos para los tratamientos resultantes de la combinación de los dos niveles de cada parámetro respectivamente: 1 y 2 subconjuntos.

En la Tabla 20 se observa que el menor error absoluto medio se obtiene cuando se combinan los niveles bajos de cada factor o parámetro en cuestión, es decir, cuando la cantidad de subconjuntos en la partición del conjunto de usuarios es $K = 1$ y la cantidad de subconjuntos en la partición del conjunto de productos es $L = 1$. En este caso se obtuvo un EAM de 0,1187.

Para verificar el efecto de cada factor independientemente y de su interacción se realizó un análisis de varianza para estos dos factores. En la Tabla 21 se observan los cálculos realizados para determinar los estadísticos de la prueba F de Fisher.

Tabla 20 Resultados de la combinación de los parámetros *K y L*

RESUMEN	K1	K2	Total
L1			
Cuenta	6	6	12
Suma	0,7122	0,827621	1,539821
Promedio	0,1187	0,137937	0,128318
Varianza	0,000374	0,000228	0,000374
L2			
Cuenta	6	6	12
Suma	0,832072	0,951593	1,783665
Promedio	0,138679	0,158599	0,148639
Varianza	0,000265	0,00022	0,000329
Total			
Cuenta	12	12	
Suma	1,544272	1,779213	
Promedio	0,128689	0,148268	
Varianza	0,000399	0,00032	

Tabla 21 Análisis de varianza parámetros K y L

Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Muestra	0,00247749	1	0,002477492	9,118591	0,006768743	4,351243503
Columnas	0,00229989	1	0,002299891	8,464919	0,008669066	4,351243503
Interacción	7,0017E-07	1	7,0017E-07	0,002577	0,960016823	4,351243503
Dentro del grupo	0,00543394	20	0,000271697			
Total	0,01021202	23				

Teniendo en cuenta los resultados observados en la Tabla 21 donde $F = 9{,}1186 > 4{,}3512$ se rechaza la hipótesis nula, indicando que si existe diferencia significativa entre tener una partición de 1 o 2 subconjuntos a partir del conjunto de productos para un nivel de significancia del 5%. De manera similar, dado que $F = 8{,}4649 > 4{,}3512$ se rechaza la hipótesis nula, indicando que existe diferencia significativa entre tener una partición de 1 o 2 subconjuntos a partir del conjunto de usuarios.

Finalmente, dado que $F = 0{,}00257 < 4{,}3512$ no hay evidencia suficiente para rechazar la hipótesis nula, lo cual indica que el efecto de interacción no genera variaciones significativas en los resultados obtenidos para la métrica EAM.

8.3. RENDIMIENTO COMPUTACIONAL

Esta métrica se calculó con el objetivo de evaluar el rendimiento en términos del tiempo de cómputo requerido por el algoritmo propuesto DeMBaR. Particularmente, se analizó si la propuesta de subdividir la matriz original en varias sub-matrices permitía reducir el tiempo computacional consumido.

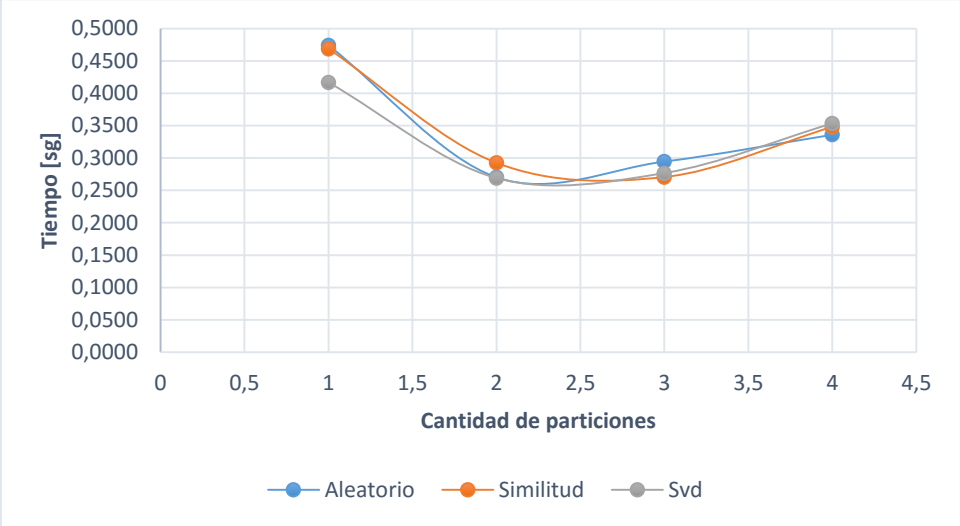
En la Tabla 22 se muestra el tiempo de cómputo promedio en segundos para cada uno de los tratamientos del diseño experimental. En esta tabla se puede observar que se obtuvo una reducción en el tiempo computacional de hasta 43% para el algoritmo LmaFit y de hasta 97% para el algoritmo FPC. Adicionalmente, note que el algoritmo LmaFit es computacionalmente más eficiente ya que tarda aproximadamente 0,3% del tiempo que consume el algoritmo FPC.

