UNIVERSIDAD POLITÉCNICA DE MADRID

**Escuela Técnica Superior de Ingeniería
de Sistemas Informáticos**

**Doctorado en Ciencias y Tecnologías de la Computación
para Smart Cities**

Tesis Doctoral

Título:

Clustering de sistemas de recomendación
mediante técnicas de factorización matricial

Autor:

Rodolfo Xavier Bojorque Chasi

Tutores: Jesús Bobadilla , Fernando Ortega

Fecha: Febrero 2020

Dedicado a mi familia, sin su amor nada sería posible.

Con amor para Daysi, Matías y Eimi.

¡En todos los peligros yo te invoco Madre mía,

porque sé que eres mi escudo protector!

Don Bosco.

Declaración de autoría

Yo, Rodolfo Bojorque Chasi, declaro que esta tesis titulada “Clustering de sistemas de recomendación mediante técnicas de factorización matricial” y el trabajo presentado en ella es de mi autoría. Y confirmo que:

- Este trabajo se realizó en su totalidad durante la candidatura para un grado de investigación en esta universidad.
- Los contenidos de este trabajo son originales y no han sido enviados completa o parcialmente para la obtención de algún grado académico o título universitario en esta u otra universidad.
- Donde se ha utilizado trabajos publicados por otros, esto es claramente atribuido a sus autores.
- Donde se ha citado los trabajos de otros autores, las fuentes son siempre proporcionadas. Con la excepción de las fuentes que son de mi autoría.

Fmdo.

Fecha.

Agradecimientos

Primeramente deseo expresarle mi gratitud al creador, a mi papá Dios. A mi esposa Daysi, quien con su amor y fidelidad me brindó la oportunidad de alcanzar este logro, su fortaleza y entrega son los pilares de mi hogar. Mis hijos Mati y Eimicita son la luz que motiva mi vida. A mis tutores de tesis doctoral, Dr. Jesús Bobadilla y Dr. Fernando Ortega, sin su ayuda, guía y consejo nada hubiera sido posible. La academia más que nunca necesita profesores e investigadores de su calidad humana y profesional; cuando más oscuro se ponía el camino siempre sentí su palmada en la espalda, definitivamente un gran honor haber compartido este camino bajo su supervisión.

A la Universidad Politécnica Salesiana, de manera especial a su Rector, Padre Javier Herrán Gómez, Ph.D. Gracias por el apoyo y la confianza.

Resumen

La expansión de los sistemas de recomendación a nivel comercial e industrial ha permitido una rápida evolución de técnicas, métodos y algoritmos. Inicialmente las investigaciones se enfocaban en mejorar la calidad de las predicciones; sin embargo, existen desafíos importantes en cuanto a la generación de modelos que permitan el trabajo con los grandes volúmenes de información que actualmente se genera a todos los niveles. La dispersión de los conjuntos de datos es un desafío actual de los sistemas de recomendación. Aplicar técnicas como clustering y explicación de las recomendaciones son problemas de investigación actuales en el área de sistemas de recomendación. Los sistemas de recomendación basados en modelos proveen resultados más precisos, más escalables y abordan de mejor manera el problema de la dispersión. El modelo más adoptado por los sistemas de recomendación modernos es la Factorización Matricial y las técnicas derivadas de este enfoque.

Esta tesis expone un estudio exhaustivo de trabajos de investigación del estado del arte y propone el uso de un método de Factorización Matricial no Negativo Bayesiano (BNMF) para mejorar los resultados de clustering de los trabajos más recientes y representativos en el área de filtrado colaborativo. También propone un innovador algoritmo de pre-clustering adaptado al método probabilístico propuesto. Los resultados obtenidos sobre varios conjuntos de datos abiertos demuestran: 1) Mejoras importantes en la calidad de clustering cuando se utiliza BNMF, en comparación con la técnica de Factorización Matricial Clásica y con técnicas mejoradas y recientes del algoritmo k-means, 2) Una mejora en la calidad de las predicciones cuando se utiliza técnicas de clustering basadas en factorización matricial, en comparación con los estudios recientes de k-means, y 3) Mejores tiempos de ejecución de BNMF en comparación con los tiempos de ejecución de la factorización matricial clásica, y una mejora adicional cuando se utiliza el algoritmo de pre-clustering propuesto.

Abstract

The expansion of recommender systems to the commercial and industrial level has allowed a rapid evolution of techniques, methods and algorithms. Initially, research focused on improving the quality of predictions, however, there are significant challenges such as generating models able to work with large volumes of information. Data sparsity on datasets is a challenge for recommender systems. Clustering and explanation of recommendations are growing research fields in the recommender systems area. Model-based recommender systems provide more accurate predictions and recommendations, more scalable results and they better address the problem of sparsity. The most adopted model by the modern recommender systems is the Matrix Factorization and their derived techniques.

This thesis exposes a comprehensive study of the state-of-the-art research works and it proposes a Bayesian non-negative matrix factorization method to improve the current clustering results in the collaborative filtering area. We also provide an innovative pre-clustering algorithm adapted to the proposed probabilistic method. Results obtained using several open datasets show: 1) A conclusive clustering quality improvement when BNMF is used, compared to the classical Matrix Factorization or to the improved k-means results, 2) A higher predictions accuracy using Matrix Factorization based methods than using improved KMeans, and 3) Better BNMF execution times compared to those of the classic matrix factorization, and an additional improvement when using the proposed pre-clustering algorithm.

Índice general

Agradecimientos	V
Resumen	VII
Abstract	IX
Lista de figuras	XIII
Lista de tablas	XV
Lista de algoritmos	XVII
1. Introducción	1
1.1. ¿Qué son los sistemas de recomendación?	2
1.1.1. Tipos de sistemas de recomendación	5
1.1.2. Métodos o algoritmos de predicción	7
1.1.3. Métricas de calidad	8
1.2. ¿Qué es clustering?	14
1.2.1. Hard clustering	16
1.2.2. Soft clustering	17
1.2.3. Métricas de calidad de clustering	18
2. Estado del Arte	21
2.1. Sistemas de Recomendación basados en Factorización Matricial	22
2.2. Clustering en Sistemas de Recomendación	24
2.3. Motivación e Hipótesis	27

3. Formalización	30
3.1. Diseño del método y formulación	31
3.2. Ejemplo de ejecución (Running Example)	38
3.2.1. K-means y pre-clustering k-means (k-means+)	39
3.2.2. Factorización Matricial	40
3.2.3. Factorización Matricial no Negativa Bayesiana (BNMF)	42
3.2.4. Pre-Clustering y BNMF+	42
3.3. Pre-clustering BNMF+	44
4. Experimentos	49
4.1. Diseño de experimentos	50
4.2. Selección del número de clústers	52
4.3. Selección de los valores de α y β	54
4.4. Mejoras de clustering	59
4.5. Mejoras en la exactitud de las predicciones	62
4.6. Mejoras en tiempo de ejecución de Pre-clustering	64
Conclusiones	67
Conclusions	69
Apéndices	71
1. Lista de publicaciones del autor	71

Índice de figuras

1.1. Nivel de dispersión en conjuntos de datos abiertos	5
3.1. Representación gráfica del modelo probabilístico	31
3.2. Matriz de votos del datatoy	39
3.3. Clustering k-means y resultados de la predicción	39
3.4. Clustering k-means+ y resultados de la predicción	40
3.5. Clustering de factorización matricial y resultados de predicción	41
3.6. Clustering BNMF y resultados de predicción	42
3.7. Running example	43
3.8. Pre-clustering y BNMF	45
4.1. Distribución de los votos en los conjuntos de datos	51
4.2. MAE y Within-Cluster para determinar K	53
4.3. Distribución de los factores ocultos para los usuarios	56
4.4. Resultados de MAE y calidad de within-cluster	58
4.5. Resultados de clustering	61
4.6. Mejoras de clustering BNMF vs. baselines	62
4.7. Resultados de MAE	63
4.8. Tendencias de la precisión y within-cluster	64
4.9. Entrenamiento BNMF con y sin pre-clustering	65
4.10. Comparación de tiempos de ejecución	66

Índice de tablas

1.1. Medidas de similitud para RS	9
1.2. Medidas de calidad para Predicción en RS	11
1.3. Matriz de confusión para RS	11
1.4. Métricas para evaluación de la recomendación como conjunto	12
1.5. Medidas de calidad para evaluación de listas en RS	12
3.1. MAE con votos por defecto	45
4.1. Esquema del diseño de los experimentos	52
4.2. Distribución factores ocultos de los usuarios para MovieLens 1M	57
4.3. Resultados de α y β combinados	59

Índice de algoritmos

1.	k-means clustering	16
2.	Inicialización de parámetros BNMF	35
3.	Entrenamiento BNMF	36
4.	Predicciones BNMF	37
5.	Clustering de usuarios BNMF	38
6.	Inicialización de parámetros BNMF+	46
7.	Pre-clustering k-meansPlusLogPower	47

Capítulo 1 Introducción

La expansión de los sistemas de recomendación a nivel comercial e industrial ha permitido una rápida evolución de técnicas, métodos y algoritmos. En principio las investigaciones se enfocaban en mejorar la calidad de las predicciones, sin embargo, existen desafíos importantes en cuanto a la explicación de las recomendaciones y la generación de modelos que permitan el trabajo con los grandes volúmenes de información que actualmente se genera a todos los niveles. Este primer capítulo ofrece una introducción a los conceptos fundamentales con la finalidad de preparar a los lectores para las discusiones de los capítulos posteriores.

En la sección 1.1 se resume los conceptos fundamentales de los sistemas de recomendación. También se ofrece una visión global de las técnicas y métricas de calidad para ubicar el objeto de estudio. En la sección 1.2 se propone una visión general de las diferentes técnicas de clustering y cómo se orientan a los sistemas de recomendación. También se exponen las principales métricas de calidad de clustering.

1.1. ¿Qué son los sistemas de recomendación?

La expansión de las computadoras y redes de computadores ha permitido un crecimiento exponencial de la variedad de información disponible en la Web. La introducción de nuevos servicios de e-business como la venta de productos, comparaciones entre ellos y subastas, han abrumado a los usuarios, quienes desean tomar buenas decisiones sobre lo que compran, escuchan, leen, miran o desean. Gracias a los costos cada vez menores del almacenamiento y procesamiento de datos, los vendedores analizan cuidadosamente nuestras compras para recomendarnos otros productos y mejorar sus ventas. Los sitios web sociales acceden a nuestros contactos para ayudarnos a conectarnos con nuevos amigos y engancharnos con el sitio, y las estaciones de radio en línea recuerdan canciones omitidas para que nos sirvan mejor en el futuro [Lü et al., 2012]. Esta gigantesca disponibilidad de opciones en lugar de producir un beneficio para los usuarios, paradójicamente, a disminuido su bienestar. Esto ha abierto el debate, puesto que tener la posibilidad de elegir es bueno pero nuestra elección no siempre es la mejor. Según [Schwartz, 2004] la elección, con sus implicaciones de libertad, autonomía y autodeterminación, puede llegar a ser excesiva y, en última instancia, crear el sentido de que esta libertad puede llegar a ser considerada como una especie de tiranía que induce a la miseria.

Los sistemas de recomendación (RS) han llegado a ser indispensables en los entornos web, ya que constituyen una vía para incrementar la satisfacción de los clientes y posicionarse competitivamente en el mercado de las actividades del comercio electrónico [Lucas et al., 2013]. Sitios como Amazon.com, YouTube, Netflix, LinkedIn, Facebook, TripAdvisor, entre otros, se han interesado en el desarrollo de RS, al punto que varias compañías están desarrollando RS como parte de los servicios al que tienen acceso sus subscriptores. Este interés, según [Ricci et al., 2015d], se debe a las exigencias del mercado moderno: incrementar la venta de productos, diversificar las ventas, incrementar la satisfacción del usuario, fidelizar al usuario y comprender mejor los deseos del usuario. [Herlocker et al., 2004] resalta las ventajas que los RS representan para los usuarios: encontrar buenos productos, descubrir todos los buenos productos, anotar en contexto, recomendar

una secuencia de productos (lista de música), recomendar un grupo de productos (paquete turístico), simplemente revisar listas de interés, encontrar recomendadores veraces (validar la calidad de las recomendaciones), mejorar el perfil, medio de expresión, ayudar a otros e influir en otros.

Los RS funcionan como un filtro de información para sobrellevar el problema de la sobrecarga de información [Ortega et al., 2016, Noh et al., 2018]. Los RS tienen la capacidad de predecir el agrado de un usuario por un ítem que él desconoce. El funcionamiento de un RS consiste en recolectar información de las preferencias de sus usuarios para un conjunto de ítems (por ejemplo: películas, música, noticias, libros, chistes, aplicaciones, gadgets, websites, e-learning y destinos de viajes)[Bobadilla et al., 2013, Lu et al., 2015].

Los RS han sido usados en diferentes áreas [Park et al., 2012]. La mayoría de trabajos recientes se enfocan en recomendaciones de películas [Chen et al., 2015, Diao et al., 2014, Winoto and Tang, 2010, Qu et al., 2013, Su and Khoshgoftaar, 2009], sin embargo, otra gran cantidad de literatura se concentra en la recomendación de música [Tan et al., 2011, Schedl and Hauger, 2015, Kaminskis et al., 2013], RS de noticias [Karimi et al., 2018, Turcotte et al., 2015, Hopfgartner et al., 2014, Lu et al., 2014], recomendación de páginas Web y documentos [Lu et al., 2015]. El gran impulso en investigación de los RS se sustenta en la liberación de grandes bases de datos por parte de compañías y centros académicos, como ocurre con los conjuntos de datos: Netflix, MovieLens, Jester, EachMovie y Book-crossing; por ejemplo, Netflix ha galardonado con un millón de dólares, desde el año 2006 al 2009, al primer equipo que ha tenido éxito en mejorar sustancialmente el rendimiento de su sistema de recomendación privado [Koren, 2009]. A través de estas bases de datos la comunidad científica puede replicar experimentos para validar y mejorar las técnicas.

En los últimos años, los RS han demostrado ser un valioso medio para mitigar el problema de la sobrecarga de información, incluso abordan la problemática ofreciendo a los usuarios nuevas elecciones pertinentes y relevantes a sus afinidades. La tarea de los RS es convertir

los datos de los usuarios y sus preferencias en predicciones de sus posibles gustos e intereses futuros [Lü et al., 2012]. Los RS emergen, como un área de investigación independiente, a mediados de los noventa cuando los investigadores comienzan a enfocarse en problemas de recomendación [Adomavicius and Tuzhilin, 2005] con la aparición del primer trabajo sobre filtrado colaborativo [Resnick et al., 1994].

Los RS presentan un desafío: los conjuntos de datos son dispersos (sparse) y esta situación dificulta la predicción y clustering de grupos de usuarios [Lü et al., 2012]. La dispersión se debe a que en los conjuntos de datos para RS la mayoría de los usuarios emiten votos acerca de una pequeña porción de los ítems. La razón detrás de la dispersión de datos es que los usuarios no pueden expresar una opinión con respecto a todos los ítems, que suelen ser miles o decenas de miles. Por ejemplo: miles de películas o decenas de miles de productos en venta. Debido a ello las recomendaciones no serán tan precisas como desearíamos [Najafabadi et al., 2017]. Los investigadores han abordado este problema trabajando con grandes conjuntos de datos, puesto que disponer de más usuarios e ítems brinda mayor información para los RS en la fase de aprendizaje, a pesar de que los conjuntos de datos más grandes suelen ser más dispersos [Duma and Twala, 2018]. Se resalta el desafío de la dispersión puesto que uno de los objetivos de la presente tesis es desarrollar modelos que mejoren la exactitud de las predicciones para entornos dispersos. En la Figura 1.1 se puede apreciar las gráficas de los mapas de color de los votos en los conjuntos de datos MovieLens 1M y FilmTrust, el color negro representa que no existe voto en la intersección usuario-ítem, mientras que el color blanco representa la existencia de voto.

Para [Bobadilla et al., 2013] el proceso para clasificar un RS está basado en la combinación de las siguientes consideraciones:

- El tipo de datos disponibles en la base de datos (ej., votos, información de registro de los usuarios, relaciones entre usuarios como región, idioma, etc.).
- El algoritmo de filtrado.
- El modelo escogido.
- Las técnicas empleadas.
- Nivel de dispersión de la base de datos y la escalabilidad deseada.
- Rendimiento del sistema tanto en escalabilidad espacial(memoria) y temporal(cómputo).

Figura 1.1: Nivel de dispersión en conjuntos de datos abiertos

- El objetivo a conseguir (por ejemplo, mejorar la calidad de las predicciones, evaluar la lista de recomendaciones y el conjunto de recomendaciones).
- La calidad deseada de los resultados.

1.1.1. Tipos de sistemas de recomendación

La característica fundamental de un sistema de recomendación es su algoritmo de filtrado [Bobadilla et al., 2013]. La clasificación más difundida según [Ricci et al., 2015d, Bobadilla et al., 2013, Lü et al., 2012] divide los tipos de filtrado en:

- **Filtrado basado en contenido:** las recomendaciones se basan en el conocimiento que se tiene sobre los ítems que el usuario ha valorado o seleccionado (ya sea de forma implícita o explícita). El usuario recibe recomendaciones de ítems similares que le puedan agradar. El filtrado basado en contenido realiza recomendaciones basándose en las características similares entre ítems. Esto permite que se puedan hacer recomendaciones según los datos históricos, por ejemplo, de acuerdo al historial de compras de los usuarios, también es posible realizar recomendaciones basadas en el contenido de los objetos que el usuario consume o utiliza, como imágenes, sonidos y texto. Otra alternativa de análisis es realizar recomendaciones de acuerdo a las características de los objetos, por ejemplo lo que el usuario escucha, compra o mira. Ejemplo de estos sistemas recomendadores son YouTube y Google.
- **Filtrado demográfico:** las recomendaciones se realizan en función de las características particulares de los usuarios (edad, sexo, idioma, situación geográfica, pro-

fesión, etc.) para determinar las preferencias comunes de un usuario según su perfil [Bobadilla et al., 2013]. Para [Ricci et al., 2015d] este filtrado parte de la suposición de que las recomendaciones deberían ser generadas para determinados nichos demográficos, este enfoque ha adquirido popularidad en la literatura de mercadotecnia pero ha sido relativamente menos apropiado para la investigación de sistemas de recomendación.

- **Filtrado colaborativo:** El filtrado colaborativo consiste en determinar qué usuarios son similares a un usuario activo (o usuario al que hay que realizarle las recomendaciones) y a continuación, recomendar aquellos ítems que no han sido votados por el usuario y que han resultado bien valorados por los usuarios similares. En otras palabras, el filtrado colaborativo se basa en los votos que los distintos usuarios realizan sobre diferentes ítems, de tal manera que al contar con la suficiente información almacenada en el sistema es posible emitir recomendaciones al usuario activo según la información proporcionada por otros usuarios de características similares [Bobadilla et al., 2013]. Ejemplo de estos sistemas de recomendación son Netflix, MovieLens, FilmAffinity, etc.

Para [Najafabadi et al., 2017, Zhang et al., 2016b] el filtrado colaborativo K-NN (K nearest neighbors) se basa en tres pasos:

1. Se calcula las similitudes entre el usuario activo y el resto de usuarios, con alguna medida de similitud de la tabla 1.1.
2. Según las similitudes del paso uno se selecciona los vecinos más cercanos al usuario activo.
3. Se recomienda una lista de tamaño k de los mejores ítems por agregación de las preferencias de los vecinos más cercanos.

El filtrado colaborativo presenta dos ventajas muy importantes: la capacidad de realizar recomendaciones novedosas y la ventaja de no necesitar información personal de los usuarios registrados. Por lo tanto, no resulta muy costoso recolectar la información de los usuarios, ni mucho menos se requiere romper la barrera de su confidencialidad. Los sistemas de recomendación de filtrado colaborativo (CF RS) son los más difundidos por su interés práctico.

- **Filtrado híbrido:** Se combinan algunos de los tres algoritmos de filtrado mencionados anteriormente para realizar recomendaciones, generalmente una combinación de filtrado colaborativo y filtrado basado en contenido para explotar las ventajas de cada técnica.

Según [Adomavicius and Tuzhilin, 2005] el filtrado híbrido puede clasificarse como:

- La implementación de filtrado colaborativo y filtrado basado en contenido por separado, para combinar sus predicciones.
- Incorporar características de filtrado basado en contenido en un enfoque de filtrado colaborativo.
- Incorporar características de filtrado colaborativo en un enfoque de filtrado basado en contenido.
- Construir un modelo unificado que incorpore características tanto de filtrado colaborativo como de filtrado basado en contenido.

El filtrado híbrido suele utilizarse mediante la implementación de técnicas de inteligencia artificial como la lógica difusa [Al-Shamri and Bharadwaj, 2008], algoritmos genéticos [Gao and Li, 2008], redes de neuronas [Lee et al., 2002], redes bayesianas [de Campos et al., 2010] y características latentes [Shinde and Kulkarni, 2012]. Un ejemplo de estos sistemas es el recomendador de Amazon.

1.1.2. Métodos o algoritmos de predicción

Según [Hernando et al., 2016, Ortega et al., 2016, Lü et al., 2012, Breese et al., 1998] los sistemas de recomendación pueden ser clasificados en tres grupos:

- **Métodos basados en memoria (K-NN):** realizan las recomendaciones usando los k vecinos más similares a un usuario u determinado. Utilizan la información de los votos emitidos por los usuarios en los diferentes ítems, para determinar aquellos usuarios (vecinos de u) que comparten características similares con el usuario u , con la finalidad de basar una recomendación en la similaridad de los usuarios. Por lo general, estos métodos trabajan sobre una matriz de votos, que representa un conjunto de usuarios U quienes han votado por ítems de un conjunto I , la predicción de un ítem i no votado por el usuario u es computada como una agregación de los votos de los k usuarios más similares (k -vecinos) para el mismo ítem i [Bobadilla et al., 2010, Resnick et al., 1994]. Este algoritmo se conoce como K-Nearest Neighbors (algoritmo K-NN) [Adomavicius and Tuzhilin, 2005, Ricci et al., 2015b]. La similaridad entre dos usuarios u y v se determina a partir de *métricas de similaridad* de origen estadístico, donde se considera un conjunto común de votos de los usuarios u y v por los mismos

ítems I' identificando como r_{ui} y r_{vi} los votos del usuario u y del usuario v por el mismo ítem i . De acuerdo a [Patra et al., 2015, Bobadilla et al., 2012, Bobadilla et al., 2010], las medidas de similaridad más comunes se muestran en la tabla 1.1.

- **Métodos basados en modelos:** Mediante la información de los votos realizados por los usuarios en los diferentes ítems se entrenan modelos que identifican patrones que permiten realizar una recomendación. El objetivo de un método de CF basado en modelos es estimar los parámetros del modelo a partir de la matriz de votos (datos conocidos) bajo una función objetivo definida [Duma and Twala, 2018]. Actualmente, los métodos basados en modelos tienen mayor calidad en exactitud de las predicciones. Tienen particular importancia los modelos de Factorización Matricial (MF) que transforman a los usuarios e ítems al mismo espacio de factores latentes u ocultos. Para RS los modelos de MF se asemejan a una técnica común y bien establecida: Singular Value Decomposition (SVD). El desafío del CF basado en modelos es la falta de votos debido a la alta dispersión de los conjuntos de datos para RS.

[Koren et al., 2009] proporciona un enfoque importante en el área de RS basados en modelos: se muestra que MF mapea a los usuarios e ítems a un espacio de factores latentes, de acuerdo al modelo cada usuario es asociado a un vector $p_u \in \mathcal{R}^f$ y cada ítem a un vector $q_i \in \mathcal{R}^f$ de dimensionalidad f . Además, este enfoque se presta bien para modelar efectos temporales de la matriz de votos, lo que puede mejorar significativamente su exactitud. Descomponer los votos en términos distintos nos permite tratar diferentes aspectos temporales por separado. Específicamente, se identifica los siguientes efectos que varían con el tiempo: (1) biases de usuario $b_u(t)$, (2) biases de ítems $b_i(t)$ y (3) preferencias del usuario $P_u(t)$. Por otro lado, se especifica características de elementos estáticos, q_i , porque no esperamos una variación temporal significativa para los ítems, a diferencia de los humanos (usuarios), ya que son de naturaleza estática [Ricci et al., 2015a].

- **Métodos híbridos:** realizan la combinación de las técnicas de los dos métodos anteriores, para explotar las ventajas de cada técnica.

1.1.3. Métricas de calidad

Las métricas de calidad más utilizadas en RS son (1) Evaluación de la predicción, (2) Evaluación de la recomendación como conjunto y (3) Evaluación de la recomendación

Tabla 1.1: Diferentes medidas de similitud utilizadas para RS

Medida de similitud	Ecuación	Descripción
Coseno	$S(u,v) = \frac{\sum_{i \in I' \cap I''} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I'} r_{ui}^2} \sqrt{\sum_{i \in I''} r_{vi}^2}}$	Popular en el ámbito de information retrieval, se basa en el coseno del ángulo entre los vectores de votos de los usuarios u y v .
Coseno ajustado	$S(i,j) = \frac{\sum_{u \in U'} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U'} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{u \in U'} (r_{uj} - \bar{r}_j)^2}}$	Computa la correlación lineal entre los votos de dos ítems i y j , a diferencia de la similitud del coseno toma en cuenta los diferentes rangos de las escalas de votos, para ello se obtiene el respectivo promedio $\bar{r}$ de los votos de los ítems que comparten usuarios en común U' .
Correlación de Pearson (PC)	$S(u,v) = \frac{\sum_{i \in I'} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I'} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I'} (r_{vi} - \bar{r}_v)^2}}$	Popular para CF basado en memoria, mide como dos usuarios o ítems están linealmente relacionados, para ello es necesario calcular $\bar{r}_u$ que representa el promedio de todos los votos del usuario u y $\bar{r}_v$ que representa el promedio de todos los votos del usuario v .
Correlación de Pearson restringida (CPC)	$S(u,v) = \frac{\sum_{i \in I'} (r_{ui} - r_{med})(r_{vi} - r_{med})}{\sqrt{\sum_{i \in I'} (r_{ui} - r_{med})^2} \sqrt{\sum_{i \in I'} (r_{vi} - r_{med})^2}}$	Variante de PC, una referencia absoluta (la mediana en la escala de votos r_{med}) es utilizada en lugar del promedio de los votos de los usuarios.
Media de la Diferencia Cuadrática (MSD)	$S(u,v) = 1 - \frac{\sum_{i \in I'} (r_{ui} - r_{vi})^2}{ I' }$	Computa la diferencia Euclidea entre los vectores de votos del usuario u y v .
Jaccard	$S(u,v) = \frac{ I_u \cap I_v }{ I_u \cup I_v }$	Utiliza la información de los votos provista por un par de usuarios.
JMSD	$S(u,v) = S_{MSD}(u,v) \times S_{Jacc}(u,v)$	Combina la similitud de Jaccard $S_{Jacc}(u,v)$ y la similitud MSD $S_{MSD}(u,v)$ mediante el producto [Bobadilla et al., 2010].
(Mean-Jaccard-Differences) MJD	$S(u,v) = S_{v^0}(u,v) + S_{v^1}(u,v) + S_{v^2}(u,v) + S_{v^3}(u,v) + S_{v^4}(u,v) + S_{MSD}(u,v) + S_{Jacc}(u,v)$	Es una medida de similitud específicamente aplicada en filtrado colaborativo entre los usuarios u y v que combina varias medidas de similitud normalizadas [Bobadilla et al., 2012].

Donde v^0 es la cantidad de ítems en los que los dos usuarios tienen una diferencia de votos de 0 (el número de casos en que los dos usuarios tienen el mismo voto). v^4 corresponde a la cantidad de casos en los que los dos usuarios tienen la máxima diferencia de votos.

como listas jerárquicas [Hernández del Olmo and Gaudioso, 2008]. Los principales inconvenientes de los frameworks existentes de RS son: 1) falta de formalización: aunque las métricas están bien definidas existen una variedad de detalles que pueden conducir a la generación de resultados diferentes en experimentos similares. 2) Ausencia de estandarización de las medidas de evaluación en aspectos como la novedad y veracidad de la recomendación [Bobadilla et al., 2013].

- **Evaluación de la predicción:** Las métricas más utilizadas son: a) accuracy (exactitud), que consiste en medir la diferencia entre la predicción ($p_{u,i}$) y el valor real ($r_{u,i}$); y b) coverage (cobertura) que es la capacidad de predicción de una métrica aplicada a un sistema de recomendación específico. Que un RS sea exacto no es suficiente, por ello se suele utilizar la métrica de coverage. El problema surge cuando un RS realiza recomendaciones de alta exactitud solo para una pequeña porción de los ítems que tienen una enorme cantidad de votos. Los métodos basados en modelos han superado el problema de coverage, puesto que el modelo permite obtener la totalidad de las predicciones sin depender de los k-vecinos. La tabla 1.2 muestra diferentes medidas de similitud, donde sea O_u el conjunto de ítems votados por el usuario u con valores de predicción $O_u = \{i \in I | p_{u,i} \neq \bullet \wedge r_{u,i} \neq \bullet\}$ y $\bullet$ representa la falta de voto ($r_{u,i} = \bullet$ significa que el usuario u no ha votado por el ítem i). Según la literatura de RS las métricas más convencionales para evaluar la exactitud de los RS son MAE y RMSE.
- **Evaluación de la recomendación como conjunto:** La exactitud de un RS no asegura directamente confianza de los usuarios ante el conjunto de posibles predicciones. El usuario confía cuando está de acuerdo con un conjunto reducido de recomendaciones hechas por el RS [Bobadilla et al., 2013]. [Ricci et al., 2015c] A un usuario, la exactitud de las recomendaciones de un conjunto no le interesa, sino que la lista de ítems relevantes que son recomendados le resulten de interés. Las métricas comunes para evaluar la recomendación como conjunto, según [Hernández del Olmo and Gaudioso, 2008] son 1) Precisión, 2) Recall y 3) F1 Score. Todas ellas han sido adaptadas para RS a partir del área de “information retrieval”. La tabla 1.3 muestra dos tipos de recomendación: 1) la recomendación que ofrece el sistema y 2) la recomendación exitosa, la que el usuario acepta como tal; para el caso de los RS se suele utilizar un umbral de aceptación, puesto que la predicción no suele codificarse de manera binaria sino en el rango del voto

Tabla 1.2: Medidas de calidad para evaluación de la Predicción en RS

MAE	$MAE = \frac{1}{\#U} \sum_{u \in U} \left(\frac{1}{\#O_u} p_{u,i} - r_{u,i} \right)$	El error medio absoluto mide la diferencia absoluta entre las predicciones $p_{u,i}$ y los votos reales $r_{u,i}$.
NMAE	$NMAE = \frac{MAE}{r_{max} - r_{min}}$	El Error Medio Absoluto Normalizado toma en cuenta el rango de votos donde r_{max} indica el valor máximo de un voto y r_{min} el valor mínimo.
RMSE	$RMSE = \frac{1}{\#U} \sum_{u \in U} \sqrt{\frac{1}{\#O_u} (p_{u,i} - r_{u,i})^2}$	Computa la raíz del error medio cuadrado.
MSE	$MSE = \frac{1}{\#U} \sum_{u \in U} \frac{1}{\#O_u} (p_{u,i} - r_{u,i})^2$	Error medio cuadrado.
Coverage	$coverage = \frac{1}{\#U} \sum_{u \in U} \left(100 \times \frac{\#C_u}{\#D_u} \right)$	Calcula el porcentaje de situaciones en el que al menos un k-vecino de un usuario activo puede evaluar un ítem que no ha sido votado por el usuario activo.

Definido $K_{u,i}$ como el conjunto de vecinos del usuario u quienes han votado por el ítem i sea $C_u = \{i \in I | r_{u,i} = \bullet \wedge K_{u,i} \neq \emptyset\}$ y $D_u = \{i \in I | r_{u,i} = \bullet\}$.

máximo y mínimo de cada conjunto de datos para RS; a y d representan recomendaciones correctas (recomiendan un ítem relevante y no se recomienda un ítem no relevante), mientras que b y c son los casos incorrectos de recomendación. La tabla 1.4 presenta la formulación de las métricas para evaluar la recomendación como conjunto.

Tabla 1.3: Matriz de confusión de Information Retrieval adaptada a los RS.

	Recomendación exitosa	Recomendación no exitosa
Recomendado	a	b
No recomendado	c	d

- **Evaluación de la recomendación como listas jerárquicas:** [Bobadilla et al., 2013] En determinadas situaciones los sistemas de recomendación brindan extensas listas de ítems recomendados, en estos casos cobra importancia el orden de la recomendación puesto que es evidente que aquellos ítems que aparecen al inicio de la lista tienen más importancia

Tabla 1.4: Diferentes medidas de calidad para realizar la evaluación de la recomendación como conjunto.

Precisión	$precision = \frac{a}{a+b}$	Proporción de ítems recomendados relevantes respecto al número total de ítems.
Recall	$recall = \frac{a}{a+c}$	Proporción de ítems recomendados relevantes respecto al número total de ítems relevantes. Esta medida es una tasa de verdaderos positivos (true positive).
Fallout	$fallout = \frac{b}{b+d}$	Mide la tasa de falsos positivos(false positive).
F1	$F1 = \frac{2 \times precision \times recall}{precision + recall}$	Combina la precisión contra el recall.

Tabla 1.5: Medidas de calidad de en RS para la recomendación de listas jerárquicas

Half-life	$HL = \frac{1}{\#U} \sum_{u \in U} \sum_{i=1}^N \frac{\max(r_{u,p_i} - d, 0)}{2^{(i-1)/(\alpha-1)}}$	Estima que tan probable es que un usuario visite un ítem en una lista con ranking. Entiéndase por half-life el número de un ítem en la lista tal que existe el 50-50 de probabilidad que el usuario revise el ítem[Breese et al., 1998].
Donde d es el voto neutral y α es el viewing half-life.		
Discounted cumulative gain	$DCG^k = \frac{1}{\#U} \sum_{u \in U} \left(r_{u,p_1} + \sum_{i=2}^k \frac{r_{u,p_i}}{\log_2(i)} \right)$	Asume un decrecimiento del interés del usuario con curva logarítmica.
Donde k es el número de resultados encontrados que aparecen en el tope de la lista.		

para el usuario que aquellos que aparecen al final. Los errores incurridos en los ítems de inicio son más serios que los errores en los últimos ítems de las listas, por ello las métricas de ranking no se ocupan del accuracy, sino del orden en el que aparecen las listas. Las métricas de ranking o de listas jerárquicas consideran esta situación. Las métricas más utilizadas provienen del área de *information retrieval*. La tabla 1.5 presenta dos métricas comunes de la evaluación de la recomendación como listas jerárquicas.

- **Otras medidas de calidad de RS:** Incluso un ítem relevante recomendado con éxito

tiene poco valor para el usuario cuando es notorio o evidente. Ante este desafío se plantean métricas que van más allá de la exactitud, como la novedad y diversidad [Lü et al., 2012, Kaminskas et al., 2016].

- **Novelty (novedad):** en RS la novedad se refiere a que tan diferentes son los ítems recomendados con respecto a lo que los usuarios ya han visto antes, es decir se expresa recomendaciones que el usuario no conocía o no se esperaba.
- **Diversity (Diversidad):** se refiere a cuán diferentes son los ítems recomendados entre sí. Se puede considerar como lo opuesto a similaridad en el caso de los ítems, se utiliza principalmente para recomendar al usuario una lista de ítems diversos a los ítems que ha seleccionado.

En [Ricci et al., 2015c] se considera otras medidas de calidad en RS como: *Confidence* (*confidencia*) definida como la confianza en las predicciones de todo el sistema, es decir existe un nivel de confianza asociado al uso del RS. Otra aplicación de la confidencia, sobre todo en el filtrado colaborativo, es utilizar un umbral para la recomendación. *Trust* (*confianza en el voto*) representa la confianza que existe en la predicción para un voto específico, muchas de las veces la confianza en el voto provee una explicación de la recomendación. *Utility* (*Utilidad*) es una métrica para medir la perspectiva de la recomendación, por ejemplo un usuario puede desear recomendaciones en función del precio de los ítems, o una empresa puede desear realizar las recomendaciones en función de los ítems que mejores ingresos representan. *Risk* (*Riesgo*) es una métrica para asociar la recomendación con un riesgo potencial, es muy utilizado en sistemas de recomendación bursátiles, bancarios, etc. *Robustness* (*Robustez*) es la estabilidad de la recomendación ante la falta de información. Por ejemplo en ocasiones se trata de empeorar los sistemas de recomendación de la competencia mediante la inserción de usuarios vacíos que ocasionen problemas de inicio influenciando las recomendaciones del sistema. Finalmente, la *Serendipity* (*Serendipia*) es una métrica para medir cuan sorprendente es el éxito de una recomendación. Para [Herlocker et al., 2004] la serendipia se resume en dos componentes: sorpresa y relevancia. Novelty y Serendipity son dos métricas muy similares; para Herlocker et al. una recomendación con novedad puede no tener serendipia, pero una recomendación con serendipia necesariamente debe incluir ítems novedosos.

1.2. ¿Qué es clustering?

Agrupar objetos es una tarea requerida en diferentes áreas de la ingeniería, ciencia y tecnología; humanidades, medicina y la vida diaria [Saxena et al., 2017]. Un método de clustering identifica categorías o grupos de conjuntos finitos para describir los datos [Terán et al., 2018].