Tabla 22 Tiempo de cómputo algoritmo DeMBaR

Grupos	Aleatorio		Similitud		Svd	
	LmaFit	FPC	LmaFit	FPC	LmaFit	FPC
K1L1	0,4744	155,6935	0,4686	146,8858	0,4167	143,3964
K2L1	0,2704	13,5543	0,2925	11,9540	0,2689	13,3313
K1L2	0,2947	4,1731	0,2706	4,3546	0,2766	4,4370
K2L2	0,3361	8,1955	0,3481	8,4627	0,3533	7,7567

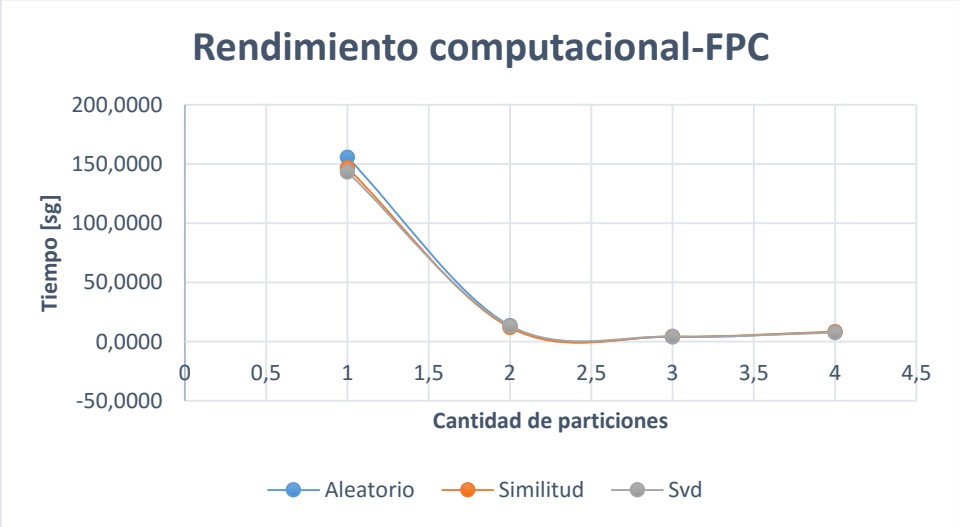
Observe en la Figura 19 y Figura 20 que efectivamente operar sobre una cantidad mayor de matrices pero de una dimensión menor a la original, permite reducir el tiempo computacional consumido por el algoritmo sin importar si el solucionador es el LmaFit o el FPC.

Figura 19 Rendimiento computacional con el solucionador LmaFit



En el eje x las etiquetas $1, 2, 3\ y\ 4$ corresponden a los tratamientos $K_1L_1, K_2L_1, K_1L_2\ y\ K_2L_2$. Las etiquetas Aleatorio, Similitud y Svd hacen referencia al algoritmo empleado para definir los subconjuntos en las particiones.

Figura 20 Rendimiento computacional con el solucionador FPC



En el eje x las etiquetas $1, 2, 3\ y\ 4$ corresponden a los tratamientos $K_1L_1, K_2L_1, K_1L_2\ y\ K_2L_2$. Las etiquetas Aleatorio, Similitud y Svd hacen referencia al algoritmo empleado para definir los subconjuntos en las particiones.