El clustering es un problema de clasificación no supervisada para descubrir agrupamientos naturales y reconocimiento de patrones [Ghazanfar and Prügel-Bennett, 2014, Witten et al., 2016]. Se fundamenta en agrupar objetos similares para crear un conjunto o clúster. El clustering generalmente es utilizado para reconocimiento de patrones en conjuntos de datos sin etiquetar. Puesto que los datos del conjunto están sin etiquetar, es difícil evaluar o cuantificar el rendimiento de este tipo de aprendizaje. Algunas de las principales áreas de aplicación del clustering son: el aprendizaje automático, bioinformática, análisis de imágenes, reconocimiento de patrones detección de valores atípicos [Ghosh and Dubey, 2013, Zahra et al., 2015]. Según [Jain, 2010] el clustering de datos se utiliza para tres propósitos principales:

- Estructura subyacente: para comprender mejor los datos, generar hipótesis, detectar anomalías e identificar características destacadas.
- Clasificación natural: para identificar el grado de similitud entre formas u organismos (relación filogenética).
- Compresión: como un método para organizar los datos y resumirlos a través de prototipos de clúster.

Son diversas las áreas y aplicaciones científicas que han utilizado técnicas de clustering, así mismo existe una gran cantidad de algoritmos publicados. Problemas de visión por computador como la segmentación de imágenes son abordados con técnicas de clustering en [Shi and Malik, 2000, Vishnuvarthanan et al., 2016]; jerarquización de tópicos en documentos para acceso eficiente a la información [Bhatia and Deogun, 1998, Djenouri et al., 2018]; segmentación de mercados y grupos de interés en el área de mercadotecnia [Vishnuvarthanan et al., 2016, Seret et al., 2014].

Existen dos enfoques de clustering según [Fraley and Raftery, 1998], los cuales son el jerárquico y el particional. El clustering jerárquico produce series anidadas de particiones y brinda como resultado un dendograma que muestra los patrones y diferentes niveles de similitud de los grupos [Bouguettaya et al., 2015]. Existen dos formas de clusterización jerárquica denominadas el método aglomerativo y el jerárquico divisivo [Murtagh, 1983].

- El método aglomerativo sigue un enfoque bottom-up, el cual comienza a construir clústers desde el objeto singular y empieza a mezclar estos clústers atómicos en clústers cada vez más grandes, hasta que los objetos finalmente quedan ubicados en un único clúster o caso contrario hasta que una condición de parada se cumpla.
- El método divisivo jerárquico sigue un enfoque top-down el cual comienza por disolver el clúster que contiene a todos los objetos en clústers cada vez más pequeños, hasta que cada objeto conforma un clúster o hasta que una condición de parada se cumpla.

Los métodos de clustering jerárquico pueden ser agrupados en tres categorías basadas en medidas de similitud o vínculos; según la literatura existen tres tipos de vínculos: single-linkage clustering, complete-linkage clustering y average-linkage clustering [Jain et al., 1999]. En single-linkage clustering el enlace entre dos clústers está conformado por una pareja de elementos, uno de cada clúster, que son los más cercanos. En complete-linkage clustering la distancia entre dos clústers está determinada por la mayor distancia entre algún miembro de un clúster y algún miembro del otro clúster. En average-linkage clustering la distancia entre dos clústers está determinada por la distancia promedio de algún miembro del un clúster y algún miembro del otro clúster.

Para [Hruschka et al., 2009, Ghazanfar et al., 2011] el clustering particional tiene por objetivo encontrar particiones individuales en lugar de numerosos grupos, como en el método jerárquico. Generalmente el segundo enfoque tiene la ventaja que puede ser aplicado a grandes conjuntos de datos para los cuales el dendograma no trabaja correctamente. Debido a este particular los RS se han enfocado en estrategias de clustering particional.

Según el tipo de partición que se desea construir, los enfoques utilizados en el clus-

tering particional son: *hard clustering* (clustering exclusivo) y *soft clustering* (clustering solapado)[Vilar et al., 2018].

1.2.1. Hard clustering

En hard clustering cada elemento (usuario o ítem) pertenece exactamente a un clúster [Ghosh and Dubey, 2013]. Los elementos son colocados en el clúster con el cual tienen el mayor nivel de similaridad [Wen and Celebi, 2011].

La técnica más famosa y bien conocida de hard clustering es el algoritmo k-means o hard c-means, que presenta las siguientes ventajas: a) Conceptualmente es simple, versátil y fácil de implementar; b) Presenta una complejidad lineal con respecto al número de elementos y clústers; y c) Se garantiza que el algoritmo termina con una tasa de convergencia cuadrática. El algoritmo 1 muestra la simplicidad del algoritmo k-means, donde *SeleccionCentroides* representa la selección inicial de centroides.

Algoritmo 1 k-means clustering

Input: X datos, k clústers, $iteraciones$ Máximo de Iteraciones

Output: C centroides, idx índices

```

1:  $C = \text{SeleccionCentroides}$ 
2:  $Iter = 0$ 
3: repeat
4:   for all muestra  $x_i$  do
5:     asignar  $idx_i$  al centroide más cercano  $C_j$ 
6:   end for
7:   asignar  $C_j$  al centro de masa de todos los usuarios en  $idx$ 
8:
```

$$c_j = \frac{1}{|idx| \quad \forall \quad idx = j} \sum_{X \in idx \quad \forall \quad idx = j} X$$

```

9:    $iter = iter + 1$ 
10: until ( $C$  no cambia) OR ( $Iter == Iteraciones$ )
11: return  $C, idx$ 
```

Existe una amplia gama de literatura sobre RS basados en k-means: En [Zahra et al., 2015] se proponen diferentes técnicas de inicialización de centroides para enfoques hard y soft clustering. Los resultados muestran que una selección efectiva de los centroides proporcionan beneficios de exactitud y rendimiento en la técnica k-means. [Degenmmis et al., 2007] utiliza un RS híbrido mediante la creación de particiones hard con k-means de los perfiles de los usuarios y utilizan dichas particiones como vecindario de usuarios similares y disimilares para realizar las predicciones. En [Sarwar et al., 2002] se utiliza el algoritmo bisecting k-means para mejorar la escalabilidad de los RS en línea; los resultados demuestran la eficiencia de crear vecindarios en base a clustering en lugar del uso de K-NN, a pesar de perjudicar la exactitud de las recomendaciones. [Zarzour et al., 2018] propone crear una matriz de similitudes a partir de la matriz de votos, posteriormente k-means clusteriza en un entorno denso y brinda los clústers de usuarios sobre los cuales se aplicará el proceso de predicción y recomendación.

1.2.2. Soft clustering

En soft clustering cada elemento puede pertenecer a dos o más clústers con diferentes grados de membresía a cada clúster [Wen and Celebi, 2011]. Los elementos que se ubican en los límites de varios clústers no pertenecen por completo a uno de los clústers, en su lugar se asignan grados de membresía entre 0 y 1 que indican el grado de pertenencia. La técnica más popular para soft clustering es fuzzy c-means, de hecho se comporta de manera similar al algoritmo k-means porque su estructura algorítmica es la misma. En principio, el soft clustering debería producir mejores resultados, pero los RS utilizan conjuntos de datos con alta dispersión, lo cual provoca que la asignación de un elemento a varios clústers no sea suficiente para producir mejores recomendaciones para los diferentes conjuntos de datos [Zahra et al., 2015]. De hecho [Zahra et al., 2015] proporciona un aporte fundamental: la calidad de clustering de fuzzy c-means también dependen de la selección inicial aleatoria de los centroides, y la matriz de membresía es más sensible que el enfoque hard clustering a la asignación inicial de los elementos a los clústers. Uno de los mayores problemas de fuzzy c-means es que puede producir recomendaciones inexactas, especialmente en grandes conjuntos de datos [Wu and Yang, 2002].

1.2.3. Métricas de calidad de clustering

La idea del clustering es separar a los usuarios o a los ítems en diferentes grupos en función de su similaridad. Pero el clustering es una tarea muy costosa computacionalmente (NP-Hard)[Arthur and Vassilvitskii, 2007], puesto que varios mínimos locales son posibles. Los algoritmos convencionales requieren una inicialización aleatoria de parámetros, lo que provoca que converjan en un mínimo local y por ende una calidad de clustering pobre.

La validación de clustering, consiste en evaluar que tan buenos son los resultados de los clústers conformados [Maulik and Bandyopadhyay, 2002]. La validación de clustering externa y la validación de clustering interna son las dos categorías principales de la validación de clustering. La principal diferencia es si la información externa se utiliza o no para la validación del clustering [Liu et al., 2010]. Un ejemplo de medida de validación externa es la entropía, que evalúa la pureza de los grupos en función de las etiquetas de clase dadas.

Las medidas de validación externa requieren información “externa” no presente en los datos, como las etiquetas de cada elemento y el número de clústers reales, por ello se utilizan principalmente para elegir el algoritmo de agrupamiento óptimo en un conjunto de datos específico [Wu et al., 2009]. En CF RS los conjuntos de datos no brindan este tipo de información, en principio se cuenta únicamente con los votos que los usuarios han realizado por determinados ítems, con un alto nivel de dispersión. En la práctica, la información externa, como las etiquetas de clase, a menudo no está disponible en muchos escenarios de aplicaciones, esta situación impide que medidas de evaluación externa de clustering puedan ser utilizadas para CF RS.

Las medidas de validación interna se basan en la información de los datos. Las medidas internas evalúan la bondad de una estructura de clustering sin considerar la información externa. Se utilizan para elegir el mejor algoritmo de clustering, así como el número de clústers óptimo sin ninguna información adicional. En el caso de CF RS, al no contar con

información externa disponible, las medidas de validación interna son la única opción para la validación del clustering.

En clustering particional se utilizan las siguientes métricas para evaluar la calidad del clustering:

- **Compactness o within-cluster:** Mide qué tan estrechamente relacionados están los objetos en un clúster.

$$J = \sum_{j=1}^n \sum_{k=1}^K u_{kj}^a \|x_j - c_k\|^2, \quad 1 \leq a \leq \infty \quad (1.1)$$

Donde:

c_k representa el centroide del grupo k .

n es el número de elementos en el clúster k

u_{kj} es la membresía de pertenencia del elemento x_j al clúster k . Para el caso de hard clustering la membresía vale 1 para un clúster exclusivo y 0 para el resto.

El parámetro a controla el nivel de solapamiento del usuario con respecto a los clústers.

Para el caso CF RS la distancia Euclidea no trabaja bien debido a la naturaleza dispersa de los conjuntos de datos de RS, por ello la ecuación 1.1 se ve reemplazada por:

$$J = \sum_{j=1}^n \sum_{k=1}^K u_{kj} \text{sim}(x_j - c_k) \quad (1.2)$$

Donde:

$\text{sim}(x_j - c_k)$ representa la similaridad de la correlación de Pearson entre el elemento x_j y el clúster k .

- **Separación:** Mide qué tan distinto o bien separado está un clúster de otros clústeres. Se utiliza la distancia mínima, debido a que permite seleccionar el peor de los casos (los clústers menos diferenciados)

$$s = (dmin)^2 \quad (1.3)$$

$$dmin = \min_{i,j, i \neq j} \|c_i - c_j\| \quad (1.4)$$

Donde:

$\|c_i - c_j\|$ representan la distancia Euclidea entre centroides.

- **Índice Xie-Beni:** Fue definido por Xie y Beni, representa la proporción entre el within-cluster y la separación.

$$XB = \frac{\sum_{j=1}^n \sum_{k=1}^K u_{kj}^2 \|x_j - c_k\|^2}{\min_{i,j, i \neq j} \|c_i - c_j\|^2} \quad (1.5)$$

La naturaleza de alta dispersión de los conjuntos de datos de RS dificulta que las medidas de calidad de clustering puedan ser utilizadas con métricas de distancia Euclidea, por lo que se han optado por soluciones como la de la ecuación 1.2. Debido a que las métricas de similaridad devuelven valores entre 0 y 1, la medida “separación” suele proporcionar resultados cercanos a cero, y esto provoca que otras métricas como el índice de Xie-Beni sean extremadamente sensibles a la separación. Por ello los trabajos de clustering en CF RS, generalmente muestran resultados de within-cluster.

Capítulo 2 Estado del Arte

El estado del arte se basa en MF debido a que es el método más utilizado en CF basado en modelos para RS. MF ofrece una alta calidad de las predicciones, reducción de dimensionalidad, lo cual mejora el rendimiento; y los aportes de NMF y técnicas Bayesianas han enriquecido los factores latentes, proporcionándoles un significado para explicar las predicciones. Esta tesis utiliza un modelo Bayesiano de NMF para ofrecer resultados de clustering a partir de la información colaborativa de los RS, por ello este capítulo se centra en las áreas de estudio específicas de la tesis y la referencias proporcionadas complementan el capítulo anterior, donde se repasa el estado del arte general del campo de los RS y del clustering.

El capítulo está organizado de la siguiente manera: la sección 2.1 contiene el estado del arte de la investigación en sistemas de recomendación basados en factorización matricial, la sección 2.2 detalla las investigaciones referentes a clustering en sistemas de recomendación. Finalmente la sección 2.3 expone la motivación, hipótesis y objetivos de la presente tesis.

2.1. Sistemas de Recomendación basados en Factorización Matricial

Los Sistemas de Recomendación(RS) basados en Filtrado Colaborativo(CF), en sus inicios principalmente utilizaban los métodos basados en memoria como el de los K-Vecinos (K Nearest Neighbours [K-NN]). Estos métodos han demostrado ser sensibles al grado de dispersión presentado por los conjuntos de datos de RS. Recientemente, los RS basados en modelos proveen resultados más precisos, más escalables y dirigen mejor el problema de la dispersión [Bobadilla et al., 2013]. El modelo más adoptado por los RS modernos es la Factorización Matricial (MF) [Koren et al., 2009], y recientemente su variación de la Factorización Matricial No Negativa (NMF). NMF fue inicialmente introducida y popularizada como una técnica de aproximación low-rank [Lee and Seung, 1999]. Es una de las herramientas de descomposición más típicas para extraer características clave de una matriz fuente. Recientemente, NMF se usa ampliamente en procesos de aprendizaje automático y ha sido aplicada en numerosas aplicaciones: Procesamiento de lenguaje [Arabnejad et al., 2017], recuperación de imágenes [Rad and Jamzad, 2017], computación biológica [Devarajan, 2008], redes de gran escala [Wang et al., 2016], señales de audio [Mirzaei et al., 2016], filtrado colaborativo [Zhang et al., 2017a], etc. MF es el método más utilizado para encontrar características ocultas, y adopta varios enfoques de baja dimensionalidad para reducir la complejidad computacional de los conjuntos de datos de RS que constan de usuarios e ítems [Huang et al., 2004].

Cada día emergen aplicaciones de RS, utilizando técnicas de factorización para proveer nuevas soluciones en el campo de CF o para mejorar los existentes [Bokde et al., 2015]. Los siguientes son una serie de estudios recientes utilizando MF o NMF en el campo del CF RS: [Li et al., 2017] propone un framework, basado en MF, para mejorar el uso de estructuras intrínsecas de la matriz de votos usuarios-ítems e información basada en contenido. [Ortega et al., 2016] reporta la MF basada en CF como la mejor opción para grupos de RS. [Zhang et al., 2017b] muestra que MF es una técnica robusta contra los ataques a los CF RS, mediante una variación a la función objetivo de NMF utilizando la norma R_1 en lugar de la diferencia cuadrada.[Kumar et al., 2017] usa técni-

cas de margen máximo MF cuando la matriz de votos contiene solo dos niveles (por ejemplo agrado/desagrado). [Baltrunas et al., 2011] utiliza MF en context-aware CF RS, específicamente utilizando conjuntos de datos con información geográfica y otros conjuntos de datos con información musical. [Friedman et al., 2016] dirigen el problema de la protección de privacidad del usuario; proveen un framework genérico para preservar la privacidad aprovechando MF y utilizando técnicas estadísticas de privacidad diferencial. [Zhang et al., 2017c] propone un modelo que provee un alto rendimiento en recomendación del cuidado de la salud. Utilizan el enfoque basado en topic model para descubrir la distribución de las preferencias del usuario e incorporan topic model y compensación emocional (descubierta por análisis de sentimientos) dentro de MF. [Zhang et al., 2016c] provee una múltiple MF no negativa con similaridad social para RS, considerando las similitudes entre usuarios, la relación de usuarios-recursos y la relación de etiquetas-recursos. [Su et al., 2016] presenta un novedoso enfoque de predicción QoS(calidad del servicio) híbrido del servicio web que sistemáticamente combina el CF basado en memoria y el CF basado en modelos, utilizando NMF y técnicas de Expectation-Maximization.

NMF es utilizada en un amplio número de aplicaciones como un método de aprendizaje automático. NMF es una técnica de separación de fuente ciega que descompone conjuntos de datos multivariante en factores no negativos significativos. Un enfoque realmente prometedor es incorporar soporte de estadística regular en el modelo NMF. En este sentido es posible asociar probabilidades con las predicciones y las recomendaciones, así como hacer uso de propiedades inherentes del enfoque probabilístico. En resumen, incorporar teoría Bayesiana en el modelo NMF enriquece sus resultados y también su interpretación.

Los siguientes estudios muestran el uso reciente del enfoque Bayesiano en el modelo de factorización NMF, y su aplicación al CF RS: [Zafari and Moser, 2017] propone dos novedosos modelos de factores latentes, incorporando discrepancias de valor de características con influencia social y discrepancias de valor de características condicionales de influencia social. Este método modela las preferencias del usuario como una red Bayesiana conside-

rando la influencia social de los amigos. [Hayashi and Watanabe, 2016] estudia el umbral real-log canonical y proporciona el límite superior del error de generalización en el aprendizaje Bayesiano. Resultados muestran que el error de generalización de MF puede hacerse más pequeño que los modelos estadísticos regulares si el aprendizaje Bayesiano es aplicado. Unicidad es un problema de NMF abierto y direccionado [Schachtner et al., 2014], donde los autores proponen un criterio de optimización Bayesiano para soluciones NMF que se pueden entregar en ausencia de conocimiento previo. [Sun et al., 2015] enfrenta los problemas de determinación del orden del modelo NMF; el orden del modelo determina la capacidad y exactitud de descubrir estructuras de datos. Los autores proponen un método basado en el modelo jerárquico de inferencia Bayesiana, criterio máximo a posteriori y un esquema de parámetros no informativos. [Salakhutdinov and Mnih, 2007] introduce una técnica MF probabilística linealmente escalable con el número de observaciones e incluye una adaptación previa sobre los parámetros del modelo.

2.2. Clustering en Sistemas de Recomendación

El Clustering de datos [Aggarwal and Reddy, 2013] presenta tres aspectos principales:

1. Métodos: Técnicas usadas comúnmente.
2. Dominios: Tipos de datos sin procesar (texto, multimedia, votos, flujos, biológicos, etc.)
3. Variaciones: Validación del agrupamiento, validación de clúster, conjuntos de clústers, etc.