9. CONCLUSIONES

- En este proyecto de grado se diseñó, implementó y aplicó un algoritmo para generar recomendaciones bajo el enfoque de reducción dimensional, particularmente la metodología de minimización del rango de una matriz de interacción U-P.

- El algoritmo propuesto denominado DeMBaR propone una metodología de descomposición de la matriz original en pequeñas sub-matrices de bajo rango que deben ser completadas con el objetivo de reducir la complejidad computacional.

- El rendimiento del algoritmo propuesto DeMBaR fue evaluado en un prototipo de sistema de recomendación de restaurantes mediante las métricas de soporte de decisión, exactitud y rendimiento computacional.

- Se obtuvo una mejora de hasta 5% en términos de precisión y de hasta 53% en términos de exhaustividad cuando se empleó el algoritmo propuesto comparado con el enfoque semántico propuesto en el estado del arte.

- Los factores cantidad de subconjuntos en la partición del conjunto de usuarios K y cantidad de subconjuntos en la partición del conjunto de productos L generan una varianza significativa en los resultados obtenidos según las métricas precisión, exhaustividad y EAM.

- El factor *OptSol* que indica el algoritmo para completar las matrices de bajo rango genera una variación significativa en los resultados de acuerdo a la métrica EAM.

- El factor *OptSub* que indica el enfoque para seleccionar los elementos en cada subconjunto de las particiones no genera una variación significativa en los resultados de acuerdo a la métrica EAM. Es decir, usar cualquiera de las tres alternativas produce los mismos resultados estadísticamente.

- Se observó que variar independientemente los niveles de los parámetros K y L genera variaciones significativas en el resultado final en términos del EAM, sin embargo, la interacción de ambos factores no genera variaciones significativas.

- Se concluye que descomponer la matriz de interacción U-P en un conjunto de sub-matrices para aplicar la teoría de completar matrices reduce el tiempo de procesamiento de los datos sin importar cuál sea el algoritmo de solución empleado. Se obtuvo una reducción en el tiempo computacional de hasta 43% para el algoritmo LmaFit y de hasta 97% para el algoritmo FPC.

- El algoritmo FPC es un 30% más exacto comparado con el algoritmo LmaFit, sin embargo, es aproximadamente 330 veces más lento. Note que con la propuesta de dividir la matriz en un conjunto de sub-matrices se logró que el algoritmo FPC fuera un 23% más exacto y tan solo 16 veces más lento comparado con el algoritmo LmaFit.

10. RECOMENDACIONES

- Evaluar el algoritmo propuesto DeMBaR en otros dominios de aplicación con bases de datos más grandes para generalizar los resultados obtenidos en el proyecto presente.

- Proponer una metodología que minimice la limitación del algoritmo por escasez de datos. Es decir, que permita dividir la matriz original en un mayor número de sub-matrices.

- Proponer otros enfoques para determinar los elementos en cada subconjunto creado, de tal forma que genere diferencia significativa en los resultados obtenidos. De esta manera, se logra una mejor compresión del problema en su dominio de aplicación.

BIBLIOGRAFÍA

ADOMAVICIUS, Gediminas y TUZHILIN, Alexander. Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. En: IEEE transactions on knowledge and data engineering. Junio, 2005. vol.17, no. 6, p.734-749.

CAI, Jian, CANDES Emmanuel y SHEN Zuowei. A singular value thresholding algorithm for matrix completion. En: SIAM Journal on Optimization. Marzo, 2010. vol. 20. no. 4, p.1956-1982.

CANDES, Enmanuel, RECHT, Benjamin. Exact low-rank matrix completion via convex optimization. En: 46 Annual Allerton Conference on communication, control, and computing (4: 23-26, septie: Monticello, IL, USA). Communication, control, and computing 2008. Monticello, IL, USA: University of Illinois. 2008, p. 806–812.