La investigación en CF RS principalmente utiliza los datos de los votos como fuente de información. Los RS pueden incorporar una gran cantidad de información adicional a los votos (demográfica, basada en contenido, sensible al contexto, social, etc.); sin embargo, el denominador común para todo RS moderno es su matriz de votos. El clustering efectivo basado en la matriz de votos será útil para cualquier RS, y por lo tanto este enfoque puede ser considerado como universal. El tipo de datos (dominio) determina, en gran medida, el conjunto de métodos de clustering que mejor se ajustan a dicho dominio. Los principales tipos de clustering utilizados en el campo de RS son a) algoritmos basados en distancia y b) métodos de reducción dimensional. Métodos de selección de características y enfoques basados en densidad son métodos no muy utilizados en el campo de los RS

[Bobadilla et al., 2018].

Los siguientes son trabajos representativos en el campo de clustering en filtrado colaborativo: [Birtolo and Ronca, 2013] propone dos algoritmos de CF basados en clustering: Clustering difuso basado en ítems y trust-aware clustering; los dos métodos consiguen un valor mayor de coverage sin afectar la calidad de las recomendaciones. [Zhang et al., 2014] optimiza la MF estándar integrando el término de regularización del clúster del usuario; informan mejoras en la exactitud de las recomendaciones, comparado con los algoritmos estándar. Dimensionar el número de clústers (K) es un proceso que requiere experiencia, conocimiento de los datos y un mecanismo de prueba y error para elegir los valores más adecuados. La elección correcta de este parámetro determina la calidad de los clústers resultantes, así como las predicciones y recomendaciones hechas; [Wang and Zhang, 2014] establece dinámicamente el parámetro: con la llegada de más datos, el algoritmo de clustering incremental determina si aumentar el número de clústers o fusionar los clústers existentes. Los autores reportan una exactitud de las predicciones alentadora. [Hu et al., 2010] usan un método de co-cluster para dividir la matriz de datos sin procesar en clústers, y luego emplea NMF para hacer predicciones mejoradas de votos desconocidos.

Las preferencias de los usuarios cambian con el tiempo. Un método para modelar el cambio de preferencias esta basado en tomar en cuenta la evolución de características temporales. [Rana and Jain, 2014] propone un algoritmo de clustering evolutivo. Este algoritmo realiza una optimización de parámetros en conflicto en lugar de utilizar los algoritmos tradicionales como los algoritmos genéticos. [Salah et al., 2016] propone un sistema CF incremental eficiente, basado en un enfoque de clustering ponderado; este método provee un costo computacional muy bajo. Las recomendaciones de productos de comercio electrónico se pueden mejorar mediante clustering de productos similares [Liao and Lee, 2016]; el trabajo de recomendación es entonces realizado con los clústers resultantes. [Najafabadi et al., 2017] aplica registros de interacción implícita de los usuarios con los ítems para procesar de forma eficiente datos masivos mediante el empleo de minería de reglas de asociación. La técnica de clustering ha sido empleada para reducir tanto

el tamaño de los datos y dimensionalidad del espacio de ítems como el rendimiento de minería de reglas de asociación. [Chen et al., 2017] crea un modelo de red alimentado con ítems y usuarios considerados como individuos heterogéneos. De acuerdo al modelo de red construido, los estados de los individuos evolucionan con el tiempo. Individuos con las puntuaciones más altas se agrupan juntos e individuos con puntajes más bajos se disocian.

[Zahra et al., 2015] resalta cómo la selección de centroides en RS basados en k-means puede mejorar el rendimiento, así como ahorrar costos. El método de selección de centroides propuesto tiene la habilidad para explotar las estructuras de correlación de datos subyacentes. [Jain, 2010] provee un resumen de k-means. Resume los métodos existentes, discuten los mayores desafíos y señalan algunas de las investigaciones emergentes.

Para [Kim and Park, 2008] es posible formular el clustering como un problema de descomposición matricial. De acuerdo a [Aggarwal and Reddy, 2013, Zhang et al., 2016d], MF tiene ventajas importantes cuando es usada como un método de clustering: 1) Puede modelar distribuciones de datos muy variables debido a la flexibilidad de la factorización matricial [Buono and Pio, 2015, Yoo and Choi, 2009]; 2) Es capaz de hacer agrupaciones simultáneas de las filas (usuarios) y las columnas(ítems) de la matriz de votos de entrada; y 3) Se puede configurar para obtener clustering de tipo hard o soft [Buchta et al., 2012].

A continuación se presenta un conjunto de trabajos representativos y recientes centrados en el clustering basado en MF y NMF: [Buono and Pio, 2015] extrae información latente en alta dimensionalidad, usando 3-factores NMF como un método de co-clustering. En el trabajo descubren una estructura de correlación limpia entre los clústers de documentos y términos; esta estructura de correlación puede usarse como punto de partida para otros métodos de aprendizaje automático. El co-clustering ha sido propuesto en el campo de filtrado colaborativo para descubrir correlaciones subespaciales en entornos de big data [George and Merugu, 2005, Yang et al., 2002]. [Zhang et al., 2016d]

propone un framework de clustering restringido basado en NMF, en el cual existen restricciones en la forma de must-link y cannot-link (co-clustering semi supervisado) [Li et al., 2009]).

Cuando los conjuntos de datos incorporan alguna información de restricción específica, es posible fusionar tanto la información de la distribución de datos como la información de restricción para ser procesada usando un algoritmo de clustering restringido. Cuando los datos incorporan restricciones, los algoritmos de clustering restringido proporcionan un mejor rendimiento; sin embargo su uso no es universal, ya que solo tiene sentido aplicarlos en conjuntos de datos que cumplan con las restricciones. El campo NMF regularizado introduce restricciones adicionales a la formulación estandar de NMF, como las regularizaciones locales de aprendizaje basadas en vecindarios [Gu and Zhou, 2009]. El método de clustering propuesto en esta tesis no se basa en alguna restricción específica y no usa información adicional, por lo que puede ser usada en cualquier tipo de matriz de votos para CF. [Frémal and Lecron, 2017] crea un clustering basado en géneros, y entonces usa NMF para obtener predicciones y recomendaciones. El trabajo reporta mejoras tanto en predicción como en calidad de recomendación. Combinar cada resultado de clúster es el desafío ante el cual se enfrentan usando estrategias ponderadas.

2.3. Motivación e Hipótesis

Organizar datos en grupos sensibles es uno de los modos fundamentales de comprensión y aprendizaje [Jain, 2010]. El agrupamiento o clustering, se realiza de acuerdo a las características intrínsecas y a la similaridad entre los elementos. La investigación en el campo de CF RS clustering habitualmente se ha enfocado en el objetivo de detectar grupos de usuarios o grupos de ítems, para realizar un proceso diferenciado de CF en cada grupo de usuarios o ítems. Las ventajas que presentan los resultados de dichos estudios son: a) mejora de la exactitud, b) mejora del rendimiento (tiempo de ejecución) [Rana and Jain, 2014, Liao and Lee, 2016, Wang and Zhang, 2014, Hu et al., 2010, Hu et al., 2014]. Estos trabajos generalmente prueban la calidad de los clústers en forma indirecta: verificando la mejora de la calidad de las predicciones y las recomendaciones.

El uso de métodos de RS basados en modelos, específicamente MF y NMF, ha reducido la importancia del rendimiento del método de CF, debido a la velocidad de predicción del modelo cuando ya ha sido entrenado. Sin embargo, debido al incremento de la demanda de big data [George and Merugu, 2005, Yang et al., 2002, Chen et al., 2014], el proceso de RS clustering toma importancia en si mismo como una fuente de información de datos analíticos. En consecuencia, el proceso de CF clustering de matrices de votos es un campo cada vez más importante, no solo como un medio para mejorar la calidad de las recomendaciones, sino también para proveer analíticas confiables, que pueden incorporarse en acciones comerciales de valor agregado.

Las técnicas MF proporcionan muy buenos resultados de clustering cuando se aplican a datos con alto nivel de dispersión, como en el caso de los conjuntos de datos de RS [Yang et al., 2002, Zhang et al., 2016d, Li et al., 2009]. NMF puede modelar distribuciones de datos ampliamente variables debido a su flexibilidad en comparación a los rígidos clústers esféricos que la función objetivo del k-means intenta capturar. Cuando la distribución de datos está lejos de ser clúster esférico, el uso de NMF presenta ventajas. La escalabilidad de NMF se puede abordar de manera efectiva a través de diferentes esquemas: Shrinking, partitioning [Du et al., 2017], incremental [Zhang et al., 2016a] y paralelo [Kannan et al., 2016], haciendo posible mejorar el clustering de matrices de CF en entornos big data.

La investigación actual apunta a los métodos MF como los más adecuados para realizar clustering en conjuntos de datos dispersos, como lo son los RS. Sin embargo, asignar cada usuario (o ítem) a su clúster correspondiente no siempre es lo suficientemente preciso: el problema es que, a menudo, resulta demasiado restrictivo asignar un ítem a un solo clúster; por ejemplo: asignar una película western de humor al clúster de películas de humor o al clúster de películas del oeste. Los valores diferentes de los factores ocultos pueden ser demasiado similares; en tales situaciones no se puede tomar una decisión convincente sobre qué clúster asignar a cada usuario o ítem [Bobadilla et al., 2018].

La hipótesis de esta tesis es que la calidad de clusterign en RS mejorará si se utiliza NMF Bayesiano: este enfoque enriquece el modelo NMF, proporcionando una base probabilística. De esta manera, podremos usar los parámetros de aprendizaje del método para ajustar con precisión la asignación de cada usuario (o ítem) a cada clúster, y así mejorar la calidad del clustering del sistema. Además se espera que el uso de un innovador método de pre-clustering permita mejorar los resultados de calidad en los RS.

Según la revisión de la literatura, no hay documentos relevantes que proporcionen un estudio exhaustivo de CF clustering basado en NMF Bayesiano. Sin embargo, hay varios métodos Bayesianos NMF publicados [Schachtner et al., 2014, Sun et al., 2015, Lakshminarayanan et al., 2011, Luo et al., 2017, Virtanen et al., 2008].

[Hernando et al., 2016] es el estudio que esta tesis toma como referencia. Está basado en factorizar la matriz de votos en dos matrices no negativas cuyos componentes tienen un significado probabilístico comprensible.

Los objetivos de la presente tesis son:

- Presentar el estado del arte de clusterización en CF RS y de los métodos de factorización matricial Bayesianos, proporcionando una recopilación de los trabajos más recientes y representativos.
- Desarrollar un método de clustering basado en factorización matricial no negativa Bayesiana, incorporando una innovadora fase de pre-clustering para mejorar la calidad de las recomendaciones y el rendimiento de la fase de aprendizaje.
- Mejorar la calidad de clustering en RS, comparando los resultados obtenidos con los resultados que proporcionan varios métodos actuales y representativos.

Capítulo 3 Formalización

Este capítulo esta estructurado de la siguiente manera: La sección 3.1 expone el método de Factorización Matricial no Negativa Bayesiana (BNMF), el significado de los parámetros más importantes, la razón por la cual el clustering es correcto para RS y la formulación matemática que lo implementa. La sección 3.2 expone un ejemplo de ejecución para facilitar la comprensión del porqué se ha elegido el método propuesto y la utilidad del algoritmo de pre-clustering. La sección 3.3 se enfoca en pre-clustering: La selección inicial de centroides y la inicialización de los parámetros de aprendizaje BNMF.

3.1. Diseño del método y formulación

En esta sección se resume el modelo probabilístico discutido en [Hernando et al., 2016] y el algoritmo basado en este modelo para descubrir clústers de usuarios que comparten los mismos gustos.

El algoritmo para descubrir grupos está basado en un modelo probabilístico generativo; este modelo se muestra gráficamente en la figura 3.1: Los círculos representan variables aleatorias, las flechas entre dos variables indican dependencia entre ellas. Un círculo gris indica que el valor de esa variable aleatoria es la observada. Los pequeños cuadrados negros representan los parámetros del modelo.

Figura 3.1: Representación gráfica del modelo de enfoque probabilístico propuesto. Figura tomada de [Hernando et al., 2016].

El modelo probabilístico considera que los usuarios pueden agruparse en k conjuntos de acuerdo con sus gustos. Este modelo probabilístico permite predecir los votos que los usuarios hacen sobre los ítems mediante el siguiente procedimiento:

1. Para cada usuario u , se muestrea una vector como variable aleatoria de dimensión K , ϕ_u a partir de la distribución de Dirichlet.

$$\vec{\phi}_u \sim \text{Dir}(\alpha_1, \dots, \alpha_K) \quad (3.1)$$

- El vector $(\phi_{u,1}, \dots, \phi_{u,K})$ representa la probabilidad que un usuario pertenezca a cada clúster.

- α es un parámetro del modelo. Indica el conocimiento a priori del grado de solapamiento entre clústers.
- 2. Para cada ítem i y cada factor k , se muestrea una variable aleatoria $K_{i,k}$ a partir de una distribución Beta (que toma valores de 0 a 1):

$$K_{i,k} \sim \text{Beta}(\beta, \beta) \quad (3.2)$$

- El valor $K_{i,k}$ representa la probabilidad de que a los usuarios en el clúster k les guste el ítem i .
- β es un parámetro del modelo que indica la cantidad de evidencia que el algoritmo requiere para deducir que a un grupo de usuarios les gusta un ítem.
- 3. Para el usuario u y el ítem i tal que $r_{u,i} \neq \bullet$ (el punto significa no votado), se muestrean las siguientes variables aleatorias:

- La variable aleatoria $z_{u,i}$ a partir de una distribución categórica (que toma valores en el rango $\{1, \dots, K\}$).

$$z_{u,i} \sim \text{Cat}(\vec{\phi}_u) \quad (3.3)$$

- Un valor k en $z_{u,i}$ representa que el usuario u se comporta como los usuarios en el clúster k cuando consumen el ítem i .
- La variable aleatoria $\rho_{u,i}$ a partir de una distribución Binomial.

$$\rho_{u,i} \sim \text{Bin}(R, K_{i,z_{u,i}}) \quad (3.4)$$

El parámetro R está altamente relacionado con los votos que el usuario u ha hecho sobre el ítem i . De hecho, el voto normalizado que el usuario u ha hecho sobre el ítem i es $\rho_{u,i}/R$.

$$R = \text{Max}_{\text{RatingRange}} - \text{Interval}_{\text{RatingRange}} \quad (3.5)$$

Por ejemplo: utilizando MovieLens y Netflix: $R = 5 - 1 = 4$, mientras que usando FilmTrust: $R = 4 - 0,5 = 3,5$ (Los votos de FilmTrust van desde 0.5 a 4 con salto de 0.5; los votos de MovieLens & Netflix van desde 1 a 5, con salto de 1).

Siguiendo este procedimiento podemos simular muestras de votos plausibles. Se observa el valor de las variables aleatorias $\rho_{u,i}$ (círculo gris en la figura 3.1). El resto de variables (círculos blancos en la figura 3.1) son conocidas por nosotros. Una importante cuestión sobre el modelo consiste en descubrir la distribución posterior de las variables aleatorias desconocidas $\vec{\phi}_u$ (las cuales a veces son denominadas variables latentes), indicando la probabilidad posterior de que el usuario pertenezca a cada clúster. Como casi todos los modelos

gráficos probabilísticos, el modelo considerado no permite calcular computacionalmente las distribuciones condicionales exactas de una manera analítica. Un algoritmo basado en la técnica de inferencia variacional es adoptado para calcular la distribución de probabilidad posterior. La descripción del algoritmo como una caja negra es:

- **Entrada.** La entrada del algoritmo es una matriz de votos, ya que el proceso se enfoca en CF RS. Se utilizará la siguiente notación relacionada a la matriz de votos:
 - N : Número de usuarios en la matriz de votos.
 - M : Número de ítems en la matriz de votos.
 - $r_{u,i}$: Voto que el usuario u ha hecho sobre el ítem i . Para que el algoritmo pueda usarse para cualquier RS con diferentes escalas, consideraremos los votos normalizados $r_{u,i}$ que se encuentran dentro de $[0, 1]$. Se utiliza la notación $r_{u,i} = \bullet$ para indicar que el usuario u no ha votado todavía por el ítem i .
 - **Parámetros.** Además de la matriz de votos, la técnica también toma en cuenta algunos parámetros fijados por el RS. Específicamente considera los siguientes parámetros:
 - K . Este parámetro indica el número de grupos de usuarios (clústers) que el algoritmo va a descubrir.
 - $\alpha \in (0,1)$. Este parámetro está relacionado a la posibilidad de obtener solapamiento de grupos de usuarios que comparten los mismos gustos. Un valor cercano a 0 implica que los usuarios tienden a pertenecer únicamente a un grupo. Cuanto mayor sea el valor de α , mayor será la flexibilidad del método Bayesiano para asignar a un mismo usuario varios grupos diferentes (de manera probabilística).
 - $\beta > 1$: Este parámetro está relacionado con la cantidad de evidencia que el algoritmo requiere para deducir que a un grupo de usuarios le gusta un ítem. Un valor alto de β significa que el modelo requiere más evidencia para deducir que a un grupo de usuarios le agrada/desagrada un ítem.
 - **Salida.** A partir de los datos de entrada y los valores de los parámetros, el algoritmo genera una $N \times K$ matriz $(a_{u,k})$ asociada a los usuarios. Cada término $a_{u,k}$ de la matriz informa sobre la probabilidad de que el usuario u pertenezca a cada clúster k de usuarios. El algoritmo calcula algunas variables auxiliares $\gamma_{u,k}$, $\epsilon_{i,k}^-$, $\epsilon_{i,k}^+$ para finalmente calcular los valores $a_{u,k}$.