HERLOCKER, Jonathan *et al.* Evaluating collaborative filtering recommender systems En: ACM Transactions on information systems. Enero, 2004. vol. 22. no. 1, p. 5–53

KOREN, Yehuda, BELL, Robert y VOLINSKY, Chris. Matrix factorization techniques for recommender systems En: IEEE Computer Society. Agosto, 2009. vol. 8 no. 42, p. 30-37.

MICHENKOVÁ, Marie. Numerical algorithms for low-rank matrix completion problems. 2011.

MOCEAN, Loredana, y CIPRIAN Marcel. Marketing recommender systems: a new approach in digital economy. En: Informática Economíca. Octubre, 2012. vol. 16. no. 4, p.142-149.

RICCI, Francesco, ROKACH, Lior y BRACHA, Shapira. Introduction to recommender systems handbook. En: Recommender systems handbook. Springer. New York Dordrecht Heidelberg London, 2011, p. 1-35.

SCHAFER, Ben, KONSTAN, Joseph y RIEDL, John. E-Commerce recommendation applications. En: Data mining and knowledge discovery. Enero-Abril, 2001. vol. 5, p.115-153.

SHIGIAN, Ma, GOLDFARB, Donald y CHEN, Lifeng. Fixed point and Bregman iterative methods for matrix rank minimization. En: Mathematical Programming. Junio, 2001. vol. 128. no. 1-2, p. 321-353.

ANEXOS

Anexo A Algoritmo FPC

Sea $\mathbf{R} \in \mathbb{R}^{M\times N}$ una matriz de bajo rango, $\operatorname{rank}(\mathbf{R}) \ll \min(M,N$ y $\mathbf{M} \in \mathbb{R}^{M\times N}$ una matriz que contiene los valores conocidos de la matriz $\mathbf{R}$ en el conjunto de posiciones $\Omega = \{(i,j) \in \mathbb{N}^2 \mid m_{ij}\ es\ conocido\ \}$**;** el algoritmo *Fixed Point Continuation* (FPC)[65] es un algoritmo iterativo que se basa en la minimización de la norma nuclear y resuelve el problema general en (18).

$$\begin{aligned} &\underset{\widehat{\mathbf{R}}}{\text{minimizar}}\ \left\|\widehat{\mathbf{R}}\right\|_* \\ &\text{sujeto a}\ \mathcal{P}_\Omega\left(\widehat{\mathbf{R}}\right) = \mathcal{P}_\Omega(\mathbf{M}), \end{aligned} \tag{18}$$

donde $\|\cdot\|_*$ representa la norma nuclear o suma de valores singulares. Esto es, $\left\|\widehat{\mathbf{R}}\right\|_* = \sum_{i=1}^{r}\sigma_i(\widehat{\mathbf{R}})$, con $r = rank(\widehat{\mathbf{R}})$, σ_i igual al valor singular i y $\mathcal{P}_\Omega\colon \mathbb{R}^{m\times n} \longrightarrow \mathbb{R}^{m\times n}$ es un operador de proyección definido como:

$$(\mathcal{P}_\Omega(\mathbf{B}))_{ij} = \begin{cases} B_{ij,}\ \text{if}\ (i,j) \in\ \Omega \\ 0, \text{otro caso}, \end{cases} \tag{19}$$

El método FPC[66] resuelve el problema en (20) para obtener la solución del problema en (18)

$$\widehat{\mathbf{R}} = \underset{\widehat{\mathbf{R}}}{\operatorname{argmin}}\ \mu\left\|\widehat{\mathbf{R}}\right\|_* + \frac{1}{2}\left\|\mathcal{P}_\Omega\left(\mathbf{M} - \widehat{\mathbf{R}}\right)\right\|_F^2, \tag{20}$$

donde μ es un parámetro de regularización y $\|\cdot\|_F$ denota la norma Frobenius. Esto es, la raíz cuadrada de la suma de los cuadrados de los valores absolutos de una matriz. Sea $\mathbf{B} \in \mathbb{R}^{\boldsymbol{m\times n}}$**,** la norma Frobenius, $\|\mathbf{B}\|_F = \sqrt{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|b_{ij}\right|^2}$. El pseudocódigo de la metodología iterativa de solución se presenta en Algoritmo 2.