Pasos:

- Inicializar aleatoriamente $\gamma_{u,k}$
- Inicializar aleatoriamente $\epsilon_{i,k}^+$
- Inicializar aleatoriamente $\epsilon_{i,k}^-$
- Repetir hasta que los cambios no sean significativos

Por cada usuario u :

Por cada ítem i votado por el usuario u en el conjunto de entrenamiento:

Por cada factor k : actualizar $\lambda_{u,i,k}$ de acuerdo a las ecuaciones:

$$\lambda'_{u,i,k} = \exp\left(\Psi(\gamma_{u,k}) + r_{u,i}^+ \cdot \Psi(\epsilon_{i,k}^+) + r_{u,i}^- \cdot \Psi(\epsilon_{i,k}^-) - R \cdot \Psi(\epsilon_{i,k}^+ + \epsilon_{i,k}^-)\right) \quad (3.6)$$

$$\lambda_{u,i,k} = \frac{\lambda'_{u,i,k}}{\lambda'_{u,i,1} + \dots + \lambda'_{u,i,K}} \quad (3.7)$$

Por cada usuario u :

Por cada factor k : actualizar $\gamma_{u,k}$ de acuerdo a la ecuación:

$$\gamma_{u,k} = \alpha + \sum_{\{i|r_{u,i} \neq \bullet\}} \lambda_{u,i,k} \quad (3.8)$$

Por cada ítem i votado por el usuario u en el conjunto de entrenamiento:

Por cada factor k : actualizar $\epsilon_{i,k}^+$ de acuerdo a la ecuación:

$$\epsilon_{i,k}^+ = \beta + \sum_{\{u|r_{u,i} \neq \bullet\}} \lambda_{u,i,k} \cdot r_{u,i}^+ \quad (3.9)$$

Por cada factor k : actualizar $\epsilon_{i,k}^-$ de acuerdo a la ecuación:

$$\epsilon_{i,k}^- = \beta + \sum_{\{u|r_{u,i} \neq \bullet\}} \lambda_{u,i,k} \cdot r_{u,i}^- \quad (3.10)$$

Donde

$$r_{u,i}^+ = \rho_{u,i} = R * r_{u,i} \quad (3.11)$$

$$r_{u,i}^- = R - \rho_{u,i} \quad (3.12)$$

$$R = \max RatingRange - 1, r_{u,i}^+ \ \& \ r_{u,i}^- \in \{\min RatingRange - 1, \dots, \max RatingRange - 1\}$$

Por ejemplo: para conjuntos de datos donde los votos pertenecen a $\{1, 2, 3, 4, 5\}$:
 $\min_{RatingRange} = 1, \max_{RatingRange} = 5, r_{u,i}^+ \quad \& \quad r_{u,i}^- \in \{0, 1, 2, 3, 4\}, R = 4.$

Salida $a_{u,k}$:

$$a_{u,k} = \frac{\gamma_{u,k}}{\gamma_{u,1} + \dots + \gamma_{u,K}} \quad (3.13)$$

Salida $b_{k,i}$:

$$b_{k,i} = \frac{\epsilon_{i,k}^+}{\epsilon_{i,1}^+ + \dots + \epsilon_{i,K}^+} \quad (3.14)$$

Es necesario resaltar que este método probabilístico cumple una importante restricción sobre los factores ocultos de los usuarios:

$$\sum_{k=1}^K a_{u,k} = 1 \quad (3.15)$$

Para facilitar el análisis, se ha dividido el proceso BNMF en cuatro algoritmos:

1. Inicialización BNMF (Algoritmo 2).
2. Entrenamiento BNMF (Algoritmo 3).
3. Evaluación de la predicción de BNMF (Algoritmo 4).
4. Clustering de usuarios BNMF (Algoritmo 5).

Algoritmo 2 Inicialización de parámetros BNMF

Input: $\mathbb{M}_{train}$ votos de entrenamiento, k cantidad de factores

Output: $\gamma_{u,k}, \epsilon_{i,k}^+, \epsilon_{i,k}^-$

- 1: Normalizar $\mathbb{M}_{train}$ en $r_{u,i}^+$ y $r_{u,i}^-$ con las ecuaciones 3.11 y 3.12
 - 2: Inicializar aleatoriamente $\gamma_{u,k}$
 - 3: Inicializar aleatoriamente $\epsilon_{i,k}^+$ y $\epsilon_{i,k}^-$
 - 4: **return** $\gamma_{u,k}, \epsilon_{i,k}^+$ y $\epsilon_{i,k}^-$
-

Algoritmo 3 Entrenamiento BNMF

Input: $\mathbb{M}_{train}$ votos de entrenamiento, k clústers, α solapamiento, β evidencia, $iteraciones$ Máximo de iteraciones

Output: $a_{u,k}$ matriz de membresía de usuarios, $b_{k,i}$ matriz de pertenencia de los ítems a los grupos

```

1: Call Inicialización de parámetros BNMF
2:  $Iter = 0$ 
3: repeat
4:   for all usuario  $u \in \mathbb{M}_{train}$  do
5:     for all ítem  $i \in \mathbb{M}_{train}$  do
6:       for all factor  $k$  do
7:         actualizar  $\lambda_{u,i,k}$ 
8:       end for
9:     end for
10:   end for
11:   for all usuario  $u \in \mathbb{M}_{train}$  do
12:     for all factor  $k$  do
13:       actualizar  $\lambda_{u,i,k}$ 
14:     end for
15:   end for
16:   for all ítem  $i \in \mathbb{M}_{train}$  do
17:     for all factor  $k$  do
18:       actualizar  $\epsilon_{i,k}^+, \epsilon_{i,k}^-$ 
19:     end for
20:   end for
21:    $iter = iter + 1$ 
22: until ( $\epsilon_{i,k}^+$  no cambia) OR ( $Iter == Iteraciones$ )
23: actualizar  $a_{u,k}$  con la ecuación 3.13
24: actualizar  $b_{k,i}$  con la ecuación 3.14
25: return  $a_{u,k}, b_{k,i}$ 

```

Algoritmo 4 Predicciones BNMF

Input: $\mathbb{M}_{test}$ Matriz de votos de test, $a_{u,k}$ matriz de membresía de usuarios, $b_{k,i}$ matriz de pertenencia de los ítems a los grupos, K cantidad de clústers

Output: MAE exactitud de las predicciones

- 1: $MAE = 0$
- 2: **for all** usuario $u \in \mathbb{M}_{test}$ **do**
- 3: **for all** ítem $i \in \mathbb{M}_{test}$ **do**
- 4: calcular la predicción $p_{u,i}$

$$p_{u,i} = \sum_{k=1 \dots K} a_{u,k} \cdot b_{k,i} \quad (3.16)$$

- 5: desnormalizar $p_{u,i}$ al rango de votos del RS para obtener $prediccion_{u,i}$. Para el caso de Netflix y MovieLens:

$$prediccion_{u,i} = \left\{ \begin{array}{l} 1, \text{si } 0 \leq p_{u,i} < 0,2 \\ 2, \text{si } 0,2 \leq p_{u,i} < 0,4 \\ 3, \text{si } 0,4 \leq p_{u,i} < 0,6 \\ 4, \text{si } 0,6 \leq p_{u,i} < 0,8 \\ 5, \text{si } 0,8 \leq p_{u,i} \leq 1 \end{array} \right\}$$

- 6: $MAE = MAE + |\mathbb{M}_{test} - prediccion_{u,i}|$
- 7: **end for**
- 8: **end for**
- 9: $MAE = MAE / (\#\mathbb{M}_{test})$
- 10: **return** MAE

Algoritmo 5 Clustering de usuarios BNMF

Input: $\mathbb{M}_{train}$ matriz de votos de entrenamiento, k clústers, $a_{u,k}$ matriz de membresía de usuarios

Output: idx índice de pertenencia de cada usuario a su clúster, *Within-cluster* calidad de clustering

```

1: Within-cluster = 0
2: for all usuario  $u \in \mathbb{M}_{train}$  do
3:   asignar  $idx_u$  según el valor de membresía de la matriz  $a_{u,k}$ 
4: end for
5: calcular Within-cluster según la ecuación 1.2
6: return Within-cluster

```

3.2. Ejemplo de ejecución (Running Example)

En esta sección se intenta demostrar, de una manera simple e intuitiva, la operación del método propuesto. También se explica, con ejemplos, las mejoras realizadas a los métodos existentes más importantes. Se emplea un ejemplo de ejecución didáctica. Este ejemplo se basa en un datatoy diseñado para mostrar las variaciones de calidad de clustering obtenidas en cada caso de estudio. El datatoy original se basa en la matriz de votos (12 usuarios, 12 ítems) representada en la figura 3.2: En el datatoy podemos observar, intuitivamente, la existencia de cuatro grupos; $G1 = U_1, U_2, U_3$, $G2 = U_4, U_5, U_6$, $G3 = U_7, U_8, U_9$, $G4 = U_{10}, U_{11}, U_{12}$. Usuarios del $G1$ comparten gustos similares por los ítems I_1, I_2, I_3 ya que $r_{u,i} > 4$. U_1 y U_8 tiene un gusto similar para el ítem 11; sin embargo, U_8 no pertenece a $G1$ porque no le agrada I_2 . El mismo razonamiento puede utilizarse para explicar la existencia de $G2, G3$ y $G4$. En la sección 3.2.1, se utilizarán varios métodos de clustering para probar su capacidad de detectar los cuatro grupos en cuestión.

El mayor problema en gran cantidad de métodos de clustering es su sensibilidad a la inicialización de parámetros y la selección inicial de los centroides. Una selección pobre de los parámetros de pre-clustering y centroides puede conducir a un mínimo local, baja calidad de clustering e insuficiente calidad de las predicciones. Esta tesis aborda mejoras tanto en clustering como en pre-clustering.

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}
U_1	4	5			4			1		4		
U_2	5	5	5		5			1				4
U_3	4	4	5									
U_4		1		1	1	1		1			5	
U_5	2			2		2			2			4
U_6				1	1	1						
U_7	1						4	4	5			
U_8		2			4		5	5	5	4	5	
U_9				5			4		4			
U_{10}										1	2	2
U_{11}	1		1		4		1			2	1	
U_{12}		1				5			2	2	2	1

Figura 3.2: Matriz de votos del datatoy. Figura tomada de [Bobadilla et al., 2018].

3.2.1. K-means y pre-clustering k-means (k-means+)

La figura 3.3 muestra un resultado obtenido utilizando el método k-means. La columna “cluster” muestra el clúster en el cual cada usuario ha sido clasificado:

- $G1 = U_1, U_2, U_3, U_6, U_{10}, U_{11}, U_{12}$
- $G2 = U_4, U_5, U_8$
- $G3 = U_7$
- $G4 = U_9$

Los usuarios no han sido clasificados como se esperaba. Las predicciones $p_{1,1}$, $p_{9,8}$ y $p_{11,12}$ son correctas, pero la predicción $p_{5,5}$ se muestra como “indiferente” (3), un valor que se esperaba como “desagrado” (1 o 2). Los resultados obtenidos varían considerablemente de acuerdo a cual es la inicialización de centroides del método k-means que ha sido ejecutado.

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}	cluster
U_1	4	4	5			4			1		4		1
U_2	5	5	5		5			1				4	1
U_3	4	4	5										1
U_4		1		1	1	1		1			5		2
U_5	2			2	3	2			2			4	2
U_6				1	1	1							1
U_7	1						4	4	5				3
U_8		2			4		5	5	5	4	5		2
U_9				5			4	4	4				4
U_{10}							2			1	2	2	1
U_{11}	1		1		4			1		2	1	2	1
U_{12}		1				5			2	2	2	1	1

Figura 3.3: Clustering k-means y resultados de la predicción. Figura tomada de [Bobadilla et al., 2018].

La figura 3.4 muestra el resultado obtenido utilizando clustering k-means y pre-clustering “Log power-users”[Zahra et al., 2015]. Los usuarios que han votado por un gran número de ítems son referenciados como “power users”. [Zahra et al., 2015] utilizan el power user para seleccionar los centroides del pre-clustering, obteniendo buenos resultados en RS. La versión “Log” power users está basada en una función de probabilidad específica para seleccionar los centroides. [Zahra et al., 2015] presentan tres versiones de probabilidad: “Power”, “ProbPower” and “Log Power”, la última versión es la que consigue los mejores resultados.

La figura 3.4 muestra una mejora en la selección de clústers: ahora, el clúster 4 está perfectamente constituido. Sin embargo, la calidad de la exactitud todavía es pobre, y algunos resultados son incorrectos (por ejemplo: $p_{11,12}$). Los métodos de clustering tradicional tienen poca precisión cuando son aplicados a conjuntos de datos extremadamente dispersos, tal como lo son en los RS.

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}	cluster
U_1	4	4	5			4			1		4		4
U_2	5	5	5		5			1				4	4
U_3	4	4	5										4
U_4		1		1	1	1		1			5		1
U_5	2			2	4	2			2			4	2
U_6				1	1	1							1
U_7	1						4	4	5				1
U_8		2			4		5	5	5	4	5		3
U_9				5			4	5	4				3
U_{10}							4			1	2	2	4
U_{11}	1		1		4			1		2	1	4	2
U_{12}		1				5			2	2	2	1	3

Figura 3.4: Clustering k-means con pre-clustering Log power-users (k-means+). Figura tomada de [Bobadilla et al., 2018].

3.2.2. Factorización Matricial

Los RS han evolucionado a partir de los métodos basados en memoria, como el K-NN, al uso de métodos basados en modelos, como la Factorización Matricial (MF). Los métodos MF proveen mejores resultados de predicción y recomendación, coverage total y escalabilidad. Los métodos MF también proveen resultados de clustering, mediante el análisis de sus factores ocultos.

La figura 3.5a) muestra los factores ocultos de los usuarios en una MF de dimensión $K=4$. Se asigna cada usuario al clúster cuyo valor oculto (F_1, F_2, F_3, F_4) es el más alto; por ejemplo: U_1 ha sido asignado al clúster 3, porque su valor oculto 1.87 es mayor que sus otros tres valores ocultos (0.87, 0.75, 0.04). U_4 ha sido asignado al clúster 4, por su valor oculto 2.07, mayor que -0.33, 0.47 y 0.06, y así sucesivamente.

a)

	F1	F2	F3	F4
U_1	0.8766	0.7546	1.878	0.049
U_2	0.9589	1.3451	1.84	-0.234
U_3	1.1819	1.2768	1.349	0.751
U_4	2.0721	-0.3327	0.479	0.064
U_5	1.5901	0.501	0.138	0.27
U_6	0.3199	0.2788	0.231	0.108
U_7	0.6568	1.1921	0.163	2.058
U_8	0.8138	0.7968	0.758	2.477
U_9	1.0992	1.2825	1.024	1.352
U_{10}	0.6892	0.3393	0.282	0.14
U_{11}	-0.1735	1.7502	0.216	0.442
U_{12}	-0.0883	0.1962	0.825	1.419

b)

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	cluster
U_1	5	4	5			4			1		4		3
U_2	5	5	5		5			1					3
U_3	4	4	5										3
U_4		1		1	1	1		1				5	1
U_5	2			2	2	2			2			4	1
U_6				1	1	1							1
U_7	1						4	4	5				4
U_8		2			4		5	5	5	4	5		4
U_9				5			4	3	4				4
U_{10}							1			1	2	2	1
U_{11}	1		1		4			1		2	1	1	2
U_{12}		1				5			2	2	2	1	4

Figura 3.5: Clustering de factorización matricial y resultados de predicción: a) Matriz de factores de los usuarios, b) Resultados de clustering. Figura tomada de [Bobadilla et al., 2018].

La figura 3.5b) resume la asignación de los usuarios a los clústers (columna derecha) y también muestra algunas predicciones relevantes. Se puede observar una mejora significativa en los resultados de clustering con respecto a las soluciones de k-means+: en este caso los usuarios de U_1 a U_9 se han clasificado correctamente (U_{10} y U_{12} no lo están). Las predicciones son correctas, excepto para $p_{9,8}$ el cual muestra un valor “indiferente”(3) en lugar de que le agrada (4 o 5). En general, el método clustering de MF mejora al k-means+ cuando se aplica a conjuntos de datos dispersos; sin embargo, existe margen de mejora en los resultados de clustering y predicción.

3.2.3. Factorización Matricial no Negativa Bayesiana (BNMF)

La versión Bayesiana de NMF propuesta [Hernando et al., 2016] mejora el método existente NMF y proporciona una base probabilística que clasifica cada usuario o ítem entre los factores de la factorización k . Se espera que este enfoque probabilístico mejore tanto resultados de clustering de MF y NMF.

La figura 3.6 muestra el resultado del ejemplo de ejecución utilizando Factorización Matricial no Negativa Bayesiana (BNMF). Como se puede ver en el gráfico a), los resultados son correctos: cada usuario es asignado a su grupo de clustering. El gráfico b) muestra los grupos de clustering y las predicciones: Las predicciones (u_1, i_1) , (u_5, i_5) , (u_9, i_8) son precisas, sin embargo la predicción (u_{11}, i_{12}) no lo es.

Figura 3.6: Clustering de la Factorización Matricial no Negativa Bayesiana y resultados de predicción: a) Matriz de factores de los usuarios, b) Resultados de clustering. Figura tomada de [Bobadilla et al., 2018].

3.2.4. Pre-Clustering y Factorización matricial no Negativa Bayesiana (BNMF+)

El pre-clustering es utilizado para mejorar tanto la exactitud del clustering como el rendimiento del sistema. En esta tesis se utiliza la selección de centroides k-meansPlus

LogPower [Zahra et al., 2015] para inicializar el aprendizaje de los parámetros del método BNMF. Se denomina BNMF+ a la secuencia: 1) selección de centroides, y 2) NMF Bayesiana. La sección 3.3 explica los detalles del algoritmo propuesto de pre-clustering y su personalización al método BNMF.

Para ilustrar las ventajas del pre-clustering, se ha agregado un usuario (u_{13}) a la matriz de votos del ejemplo de ejecución, creando un clustering un poco más complicado (Figura 3.7).

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}
u_1	4	5				4			1		4	
u_2	5	5	5		5			1				4
u_3	4	4	5									
u_4		1		1	1	1		1			5	
u_5	2			2		2			2			4
u_6				1	1	1						
u_7	1						4	4	5			
u_8		2			4		5	5	5	4	5	
u_9				5			4		4			
u_{10}										1	2	2
u_{11}	1		1		4			1		2	1	
u_{12}		1				5			2	2	2	1
u_{13}	5	5	5				1	5	5			

Figura 3.7: Ejemplo de ejecución extendido de la matriz de votos. Figura tomada de [Bobadilla et al., 2018].

La tabla superior de la figura 3.8 muestra los resultados de tres ejecuciones diferentes del método BNMF. Estas ejecuciones no usan ningún tipo de pre-clustering. Por lo general, se obtiene un resultado muy preciso, como se muestra en “Experiment 1”. Sin embargo, en algunos casos, el método se basa en valores iniciales, tomados aleatoriamente, que conducen a soluciones inadecuadas. Este es el caso de los experimentos etiquetados como “Experiment 2” y “Experiment 3” donde el algoritmo ha alcanzado los mínimos locales.