[65] SHIGIAN, Ma. Op. Cit.
[66] MA, Shiqian. Op. Cit.

Algoritmo 2 FPC (Fixed Point Continuation)

— Entrada: $\widehat{\mathbf{R}}_{\mathbf{0}}, \overline{\boldsymbol{\mu}} > \mathbf{0}$. Seleccionar $\boldsymbol{\mu}_{\mathbf{1}} > \boldsymbol{\mu}_{\mathbf{2}} > \cdots > \boldsymbol{\mu}_{\boldsymbol{L}} = \overline{\boldsymbol{\mu}} > \mathbf{0}$. Ajustar $\widehat{\mathbf{R}} = \widehat{\mathbf{R}}_{\mathbf{0}}$

— **Para** $\boldsymbol{\mu} = \boldsymbol{\mu}_{\mathbf{1}}, \boldsymbol{\mu}_{\mathbf{2}}, \ldots \boldsymbol{\mu}_{\boldsymbol{L}}$**, hacer**

— **Mientras** NO converga, **hacer**

- **Seleccionar** $\boldsymbol{\tau} > \mathbf{0}$
- Calcular $\mathbf{Y} = \widehat{\mathbf{R}} - \boldsymbol{\tau}\, \boldsymbol{\mathcal{A}}^{*}(\boldsymbol{\mathcal{A}}(\widehat{\mathbf{R}}) - \mathbf{M})$ y la descomposición SVD de $\mathbf{Y}. \mathbf{Y} = \mathbf{UDiag}(\boldsymbol{\sigma})\mathbf{V}^{\mathsf{T}}$
- Calcular $\widehat{\mathbf{R}} = \mathbf{UDiag}(\boldsymbol{S}_{\boldsymbol{\tau\mu}}(\boldsymbol{\sigma}))\mathbf{V}^{\mathsf{T}}$ *

— **Fin mientras**

— **Fin para**

Adaptado de: MA, *et al*., 2011.

* $S_{\tau\mu}(\sigma)$ es un operador de umbralización que reduce el rango en cada iteración.

Anexo B Algoritmo LMaFit

Sea $\mathbf{R} \in \mathbb{R}^{M\times N}$ una matriz de bajo rango, $\mathrm{rank}(\mathbf{R}) \ll \min(M,N$ y $\mathbf{M} \in \mathbb{R}^{M\times N}$ una matriz que contiene los valores conocidos de la matriz $\mathbf{R}$ en el conjunto de posiciones $\Omega = \{(i,j) \in \mathbb{N}^2 \mid m_{ij}\ es\ conocido\ \}$**;** el algoritmo LMaFit (*Low-rank Matrix Fitting*)[67] se basa en la minimización de la norma Frobenius para resolver el problema general en (21).

$$\begin{aligned} &\underset{\widehat{\mathbf{R}}}{\text{minimizar}} \ \left\|\mathcal{P}_\Omega\left(\widehat{\mathbf{R}}\right) - \mathcal{P}_\Omega(\mathbf{M})\right\|_F \\ &\text{sujeto a } \mathrm{rank}\left(\widehat{\mathbf{R}}\right) \le r, \end{aligned} \tag{21}$$

donde r es el rango estimado de la matriz.

La propuesta del algoritmo LMaFit es relajar el problema en (21) solucionando el problema planteado en (22).

$$\begin{aligned} &\underset{\mathbf{U},\mathbf{V},\mathbf{Z}}{\text{minimize}} \ \ \|\mathbf{U}\mathbf{V}^\top - \mathbf{Z}\|_F^2 \\ &\text{subject to } \mathcal{P}_\Omega(\mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{M}), \end{aligned} \tag{22}$$

donde, $\mathbf{U} \in \mathbb{R}^{M\times \boldsymbol{r}}$**,** $\mathbf{V} \in \mathbb{R}^{r\times N}$ y $\mathbf{Z} \in \mathbb{R}^{M\times \boldsymbol{N}}$, la estimación de matriz de bajo rango resulta de $\widehat{\mathbf{R}} = \mathbf{U}\mathbf{V}^\top$.