Si utilizamos pre-clustering, podemos reducir la probabilidad de encontrar soluciones inadecuadas y mínimos locales. El método de pre-clustering elegido utiliza una medida de similitud para calcular las distancias entre los ítems del RS (o usuarios), y así determinar cuáles son los centroides iniciales más adecuados. Por lo general, la medida de similitud elegida es la distancia Euclídea, sin embargo, en los conjuntos de datos RS, es más apropiado utilizar medidas de similitud adaptadas a matrices dispersas

[Bobadilla and Serradilla, 2009]. [Sun and Dong, 2017] profundizan y detallan diferentes cuestiones de porqué las medidas de similaridad se comportan mejor a las medidas basadas en distancia en conjuntos de datos de RS

La tabla inferior de la figura 3.8 muestra los resultados de utilizar pre-clustering para seleccionar los centroides y determinar los valores iniciales de los parámetros que alimentan el algoritmo iterativo BNMF (Algoritmo 3). El lado izquierdo muestra los resultados usando la medida de similaridad Mean Jaccard Difference (MJD) [Bobadilla et al., 2010], la posición central muestra los resultados obtenidos usando JMSD y el lado derecho muestra los resultados de la correlación de Pearson. Se puede ver que, en el ejemplo, MJD [Bobadilla et al., 2012] produce los resultados esperados, mientras que JMSD [Bobadilla et al., 2010] y la correlación de Pearson no lo hacen. En general, al utilizar pre-clustering, seremos capaces de minimizar las situaciones en las que obtenemos resultados de clustering inadecuados. Como se verá más adelante, en esta tesis, el pre-clustering que hemos diseñado mejora la calidad de clustering y disminuye los tiempos de ejecución de la factorización matricial.

3.3. Pre-clustering de Factorización Matricial no Negativa Bayesiana (BNMF+)

Los métodos básicos de clustering comienzan con la selección de los centroides iniciales o con la selección de los parámetros iniciales, tomados aleatoriamente. A partir de ellos, mediante un proceso iterativo, se modifican los centroides y los clústers. El proceso de aprendizaje automático finaliza cuando se alcanza una condición de parada o cuando se cumple el número máximo de iteraciones.

Los resultados de calidad de clustering dependen, en gran medida, de la elección de los centroides o los parámetros iniciales con los que se alimenta el proceso iterativo. Una mala elección de los valores iniciales puede llevar al algoritmo a un mínimo local. Por esta razón, hay varios enfoques basados en RS para elegir los centroides iniciales en lugar de una selección aleatoria; por ejemplo: k-means-, k-means+, k-means++, k-meansUserPower, etc. [Zahra et al., 2015].

BNMF without pre-clustering														
Experiment 1					Experiment 2					Experiment 3				
F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster
0.019	0.944	0.019	0.019	2	0.02	0.02	0.71	0.25	3	0.94	0.02	0.02	0.02	1
0.016	0.953	0.016	0.016	2	0.02	0.19	0.78	0.02	3	0.95	0.02	0.02	0.02	1
0.029	0.912	0.029	0.029	2	0.03	0.03	0.91	0.03	3	0.91	0.03	0.03	0.03	1
0.016	0.016	0.953	0.016	3	0.02	0.02	0.02	0.95	4	0.17	0.79	0.02	0.02	2
0.019	0.019	0.944	0.019	3	0.02	0.02	0.02	0.94	4	0.02	0.02	0.02	0.94	4
0.029	0.029	0.912	0.029	3	0.03	0.03	0.03	0.91	4	0.03	0.91	0.03	0.03	2
0.932	0.023	0.023	0.023	1	0.93	0.02	0.02	0.02	1	0.02	0.93	0.02	0.02	2
0.959	0.014	0.014	0.014	1	0.46	0.01	0.51	0.01	3	0.01	0.96	0.01	0.01	2
0.912	0.029	0.029	0.029	1	0.91	0.03	0.03	0.03	1	0.03	0.03	0.91	0.03	3
0.029	0.029	0.029	0.912	4	0.91	0.03	0.03	0.03	1	0.91	0.03	0.03	0.03	1
0.016	0.016	0.016	0.953	4	0.02	0.95	0.02	0.02	2	0.59	0.38	0.02	0.02	1
0.016	0.016	0.016	0.953	4	0.03	0.94	0.02	0.02	2	0.02	0.02	0.95	0.02	3
0.331	0.637	0.016	0.016	2	0.02	0.02	0.95	0.02	3	0.64	0.33	0.02	0.02	1

BNMF with pre-clustering (BNMF+)														
MJD					JMSD					PEARSON				
F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster	F1	F2	F3	F4	cluster
0.019	0.944	0.019	0.019	2	0.02	0.02	0.02	0.94	4	0.93	0.02	0.02	0.02	1
0.016	0.953	0.016	0.016	2	0.02	0.02	0.02	0.95	4	0.53	0.02	0.02	0.43	1
0.029	0.912	0.029	0.029	2	0.03	0.03	0.03	0.91	4	0.87	0.04	0.04	0.04	1
0.016	0.016	0.953	0.016	3	0.19	0.02	0.77	0.02	3	0.02	0.02	0.02	0.93	4
0.019	0.019	0.944	0.019	3	0.79	0.02	0.17	0.02	1	0.02	0.93	0.02	0.03	2
0.029	0.029	0.912	0.029	3	0.03	0.03	0.91	0.03	3	0.04	0.87	0.04	0.04	2
0.932	0.023	0.023	0.023	1	0.02	0.02	0.93	0.02	3	0.91	0.03	0.03	0.03	1
0.959	0.014	0.014	0.014	1	0.01	0.01	0.96	0.01	3	0.91	0.02	0.02	0.06	1
0.912	0.029	0.029	0.029	1	0.03	0.03	0.03	0.91	4	0.04	0.04	0.87	0.04	3
0.029	0.029	0.029	0.912	4	0.03	0.91	0.03	0.03	2	0.04	0.04	0.87	0.04	3
0.016	0.016	0.016	0.953	4	0.02	0.95	0.02	0.02	2	0.02	0.02	0.93	0.02	3
0.016	0.016	0.016	0.953	4	0.02	0.95	0.02	0.02	2	0.02	0.02	0.93	0.02	3
0.331	0.637	0.016	0.016	2	0.02	0.02	0.31	0.66	4	0.93	0.02	0.02	0.02	1

Figura 3.8: Pre-clustering y Factorización Matricial no Negativa Bayesiana(BNMF+): Factores de los usuarios. Figura tomada de [Bobadilla et al., 2018].

La naturaleza dispersa de los conjuntos de datos RS hace que sea difícil elegir los centroides iniciales y los elementos constitutivos de cada clúster. Como cada usuario solo vota una cantidad muy pequeña de los ítems disponibles, la mayoría de los pares <usuario, ítem> en la matriz de votos están vacíos. Esta es una circunstancia que no existe en la mayoría de las situaciones de clustering. Completar la matriz de votos con valores cero, o con valores intermedios, no es una solución adecuada. La tabla 3.1 muestra que la calidad de los resultados de predicción es muy deficiente siguiendo estos enfoques.

Tabla 3.1: Mean Absolute Error (MAE) del RS cuando los ítems no votados por cada usuario han sido completados con: voto 0, con voto 3, con el promedio de los votos de cada usuario. Votos de 1 a 5. Conjunto de Datos MovieLens 1M. Tabla tomada de [Bobadilla et al., 2018].

$r_{u,i}=0$	$r_{u,i}=3$	$r_{u,i}=avg(r_{u,i})$
2.1144	0.9757	0.875

El CF RS basado en memoria aborda el problema de la dispersión mediante el uso de medidas de similaridad personalizadas [Bobadilla and Serradilla, 2009, Bobadilla et al., 2012, Bobadilla et al., 2010, Mahara et al., 2016]. Estas medidas de similaridad proporcionan distancias entre los ítems del RS (o usuarios) que ofrecen mejores resultados que las medidas estadísticas clásicas (distancia Euclidea, Coseno, Manhattan, etc.) [Sanchez et al., 2008, Bobadilla et al., 2013]. De esta forma, usaremos las medidas de similaridad de RS para seleccionar los centroides iniciales en el proceso de pre-clustering propuesto. Nuestra propuesta consiste en la modificación del algoritmo 3; se ha implementado el pre-clustering sustituyendo los pasos de inicialización de los parámetros (Paso 1 del Algoritmo 3) por la inicialización k-meansPlusLogPower según el algoritmo 6.

Algoritmo 6 Inicialización de parámetros BNMF+

Input: $\mathbb{M}_{train}$ Votos de entrenamiento, $\{c_1, c_2, \dots, c_k\}$ k clústers

Output: $\gamma_{u,k}, \epsilon_{i,k}^+, \epsilon_{i,k}^-$

1: **Call** *k-means clustering* (Algoritmo 1)

Con inicialización de centroides *Pre-clustering k-meansPlusLogPower* (Algoritmo 6)

2: Normalizar $\mathbb{M}_{train}$ en $r_{u,i}^+$ y $r_{u,i}^-$ con las ecuaciones 3.11 y 3.12

3: **for all** cluster c **do**

4: **for all** usuario $u \in k$ **do**

5: $votos^+ = 0$

6: $votos^- = 0$

7: **for all** ítem i votado por u **do**

8: $votos^+ = r_{u,i}^+ + votos^+$

9: $votos^- = r_{u,i}^- + votos^-$

10: **end for**

11: actualizar $\epsilon_{i,k}^+ = \beta + votos^+$

12: actualizar $\epsilon_{i,k}^- = \beta + votos^-$

13: **end for**

14: **end for**

15: Inicializar aleatoriamente $\gamma_{u,k}$

16: **return** $\gamma_{u,k}, \epsilon_{i,k}^+$ y $\epsilon_{i,k}^-$

Algoritmo 7 Pre-clustering k-meansPlusLogPower**Input:** $\mathbb{M}_{train}$ Votos de entrenamiento, k número de clústers**Output:** k centroides, $c_1, c_2, \dots, c_k$

- 1: Seleccionar el “user power” como el centroide inicial c_1
- 2: **repeat**
- 3: Seleccionar el siguiente centroide c_i según la ecuación 3.17
- 4: **until** k centroides son seleccionados
- 5: **return** $c_1, c_2, \dots, c_k$ k centroides

El método de pre-clustering propuesto en esta tesis consiste en dos fases que se ejecutan consecutivamente. La fase 1 es el algoritmo 7 [Zahra et al., 2015]. Esta es una variante de k-means, que utiliza usuarios “power” para encontrar K centroides. Los usuarios “power” son los usuarios con el máximo número de votos en el conjunto de entrenamiento. Por lo tanto, el primer centroide corresponde al usuario que ha emitido la mayor cantidad de votos (o uno aleatorio, en caso de empate). Los centroides siguientes se seleccionan buscando la mayor distancia de los existentes. Usando k-meansPlusLogPower, la probabilidad de que un usuario del conjunto de entrenamiento se convierta en uno de los K centroides iniciales se calcula mediante la ecuación 3.17.

$$P_u \in C = \sum_{c \in C} dist(u, c) + \log \left(\frac{|I_c|}{|I_u|} + 1 \right) \quad (3.17)$$

Donde:

C es el conjunto de centroides existentes; $C \in [1..K]$

$|I_u|$ es el número de ítems votados por el usuario u .

Para implementar la distancia $dist(u, c)$ en la ecuación anterior, se han probado diferentes medidas de similaridad de RS, y comparado la calidad de los resultados con la obtenida sin realizar el pre-clustering. Los resultados sugieren una mejora en la calidad cuando se aplica una medida de similaridad adecuada; en este caso: Mean Jaccard Difference (MJD) [Bobadilla et al., 2012]. MJD mejora los resultados en comparación con la correlación de Pearson tradicional, utilizada para implementar los algoritmos de k-meansPlus [Zahra et al., 2015]. MJD también provee mejores resultados que la medida de similaridad Jaccard Mean Squared Differences (JMSD) [Bobadilla et al., 2010].

La fase 1 muestra el clúster $k \in [1..K]$ al cual pertenece cada usuario del RS. En este sentido, para cada usuario del RS $u \in U$, se puede conocer si, inicialmente, pertenece a un clúster k .

La fase 2 inicializa el aprendizaje de parámetros de BNMF, tomando en cuenta los resultados de clustering obtenidos en la fase 1. Específicamente, los parámetros aprendidos $\epsilon_{i,k}^+$ y $\epsilon_{i,k}^-$ son inicializados con las ecuaciones 3.18 y 3.19. La implementación se puede apreciar en el algoritmo 6.

$$\epsilon_{i,k}^+ = \beta + \sum_{u \in Kr_{u,i} \neq \bullet} r_{u,i}^+ \quad (3.18)$$

$$\epsilon_{i,k}^- = \beta + \sum_{u \in Kr_{u,i} \neq \bullet} r_{u,i}^- \quad (3.19)$$

En resumen, el uso de la fase de pre-clustering contribuye a mejorar la calidad de los resultados de clustering. Como se verá en el capítulo siguiente, hacer uso de la fase de pre-clustering también mejora la calidad de las predicciones del RS y los tiempos de ejecución del clustering. Las contribuciones de esta tesis para el pre-clustering BNMF son a) elección de la mejor medida de similaridad para los métodos *k-meansPlus*, y b) método innovador de obtención de los valores iniciales de los parámetros BNMF ($\epsilon_{i,k}^+$ y $\epsilon_{i,k}^-$), a partir de los resultados de pre-clustering.

Capítulo 4 Experimentos

En este capítulo se diseña un conjunto de experimentos con el fin de mostrar la validez de la hipótesis: el uso de Factorización Matricial no Negativa Bayesiana mejora los resultados de clustering en entornos de CF RS. Además, se verifica que la calidad de las predicciones del RS también mejora, reduciendo el error promedio de predicción (MAE).

La sección 4.1 explica los experimentos que se han llevado a cabo: métodos que han sido ejecutados, elección de parámetros, métodos actuales y representativos (baselines), conjuntos de datos probados, y medidas de calidad para verificar las mejoras obtenidas. La sección 4.2 expone los criterios de selección del mejor número de clústers para experimentos en conjuntos de datos de RS. La sección 4.3 analiza los parámetros de BNMF usados para hacer experimentos. La sección 4.4 se enfoca en probar la calidad de clustering, midiendo la cohesión de datos. La sección 4.5 prueba la exactitud de las predicciones, analizando el error medio de las predicciones. Finalmente, la sección 4.6 muestra las mejoras en tiempo de ejecución a través del uso del pre-clustering propuesto.

4.1. Diseño de experimentos

Para los experimentos se utilizaron los siguientes conjuntos de datos públicos para RS (open datasets):

- **FilmTrust**: Es un conjunto de datos pequeño, fue construido por un crawler a partir del sitio Web de FilmTrust en junio de 2011. Contiene 35,494 votos provistos por 1,508 usuarios sobre 2,071 películas. Los ítems están valorados con una escala decimal en el rango de 0.5 (desagrado) a 4 (agrado) con saltos de 0.5. Su dispersión es 98.8% y otra de sus características es la ausencia de equilibrio en el número de votos emitidos por los usuarios (algunos usuarios han votado solo por un ítem, mientras otros han votado por docenas de ítems), en la figura 4.1.b se puede apreciar la distribución de votos para este conjunto de datos.
- **MovieLens 1M** [Harper and Konstan, 2016]: Es un conjunto de datos provisto por GroupLens [GroupLens, 2003]. Contiene 1,000,209 votos proporcionados por 6,040 usuarios sobre 3900 películas. Los ítems están valorados en el rango de 1 (agrado) a 5 (desagrado) con saltos de 1. Su dispersión es del 95.75% y es considerado un conjunto de datos sesgado (la mayoría de votos están sesgados hacia la valoración de 4, ver figura 4.1.a.).
- **NetFlix**[The Netflix Prize, 2006]: Este conjunto de datos fue provisto para el concurso “The Netflix Prize” [Bennett et al., 2007] en octubre de 2006. Contiene 100,480,507 votos, realizados por 480,189 usuarios sobre 17,770 películas. Los ítems están valorados en el rango de 1 (agrado) a 5 (desagrado) con saltos de 1. Su dispersión es del 98.8%. En los experimentos se utiliza un conjunto de datos personalizado de Netflix. Netflix*: contiene 100,000 usuarios aleatoriamente seleccionados del conjunto de datos original. Esta reducción de tamaño ha sido necesaria para realizar experimentos que implican una gran cantidad de clústers. En la figura 4.1.c se indica la distribución de los votos para este conjunto de datos.

Para el diseño de los experimentos se utiliza la técnica 3-fold cross-validation:

- 66.66% para el conjunto de entrenamiento.
- 33.33% para el conjunto de testing.

El conjunto de testing lo dividimos, a su vez en:

- 13.33% para el conjunto de validación (validation) que permite seleccionar la mejor configuración de los parámetros α , β y K .

- 20% para el conjunto de testing que permite medir la calidad de las predicciones (MAE) y la calidad del clustering (within-cluster).

Se ejecutan pruebas para:

1. El método de Factorización Matricial no Negativo Bayesiano (BNMF).
2. El pre-clustering propuesto.
3. La combinación de ambos (BNMF+).

Los métodos actuales y representativos (baselines) para comparar los resultados son: Métodos k-means mejorados y Factorización Matricial. Se mide la calidad de las predicciones, la cohesión de los datos y los resultados de rendimiento en función del tiempo de ejecución. Se prueba un amplio rango de tamaños de clústers: desde 6 a 400.

Figura 4.1: Distribución de los votos en los conjuntos de datos empleados para los experimentos, (a) MovieLens 1M, (b) FilmTrust y (c) NetFlix

En la tabla 4.1 se resume las decisiones más relevantes que se han tomado en el diseño de los experimentos.

Tabla 4.1: Esquema del diseño de los experimentos

Diseño de experimentos			
Métodos propuestos	Métodos actuales (baselines)	Conjuntos de datos	Medidas de calidad
$BNMF^a$ $Pre - clusteringBNMF +$ $(BNMF + pre - clustering)^b$	$FactorizaciónMatricial(MF)^{c,d}$ $K - meansPlusLogPower^b$	MovieLens Netflix FilmTrust	Cohesión de datos (Within-cluster) Mean Absolute Error (MAE) Rendimiento (execution time)

a [Hernando et al., 2016], b [Zahra et al., 2015], c [Li et al., 2017], d [Ortega et al., 2016]

4.2. Selección del número de clústers

Uno de los mayores desafíos de los métodos de clustering, por su naturaleza de aprendizaje no supervisado, es la determinación del número de clústers ideal (parámetro K). Los métodos actuales se basan en heurísticas computacionales para determinar el mejor K [Halkidi et al., 2001, Kodinariya and Makwana, 2013, Bholowalia and Kumar, 2014]. Para BNMF, en la sección 3.1 se especificó que el parámetro α controla el nivel de solapamiento del usuario para pertenecer a varios clústers. Por otro lado, el parámetro β representa el nivel de evidencia que necesita el modelo para asignar un usuario a un clúster o a otro. Para las pruebas de determinación del mejor número de clústers se utiliza un enfoque de hard clustering, es decir un valor bajo de α , y se modifica el parámetro hasta llegar a un enfoque de soft clustering, valor alto de α .