En Algoritmo 3 se presenta el pseudocódigo del solucionador LMaFit.

[67] ZAIWEN, Wen. Op. Cit.

Algoritmo 3 LMaFit (*Low-rank Matrix Fitting*)

— **Entrada:** Conjunto $\boldsymbol{\Omega}$, datos en $\boldsymbol{\mathcal{P}_{\Omega}(\mathbf{M})}$**,** estimación del rango $\boldsymbol{K \geq r}$**.**

— Inicializar: $\mathbf{Y^0} \in \mathbb{R}^{K \times N}$, $\mathbf{Z^0} = \boldsymbol{\mathcal{P}_{\Omega}(\mathbf{M})}, \bar{\boldsymbol{\omega}} > \mathbf{1}, \boldsymbol{\delta} > \mathbf{0}, \boldsymbol{\gamma_1} \in (\mathbf{0}, \mathbf{1})$ y $\boldsymbol{k} = \mathbf{0}$.

— **Mientras** NO converga, **hacer**

1. **Calcular** $(\mathbf{X}_+(\boldsymbol{\omega}), \mathbf{Y}_+(\boldsymbol{\omega}), \mathbf{Z}_+(\boldsymbol{\omega}))$ según (23) con $(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = (\mathbf{X^k}, \mathbf{Y^k}, \mathbf{Z^k})$**.**
2. **Calcular** $\boldsymbol{\gamma(\omega)} = \frac{\|\boldsymbol{\mathcal{P}_{\Omega}}(\mathbf{M} - \mathbf{X}_+(\boldsymbol{\omega})\mathbf{Y}_+(\boldsymbol{\omega}))\|_F}{\|\boldsymbol{\mathcal{P}_{\Omega}}(\mathbf{M} - \mathbf{XY})\|_F}$
3. **Si** $\boldsymbol{\gamma(\omega)} \geq \mathbf{1}$ **entonces** ajustar $\boldsymbol{\omega} = \mathbf{1}$ y volver a **1.**
4. Actualizar $\left(\mathbf{X^{k+1}}, \mathbf{Y^{k+1}}, \mathbf{Z^{k+1}}\right) = \mathbf{X}_+(\boldsymbol{\omega}), \mathbf{Y}_+(\boldsymbol{\omega}), \mathbf{Z}_+(\boldsymbol{\omega}). \boldsymbol{k} = \boldsymbol{k} + \mathbf{1}$
5. **Si** $\boldsymbol{\gamma(\omega)} \geq \boldsymbol{\gamma_1}$ **entonces** $\boldsymbol{\delta} = \mathbf{max}(\boldsymbol{\delta}, \mathbf{0.25}(\boldsymbol{\omega} - \mathbf{1}))$ y $\boldsymbol{\omega} = \mathbf{min}(\boldsymbol{\omega} + \boldsymbol{\delta}, \bar{\boldsymbol{\omega}})$**.**

— **Fin mientras**

Adaptado de: ZAIWEN, et al, 2012.

$$
\begin{aligned}
\mathbf{Z}_w &\leftarrow \omega\mathbf{Z} + (1-\omega)\mathbf{XY}, \\
\mathbf{X}_+(\omega) &\leftarrow \mathbf{Z}_w\mathbf{Y}^\intercal \; o \; \mathbf{Z}_w\mathbf{Y}^\intercal(\mathbf{YY}^\intercal)^\dagger, \\
\mathbf{Y}_+(\omega) &\leftarrow \left(\mathbf{X}_+(\omega)^\intercal\mathbf{X}_+(\omega)\right)^\dagger(\mathbf{X}_+(\omega)^\intercal\mathbf{Z}_w), \\
\mathcal{P}_{\Omega^e}\left(\mathbf{Z}_+(\omega)\right) &\leftarrow \mathcal{P}_{\Omega^e}\left(\mathbf{X}_+(\omega)\mathbf{Y}_+(\omega)\right), \\
\mathcal{P}_{\Omega}\left(\mathbf{Z}_+(\omega)\right) &\leftarrow \mathcal{P}_{\Omega}(\mathbf{M}),
\end{aligned}
\tag{23}
$$