Según [Hernando et al., 2016], la mejor configuración del parámetro β para RS es $\beta = 5$. En la sección 4.3 se profundiza y ratifica que la mejor inicialización de este parámetro es 5, porque en el clustering para los RS no se puede dejar de lado la calidad de las predicciones. La figura 4.2.a muestra como los mejores resultados de calidad de las predicciones (MAE) se obtienen seleccionando valores bajos de k (entre 6 y 10), sin embargo, en ese intervalo, la calidad del clustering no es la mejor.

a)

b)

Figura 4.2: MAE y Within-Cluster para determinar el mejor K con diferentes configuraciones del parámetro α . Conjunto de datos: MovieLens 1M. Cuanto menor es el valor del MAE, mejor es la calidad de predicción. Cuanto mayor es el valor de within-cluster, mejor es la calidad de clustering

La figura 4.2.b nos permite observar un patrón característico de los métodos de clustering: mientras más alto es k , mayor es la calidad del clustering, sin que esto represente que se obtienen mejores grupos o clústers con lo que realizar recomendaciones. [Bholowalia and Kumar, 2014] indican que el método del codo “Elbow method” es el más utilizado en la determinación del mejor número de clústers; consiste en un método visual, que requiere altos esfuerzos de cómputo, puesto que se necesita calcular incrementalmente la cohesión de un método de clustering desde $k=2$ (cantidad de clústers mínima) hasta el número de muestras del conjunto de datos (máxima cohesión teórica, puesto que por cada muestra existe un clúster). Siguiendo este método el mejor k sería 200, pero las curvas con distintos valores de α muestran que el comportamiento del clustering es inestable mientras el valor de α aumenta. En la figura 4.2.b se presenta una estabilidad cuando el número de clústers es pequeño (k bajo), que coincide con la mejor calidad de las predicciones, por ello para los experimentos y demostraciones siguientes se utiliza un valor de k entre 6 y 10.

4.3. Selección de los valores de α y β

El método BNMF permite definir los parámetros α y β (sección 3.1). Cuanto más alto es el valor de β , el modelo requiere de más evidencia para deducir que a un grupo de usuarios le agrada o desagrada un ítem. En este sentido, cuanto mayor sea el valor de β , mayor será el resultado de la calidad de clustering. Por otro lado, un alto valor de β genera predicciones conservadoras y por lo tanto empeora la exactitud. Un valor de α cercano a cero significa que el usuario tiende a pertenecer a un grupo. Un valor alto de α indica que cada usuario puede pertenecer probabilísticamente, simultáneamente, a más de un grupo. Por lo tanto, un valor de α cercano a cero conduce a un enfoque hard clustering, proporcionando mejor calidad de resultados de within-cluster. Un valor α cercano a uno conduce a un enfoque flexible de soft clustering.

El parámetro α de BNMF es particularmente prometedor para establecer las propiedades de clustering. Este parámetro determina el nivel de dispersión de los factores ocultos de los usuarios. Cuanto menor sea el valor de α es más probable que se asigne la mayoría de las características del usuario a un factor oculto único, mientras que altos valores de

α proporciona más flexibilidad para distribuir las características de cada usuario entre el conjunto de factores ocultos. Valores más bajos de α facilita la asignación de cada usuario a un único clúster, mientras que si se utiliza un mayor valor de α es más natural asignar de forma probabilística a cada usuario a un conjunto de clústers (soft-clustering).

Usando MF, NMF o BNMF, cada usuario está representado por un número K de factores ocultos. Cada factor oculto representa y codifica una mezcla de características; como un ejemplo simplificado: El factor 1 podría representar películas de acción, pero no las películas de terror, mientras el factor 2 podría ser asociado con las películas de humor. En este sentido, si $K = 2$, un usuario que contiene un valor de 0.85 en el factor 1 y un valor de 0.15 en el factor 2, ha sido representado como un fan de las películas de acción, a quien le desagrade las películas de terror y no le agrada mucho las películas de humor. Utilizando BNMF, cuando α es pequeño, las características de cada usuario se conectan a un solo factor “determinante” o en un número reducido de factores. Cuando α es alto, la cantidad de evidencia necesaria para caracterizar a un usuario en varios factores es menor.

La figura 4.3 muestra los resultados del siguiente experimento: usando el conjunto de datos MovieLens 1M, el proceso BNMF es realizado para diferentes valores de α . Este experimento establece en 6 el número de clústers ($K = 6$). Para cada α , a modo de ejemplo, se toma la media del factor oculto 5 para aquellos usuarios cuyo factor 5 es mayor que los otros factores (barras negras en la figura 4.3). Como se esperaba, para pequeños valores de α un único factor (factor 5) determina casi todo el comportamiento del usuario, mientras que con grandes valores de α , el factor principal pierde peso y el resto de factores también definen las características de los usuarios. En la figura 4.3, cuando α tiende a 1, el factor 5 determina menos de la mitad de las características del usuario.

La Tabla 4.2 muestra todos los valores promediados correspondientes a BNMF con seis factores ocultos $K = 6$. Cada tabla se ha obtenido utilizando un valor α diferente ($\alpha \approx 0$;

Figura 4.3: Distribución de los factores ocultos de los usuarios de MovieLens 1M, BNMF, $K = 6$. Resultados promediados para los usuarios cuyo factor oculto máximo es f_5 . Figura tomada de [Bobadilla et al., 2018]

$\alpha = 0,2$; $\alpha = 0,4$; $\alpha = 0,6$; $\alpha = 0,8$ y $\alpha \approx 1$). Cada columna de cada tabla muestra los valores promediados del factor 1 (f_1) al factor 6 (f_6). Cada factor de columna es el que tiene el valor máximo. Para comprender la influencia del parámetro α para el enfoque de clustering se proponen los dos ejemplos siguientes:

- Ejemplo *Hard clustering*: En la tabla correspondiente a $\alpha = 0,2$; la columna f_2 muestra que el factor 2 determina, en promedio, el 78% de las características de los usuarios más caracterizados por el factor 2, mientras que esos mismos usuarios no determinan más del 5% en ninguno de los otros factores. Esto demuestra que el particionamiento con valores bajos de α es determinista (Hard clustering).
- Ejemplo *Soft clustering*: En la tabla correspondiente a $\alpha = 0,8$; la columna f_3 muestra que el factor 3 determina, en promedio, el 51% de las características de los usuarios más caracterizados por el factor 3, mientras que esos mismos usuarios determinan el 11% para el factor 6. Es evidente que ahora los usuarios más caracterizados por el factor 3 también se ven caracterizados por los otros factores pero en diferentes niveles. Por lo tanto ya no existe un particionamiento tan determinista.

Una vez que se ha detallado el significado del parámetro α , la figura 4.4 muestra los resultados de MAE y de calidad de within-cluster en BNMF cuando se aplican diferentes valores de α y β son aplicados al conjunto de datos MovieLens 1M, según la tabla 4.3. Estos resultados fueron obtenidos usando 10 clústers ($K = 10$). La fi-

Tabla 4.2: Distribución de los factores ocultos de los usuarios para MovieLens 1M, BNMF, $K=6$. Resultados promediados para todos los usuarios. Tabla tomada de [Bobadilla et al., 2018].

$\alpha \approx 0$ $k=6$ $\beta=5$						
	f1	f2	f3	f4	f5	f6
f1	0.81	0.03	0.03	0.04	0.05	0.04
f2	0.05	0.84	0.02	0.03	0.03	0.02
f3	0.04	0.02	0.85	0.03	0.02	0.04
f4	0.06	0.03	0.02	0.76	0.09	0.04
f5	0.05	0.03	0.3	0.05	0.81	0.02
f6	0.05	0.02	0.02	0.05	0.01	0.85

$\alpha=0,2$ $k=6$ $\beta=5$						
	f1	f2	f3	f4	f5	f6
f1	0.77	0.04	0.06	0.05	0.03	0.05
f2	0.06	0.78	0.04	0.04	0.03	0.04
f3	0.08	0.05	0.76	0.03	0.06	0.02
f4	0.05	0.03	0.03	0.77	0.06	0.06
f5	0.04	0.04	0.06	0.10	0.71	0.04
f6	0.05	0.03	0.03	0.07	0.04	0.78

$\alpha=0,4$ $k=6$ $\beta=5$						
	f1	f2	f3	f4	f5	f6
f1	0.73	0.06	0.06	0.04	0.07	0.06
f2	0.07	0.72	0.04	0.03	0.06	0.08
f3	0.05	0.04	0.70	0.06	0.04	0.10
f4	0.04	0.04	0.10	0.59	0.13	0.11
f5	0.06	0.05	0.03	0.06	0.70	0.09
f6	0.04	0.07	0.07	0.05	0.07	0.70

$\alpha=0,6$ $k=6$ $\beta=5$						
	f1	f2	f3	f4	f5	f6
f1	0.58	0.12	0.07	0.10	0.05	0.09
f2	0.08	0.63	0.08	0.10	0.06	0.05
f3	0.09	0.13	0.58	0.06	0.06	0.08
f4	0.07	0.10	0.04	0.64	0.08	0.07
f5	0.06	0.09	0.05	0.12	0.59	0.10
f6	0.09	0.07	0.07	0.10	0.09	0.57

$\alpha=0,8$ $k=6$ $\beta=5$						
	f1	f2	f3	f4	f5	f6
f1	0.59	0.09	0.10	0.10	0.06	0.06
f2	0.10	0.57	0.10	0.08	0.09	0.06
f3	0.13	0.12	0.51	0.09	0.05	0.09
f4	0.11	0.10	0.10	0.52	0.07	0.10
f5	0.07	0.14	0.07	0.08	0.53	0.10
f6	0.07	0.10	0.11	0.13	0.10	0.50

$\alpha \approx 1$ $k=6$ $\beta=5$						
	f1	f2	f3	f4	f5	f6
f1	0.48	0.11	0.09	0.10	0.13	0.10
f2	0.11	0.52	0.06	0.06	0.09	0.15
f3	0.10	0.07	0.49	0.14	0.10	0.10
f4	0.08	0.06	0.10	0.52	0.12	0.13
f5	0.12	0.09	0.08	0.14	0.47	0.10
f6	0.08	0.11	0.06	0.12	0.09	0.53

Figura 4.4.a muestra el impacto del MAE cuando diferentes valores son aplicados a los parámetros α y β . Como se esperaba, la exactitud mejora cuando el valor de β decrece. La línea horizontal en la figura 4.4.a indica el valor de MAE provisto por la MF clásica. El objetivo es probar la superioridad del método BNMF para hacer RS clustering, sin perder la precisión de las predicciones. En este sentido se selecciona el valor $\beta=5$ lo que brinda un margen de mejora tanto en exactitud como en calidad de clustering.

La figura 4.4.b muestra el impacto en within-cluster cuando diferentes valores son aplicados a los parámetros α y β . Los valores de Within-cluster (eje y) se han normalizado al intervalo [0..1], puesto que utilizamos medidas de similaridad, mientras más alto sea el valor, se entiende una mejor calidad de clustering. Como se predijo, un valor bajo de β ($\beta=5$) produce peores resultados de clustering que los valores altos de β . Cuando el proceso de clustering es independiente del proceso de recomendación es más apropiado seleccionar un

valor alto de β . En este caso, buscamos un equilibrio de RS: se desea mejorar la calidad de clustering sin empeorar la calidad de las predicciones, de modo que se mantiene la elección de $\beta=5$.

Una vez seleccionado el valor de $\beta=5$, en la figura 4.4.b se puede ver que los valores de α cercanos a cero brindan una mejor calidad de clustering sin comprometer la calidad de la exactitud (figura 4.4.a). En este sentido se selecciona el valor de $\alpha \approx 0$ y por lo tanto, se elige un enfoque hard-clustering. Es importante indicar que el método BNMF permite valores de $\alpha \in (0,1]$, por lo que $\alpha \approx 0$ significa que el valor de α es muy cercano a cero.

a)

b)

Figura 4.4: Resultados de MAE y calidad de within-cluster obtenidos cuando diferentes valores de los parámetros α y β son combinados. Conjunto de datos: MovieLens 1M, Método: BNMF, Número de clústers (K): 10. a) Los valores más bajos son los mejores, b) Los valores más altos son los mejores. Figura tomada de [Bobadilla et al., 2018]

Tabla 4.3: Resultados de MAE obtenidos cuando diferentes valores de los parámetros α y β son combinados. Conjunto de datos: MovieLens 1M, Método: BNMF, Número de clústers (K): 10.

α	$\beta = 5$		$\beta = 35$		$\beta = 65$		$\beta = 95$	
	Within-Cluster	MAE	Within-Cluster	MAE	Within-Cluster	MAE	Within-Cluster	MAE
≈ 0.0	0.888	0.675	0.931	0.693	0.943	0.713	0.947	0.720
0.1	0.884	0.670	0.909	0.691	0.930	0.705	0.940	0.723
0.2	0.873	0.667	0.915	0.685	0.925	0.702	0.933	0.718
0.3	0.865	0.667	0.892	0.684	0.914	0.700	0.941	0.711
0.4	0.861	0.666	0.908	0.681	0.912	0.700	0.942	0.702
0.5	0.857	0.661	0.906	0.682	0.930	0.691	0.938	0.705
0.6	0.852	0.661	0.913	0.682	0.951	0.701	0.967	0.715
0.7	0.847	0.664	0.915	0.682	0.918	0.704	0.951	0.716
0.8	0.850	0.662	0.914	0.681	0.926	0.702	0.959	0.720
0.9	0.839	0.663	0.898	0.687	0.938	0.706	0.951	0.719
≈ 1.0	0.839	0.665	0.904	0.684	0.919	0.705	0.932	0.722

4.4. Mejoras de clustering

En esta sección se presentan los resultados de calidad obtenidos de cada experimento de clustering, de acuerdo al diseño presentado en la tabla 4.1. Entre las medidas de calidad de clustering existentes [Siddiqi and Sait, 2017, Zhao et al., 2009] se ha tomado la más representativa y popular: cohesión (Within-Cluster). Por lo general, esta medida es definida como la suma de las distancias entre cada elemento del clúster y su respectivo centroide. La ecuación siguiente formaliza el concepto:

$$cohesion = \sum_{k=1}^K \sum_{\forall u \in C_k} similarity(u, c_k) \quad (4.1)$$

Donde: K es el número de clústers, C_k es el k clúster, c_k es el k centroide del clúster, u es un usuario del RS, similaridad es la medida de similaridad seleccionada (Pearson, MJD, Euclídea, etc.). Cuanto mayor sea el valor de cohesión, mejor será la calidad de clustering.

Dado que se utiliza conjuntos de datos de RS, se ha seleccionado la correlación de Pearson como medida de similaridad [Bobadilla et al., 2013]: Es una medida de similaridad para CF clásica y devuelve valores limitados entre $[-1..1]$. Además, para poder comparar resultados a través de diferentes conjuntos de datos, se utiliza una cohesión normalizada

en el rango $[-1..1]$ dividiendo su valor entre el número de usuarios del RS (U):

$$cohesion = \frac{1}{U} \sum_{k=1}^K \sum_{\forall u \in c_k} correlation(u, c_k) \quad (4.2)$$

De acuerdo al diseño de los experimentos de la tabla 4.1, se ha probado los métodos propuestos junto con los métodos más actuales seleccionados (baselines). La figura 4.5 muestra los resultados de los conjuntos de datos MovieLens, Netflix y FilmTrust. Como se explicó en la sección 4.3, los valores elegidos para los parámetros son: $\beta = 5$, $\alpha \approx 0$. Como se puede observar en la figura 4.5, los resultados de within-cluster en BNMF y BNMF+ son similares.

Analizando las tres gráficas de la figura 4.5, las conclusiones principales que se pueden extraer son:

- El método BNMF propuesto mejora en gran medida la calidad de clustering de los métodos actuales y representativos (K-means+ y MF). La figura 4.6 muestra los porcentajes de mejora del clustering de BNMF sobre el promedio de los métodos actuales y representativos.
- BNMF y BNMF+ proporcionan la misma calidad de clustering, por lo que sus líneas se superponen casi por completo: se espera mejoras en el tiempo de ejecución en lugar de en la calidad de clustering.
- K-means+ obtiene mejores resultados que MF cuando se aplica a conjuntos de datos de RS, lo cual confirma la efectividad de la inicialización Plus-Log-Power[Zahra et al., 2015].
- Como se esperaba, cuanto mayor sea el número de clústers (K) mejor será la calidad de clustering: existe una mayor variedad de centroides para asignar a cada elemento.
- La calidad de clustering disminuye cuando aumenta el tamaño del conjunto de datos; FilmTrust muestra los mejores resultados, seguido de MovieLens, y finalmente Netflix: cuanto mayor se al conjunto de datos, más elementos, en promedio, se deben asignar a cada clúster.
- La mejora de clustering de BNMF sobre los métodos actuales y representativos (MF y K-means+) es más alta cuando el número de clústers (K) es menor (figura 4.6): Existe un margen mayor de mejora en los valores de K más bajos (figura 4.5).

- A partir de un número significativo de clústers ($K > 20$), la mejora de BNMF con respecto a los métodos actuales y representativos es mayor en los conjuntos de datos más grandes (figura 4.6): existe un margen mayor de mejora en los grandes conjuntos de datos (figura 4.5).

Figura 4.5: Resultados de clustering utilizando a) FilmTrust, b) MovieLens, y c) Netflix. Eje-y: resultados de cohesión normalizada (within-cluster); eje-x: Número de clústers (K). $\beta = 5$, $\alpha \approx 0$. Los valores más altos son los mejores. Figura tomada de [Bobadilla et al., 2018]

Figura 4.6: Mejoras de clustering BNMF en comparación a los métodos actuales y representativos (MF y K-means+). Eje-x: número de clústers (K); eje-y: % de mejora. $\beta = 5$, $\alpha \approx 0$. Figura tomada de [Bobadilla et al., 2018]

4.5. Mejoras en la exactitud de las predicciones

En esta sección determinamos la capacidad de cada método puesto a prueba (K-means+, MF, BNMF, BNMF+) para generar predicciones correctas. La figura 4.7 muestra los resultados de la calidad de las predicciones (MAE) para cada conjunto de datos probado.

Las principales conclusiones de las gráficas de la figura 4.7 son:

- El método propuesto (BNMF) proporciona mejores resultados de exactitud que los métodos actuales y representativos (baselines).
- Utilizar el pre-clustering propuesto alcanza una mejora adicional en la exactitud.
- Cuando se usa el método de MF clásico para agrupar, la calidad de las predicciones se ve significativamente afectada al seleccionar valores de K altos.

Comparando los gráficos en la figura 4.5 y la figura 4.7, podemos determinar una relación inversa entre la calidad de clustering y la calidad de las predicciones. La figura 4.8 muestra, en el rango $[0..1]$, la evolución de la exactitud de BNMF + (expresada como $1 - \text{MAE}$) y la evolución del within-cluster BNMF+; Conjuntos de datos: Netflix (NF) y MovieLens (ML). Los puntos donde la exactitud y within-cluster se cruzan nos dan un $K = 35$ óptimo para Net-

flix y $K = 20$ para MovieLens. Estos valores son consistentes con la necesidad habitual de aumentar el número de clústeres a medida que aumenta el tamaño de los conjuntos de datos.

Figura 4.7: Resultados de MAE utilizando a) FilmTrust, b) MovieLens, and c) Netflix*. eje-y: Resultados de MAE; eje-x: número de clústers (K). $\beta = 5$, $\alpha \approx 0$. Los valores más bajos son los mejores. Figura tomada de [Bobadilla et al., 2018]

Figura 4.8: Tendencias de la exactitud (1-MAE) y within-cluster utilizando BNMF+ en MovieLens (ML) y Netflix* (NF). eje-y: (1-MAE) y resultados normalizados de within-cluster en el rango [0..1]; eje-x: número de clústers (K). $\beta=5$, $\alpha \approx 0$. Figura tomada de [Bobadilla et al., 2018]

4.6. Mejoras en tiempo de ejecución de Pre-clustering

En las secciones anteriores se ha verificado como el uso de pre-clustering implica una ligera mejora en la calidad de las predicciones (MAE) para BNMF. Esta subsección se enfoca en la mejora tanto del tiempo de ejecución que se logra al utilizar la técnica propuesta BNMF pre-clustering, como en la superioridad del tiempo de ejecución de BNMF con respecto a los métodos actuales y representativos seleccionados.

Si analizamos el algoritmo 3 de la sección 3.1, en el paso 1 se realiza una llamada al algoritmo 6, que consiste en la inicialización de parámetros BNMF+, cuyo núcleo es la ejecución del método k-meansPlus LogPower; en principio, esta agregación de cómputo debería representar más tiempo de ejecución. Sin embargo, la figura 4.9 muestra que la ventaja del pre-clustering consiste en reducir las iteraciones del entrenamiento del método BNMF, lo cual es una mejora importante en tiempo de ejecución. Para los experimentos se utilizó un pre-clustering basado en las medidas de similitud JMSD y MJD. Es notorio que utilizando MJD el método BNMF llega a su condición de parada en la iteración 15, con JMSD se alcanza la condición de parada en la iteración 25, mientras que sin pre-clustering el método BNMF requiere 65 iteraciones.

Figura 4.9: Entrenamiento de BNMF sin pre-clustering y con pre-clustering. Conjuntos de datos: MovieLens

En un nuevo conjunto de experimentos, se ha ejecutado tanto los métodos propuestos (BNMF y BNMF+) como los métodos actuales y representativos (K-means+ y MF). Se muestran resultados eligiendo diferentes valores de K (desde $K = 6$ a $K = 400$), $\beta = 5$, $\alpha \approx 0$. La figura 4.10 muestra los resultados; las conclusiones más relevantes son: a) El tiempo de ejecución de BNMF despues del pre-clustering (BNMF+) es mejor que el tiempo de ejecución BNMF sin pre-clustering, b) Los métodos actuales son más lentos que los de BNMF, c) Las diferencias de rendimiento son significativas cuando el número de clústers no es pequeño (mayores que 50), d) Cuando se usan conjuntos de datos pequeños (por ejemplo FilmTrust), los métodos basados en MF son más lentos que k-means+, y e) El clustering de MF es más lento que k-means+ y BNMF, lo cual enfatiza la idoneidad de BNMF y BNMF+ como un método basado en modelos para RS.

Figura 4.10: Rendimiento de ejecución tanto de los métodos propuestos como de los métodos actuales y representativos. Conjuntos de datos: a) FilmTrust, b) MovieLens, c) Netflix*; eje-x: número de clústers (K); eje-y: tiempo (minutos). $\beta=5$, $\alpha \approx 0$. Figura tomada de [Bobadilla et al., 2018]

Conclusiones

Más allá de la calidad de las recomendaciones, el clustering en sistemas de recomendación es importante; permite abordar varios desafíos del filtrado colaborativo: explicación de las recomendaciones, análisis de datos, visualización y navegación a través de la información del conjunto de datos, obtención de las características que definen a cada grupo de usuarios o elementos, etc. El clustering en RS es una área de investigación que representa un desafío para los sistemas de recomendación, debido al alto nivel de dispersión que presentan los conjuntos de datos.

Los métodos basados en modelos obtienen resultados precisos y aceleran el proceso de predicción una vez que el modelo ha sido entrenado. Entre los métodos basados en modelos, las técnicas de factorización matricial son las más populares. Los métodos de factorización matricial clásicas no proporcionan flexibilidad en su funcionamiento más allá de la elección del número de factores ocultos. El método de Factorización Matricial no Negativo Bayesiano proporciona flexibilidad en la factorización de datos. También brinda la condición útil de que cada factor oculto de un usuario representa una probabilidad.

Los experimentos confirman la hipótesis en la que se ha basado esta tesis y demuestran que el método BNMF+: a) Mejora la exactitud en las predicciones que proporcionan los métodos actuales, así como la calidad de clustering, b) Se puede configurar para incorporar mejoras de clustering reduciendo la exactitud, o viceversa, 3) Obtiene una mejora significativa en rendimiento, reduciendo tiempos de ejecución, comparados a los métodos actuales y representativos, especialmente cuando la cantidad de clústers se incrementa.

Esta tesis propone un algoritmo innovador de pre-clustering BNMF (BNMF+) que proporciona mejoras de clustering y rendimiento. Además, se abren futuras y prometedoras líneas de trabajo basadas en obtener mejoras de clúster a través de un enfoque BNMF soft clustering. BNMF se puede parametrizar de modo que cada usuario o elemento pertenezca probabilísticamente a varios clústers. Un enfoque soft de BNMF también abre la posibilidad de mejorar la exactitud de los sistemas de recomendación. La explicación del significado de los factores ocultos es una debilidad de las técnicas de factorización matricial; el incorporar características Bayesianas a las técnicas de factorización matricial puede representar otro campo de investigación para la explicación de estos factores. En entornos de aprendizaje no supervisado, como los conjuntos de datos de RS, donde se desconoce las etiquetas de los datos, las técnicas de factorización matricial pueden ofrecer una alternativa para determinar el mejor tamaño de clústers. Mientras que actualmente la selección del número de clústers se realiza en base a heurísticas computacionales, utilizando métodos de factorización matricial, como el expuesto en la tesis, la reducción de dimensionalidad ofrece una nueva vía prometedora para la determinación del número de clústers en cada dataset.

Conclusions

Beyond accuracy in predictions, recommender systems clustering is important; it allows to face several collaborative filtering challenges: recommendations explanation, data analytics, visualization and browsing through the dataset information, obtaining the characteristics that define each group of users or items, etc. RS clustering is a challenging research area for recommender systems, due to the high sparsity in their datasets.

Model-based methods get accurate results and they accelerate the prediction process once the model has been trained. Among model-based methods, matrix factorization techniques are the most popular ones. Matrix factorization methods do not provide flexibility in their operation apart from the choice of the number of hidden factors. The Bayesian Non-Negative Matrix Factorization method gives flexibility factorizing data. It also provides a useful condition: that each individual user hidden factor is a probability.

Experiments confirm the hypothesis where this thesis is based and show that the BNMF+ method: a) It improves, simultaneously, the baselines accuracy and their clustering quality results, b) It can be configured to increase clustering improvements reducing accuracy, or vice versa, 3) It obtains important execution times improvements, compared to baselines, especially when the number of clusters is not small.

This thesis proposes an innovative BNMF pre-clustering algorithm (BNMF+) that provides clustering and performance improvements. In addition, this thesis opens a hopeful future work: getting clustering improvements through a BNMF soft-clustering approach.

BNMF can be parameterized so that each user or item probabilistically belongs to several clusters. Soft BNMF also opens up the possibility of improving recommender systems accuracy. Explanation of recommendations is particularly difficult to achieve when matrix factorization methods are used; this is because the complex nature of their hidden factors. When Bayesian features are incorporated to the matrix factorization techniques, their probabilistic basis can be used to understand hidden factors and then to explain recommendations. Number of cluster selection is also an open research field in the recommender systems clustering area; Bayesian parameters can be used to get the adequate number of cluster for each dataset, opening a promising alternative to the current heuristic methods.

Apéndices

1. Lista de publicaciones del autor

Esta tesis resume años de mi investigación en sistemas de recomendación. Las contribuciones académicas a la comunidad científica son: tres papers publicados en revistas de inteligencia artificial con factor de impacto Q1 y un paper con factor de impacto Q3, indexadas en Journal Citation Report (JCR) y Science Citation Index (SCI), y tres trabajos en congresos internacionales cuyos proceedings están indexados en SCOPUS.

1. Bobadilla, J., Bojorque, R., Esteban, A. H., and Hurtado, R. (2018). Recommender systems clustering using bayesian non negative matrix factorization. *IEEE Access*, 6:3549–3564.
2. F. Ortega and R. Hurtado and J. Bobadilla and R. Bojorque. (2018). Recommendation to Groups of Users Using the Singularities Concept. *IEEE Access*, 6:39745-39761.
3. R. Hurtado and J. Bobadilla and R. Bojorque and F. Ortega and X. Li. (2019). A new recommendation approach based on probabilistic soft clustering methods. A scientific documentation case study. *IEEE Access*, 7:7522-7534.
4. J. Bobadilla and A. Gutiérrez and M.A. Patricio and R. Bojorque. (2019). Analysis of scientific production based on trending research topics. An Artificial Intelligence case study. *Revista Española de Documentación Científica*, 42(1), e228.
5. Bojorque, R., Hurtado, R., and Inga, C.. (2018). A Comparative Analysis of Similarity Metrics on Sparse Data for Clustering in Recommender Systems. *International Conference on Applied Human Factors and Ergonomics*, 291-299.
6. Hurtado, R., Bojorque, R. and Inga, C. (2018). Clustering-Based Recommender System: Bundle Recommendation Using Matrix Factorization to Single User and User Communities. *International Conference on Applied Human Factors and Ergonomics*, 330-338.

7. Inga, C., Bojorque, R. and Hurtado, R. (2018). Hierarchical Clustering for Collaborative Filtering Recommender Systems. *International Conference on Applied Human Factors and Ergonomics*, 346-356.

Bibliografía

- [Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.
- [Aggarwal and Reddy, 2013] Aggarwal, C. C. and Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 1st edition.
- [Al-Shamri and Bharadwaj, 2008] Al-Shamri, M. Y. H. and Bharadwaj, K. K. (2008). Fuzzy-genetic approach to recommender systems based on a novel hybrid user model. *Expert Systems with Applications*, 35(3):1386 – 1399.
- [Arabnejad et al., 2017] Arabnejad, E., Moghaddam, R. F., and Cheriet, M. (2017). Psi: Patch-based script identification using non-negative matrix factorization. *Pattern Recognition*, 67(Supplement C):328 – 339.
- [Arthur and Vassilvitskii, 2007] Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.
- [Baltrunas et al., 2011] Baltrunas, L., Ludwig, B., and Ricci, F. (2011). Matrix factorization techniques for context aware recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys ’11, pages 301–304, New York, NY, USA. ACM.
- [Bennett et al., 2007] Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA.
- [Bhatia and Deogun, 1998] Bhatia, S. K. and Deogun, J. S. (1998). Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(3):427–436.
- [Bholowalia and Kumar, 2014] Bholowalia, P. and Kumar, A. (2014). Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9).

- [Birtolo and Ronca, 2013] Birtolo, C. and Ronca, D. (2013). Advances in clustering collaborative filtering by means of fuzzy c-means and trust. *Expert Syst. Appl.*, 40(17):6997–7009.
- [Bobadilla et al., 2018] Bobadilla, J., Bojorque, R., Esteban, A. H., and Hurtado, R. (2018). Recommender systems clustering using bayesian non negative matrix factorization. *IEEE Access*, 6:3549–3564.
- [Bobadilla et al., 2012] Bobadilla, J., Ortega, F., Hernando, A., and Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26:225–238.
- [Bobadilla et al., 2013] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46(Supplement C):109 – 132.
- [Bobadilla and Serradilla, 2009] Bobadilla, J. and Serradilla, F. (2009). The effect of sparsity on collaborative filtering metrics. In *Proceedings of the Twentieth Australasian Conference on Australasian Database-Volume 92*, pages 9–18. Australian Computer Society, Inc.
- [Bobadilla et al., 2010] Bobadilla, J., Serradilla, F., and Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, 23(6):520–528.
- [Bokde et al., 2015] Bokde, D., Girase, S., and Mukhopadhyay, D. (2015). Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science*, 49(Supplement C):136 – 146. Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3’15).
- [Bouguettaya et al., 2015] Bouguettaya, A., Yu, Q., Liu, X., Zhou, X., and Song, A. (2015). Efficient agglomerative hierarchical clustering. *Expert Systems with Applications*, 42(5):2785 – 2797.
- [Breese et al., 1998] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, 461(8):43–52.
- [Buchta et al., 2012] Buchta, C., Kober, M., Feinerer, I., and Hornik, K. (2012). Spherical k-means clustering. *Journal of Statistical Software*, 50(10):1–22.
- [Buono and Pio, 2015] Buono, N. D. and Pio, G. (2015). Non-negative matrix tri-factorization for co-clustering: An analysis of the block matrix. *Information Sciences*, 301(Supplement C):13 – 26.
- [Chen et al., 2017] Chen, J., Ulji, Wang, H., and Yan, Z. (2017). Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering. *Swarm and Evolutionary Computation*.

- [Chen et al., 2014] Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2):171–209.
- [Chen et al., 2015] Chen, M.-H., Teng, C.-H., and Chang, P.-C. (2015). Applying artificial immune systems to collaborative filtering for movie recommendation. *Advanced Engineering Informatics*, 29(4):830 – 839. Collective Intelligence Modeling, Analysis, and Synthesis for Innovative Engineering Decision Making Special Issue of the 1st International Conference on Civil and Building Engineering Informatics.
- [de Campos et al., 2010] de Campos, L. M., Fernández-luna, J. M., Huete, J. F., and Rueda-morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*, 51(7):785–799.
- [Degemmis et al., 2007] Degemmis, M., Lops, P., and Semeraro, G. (2007). A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255.
- [Devarajan, 2008] Devarajan, K. (2008). Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. *PLoS Computational Biology*.
- [Diao et al., 2014] Diao, Q., Qiu, M., Wu, C.-Y., Smola, A. J., Jiang, J., and Wang, C. (2014). Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 193–202, New York, NY, USA. ACM.
- [Djenouri et al., 2018] Djenouri, Y., Belhadi, A., Fournier-Viger, P., and Lin, J. C.-W. (2018). Fast and effective cluster-based information retrieval using frequent closed itemsets. *Information Sciences*, 453:154 – 167.
- [Du et al., 2017] Du, R., Kuang, D., Drake, B., and Park, H. (2017). Dc-nmf: nonnegative matrix factorization based on divide-and-conquer for fast clustering and topic modeling. *Journal of Global Optimization*, pages 1–22.
- [Duma and Twala, 2018] Duma, M. and Twala, B. (2018). Optimising latent features using artificial immune system in collaborative filtering for recommender systems. *Applied Soft Computing*, 71:183 – 198.
- [Fraley and Raftery, 1998] Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588.
- [Frémal and Lecron, 2017] Frémal, S. and Lecron, F. (2017). Weighting strategies for a recommender system using item clustering based on genres. *Expert Syst. Appl.*, 77(C):105–113.

- [Friedman et al., 2016] Friedman, A., Berkovsky, S., and Kaafar, M. A. (2016). A differential privacy framework for matrix factorization recommender systems. *User Modeling and User-Adapted Interaction*, 26(5):425–458.
- [Gao and Li, 2008] Gao, L. and Li, C. (2008). Hybrid personalized recommended model based on genetic algorithm. In *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4.
- [George and Merugu, 2005] George, T. and Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM '05*, pages 625–628, Washington, DC, USA. IEEE Computer Society.
- [Ghazanfar and Prügel-Bennett, 2014] Ghazanfar, M. A. and Prügel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications*, 41(7):3261 – 3275.
- [Ghazanfar et al., 2011] Ghazanfar, M. A., Szedmak, S., and Prugel-Bennett, A. (2011). Incremental kernel mapping algorithms for scalable recommender systems. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 1077–1084.
- [Ghosh and Dubey, 2013] Ghosh, S. and Dubey, S. K. (2013). Comparative analysis of k-means and fuzzy c-means algorithms. *International Journal of Advanced Computer Science and Applications*, 4(4).
- [GroupLens, 2003] GroupLens (2003). Movielens dataset. <https://grouplens.org/datasets/movielens>.
- [Gu and Zhou, 2009] Gu, Q. and Zhou, J. (2009). Local learning regularized nonnegative matrix factorization. In *Proceedings 21st International Joint Conference on Artificial Intelligence*, page 1046–1051, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Halkidi et al., 2001] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of intelligent information systems*, 17(2-3):107–145.
- [Harper and Konstan, 2016] Harper, F. M. and Konstan, J. A. (2016). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19.
- [Hayashi and Watanabe, 2016] Hayashi, N. and Watanabe, S. (2016). Upper Bound of Bayesian Generalization Error in Non-negative Matrix Factorization. *ArXiv e-prints*.
- [Herlocker et al., 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53.
- [Hernández del Olmo and Gaudioso, 2008] Hernández del Olmo, F. and Gaudioso, E. (2008). Evaluation of recommender systems: A new approach. *Expert Systems with Applications*,

35(3):790–804.

- [Hernando et al., 2016] Hernando, A., Bobadilla, J., and Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a bayesian probabilistic model. *Knowledge-Based Systems*, 97(Supplement C):188 – 202.
- [Hopfgartner et al., 2014] Hopfgartner, F., Kille, B., Lommatzsch, A., Plumbaum, T., Brodt, T., and Heintz, T. (2014). Benchmarking news recommendations in a living lab. In Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., and Toms, E., editors, *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, pages 250–267, Cham. Springer International Publishing.
- [Hruschka et al., 2009] Hruschka, E. R., Campello, R. J. G. B., Freitas, A. A., Ponce, A. C., and de Carvalho, L. F. (2009). A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, MAN and Cybernetics*, 39(2):133 – 155.
- [Hu et al., 2014] Hu, R., Dou, W., and Liu, J. (2014). Clubcf: A clustering-based collaborative filtering approach for big data application. *IEEE Transactions on Emerging Topics in Computing*, 2(3):302–313.
- [Hu et al., 2010] Hu, W., Yong-Ji, W., Zhe, W., Xiu-Li, W., and Shuan-Zhu, D. (2010). , two-phase collaborative filtering algorithm based on co-clustering. *Journal of Software*, 21(5):1042–1054.
- [Huang et al., 2004] Huang, Z., Chen, H., and Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Inf. Syst.*, 22(1):116–142.
- [Jain, 2010] Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.
- [Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323.
- [Kaminskas et al., 2016] Kaminskas, M., Bridge, D., and Centre, I. (2016). Diversity , Serendipity , Novelty , and Coverage : A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. 7(1):1–42.
- [Kaminskas et al., 2013] Kaminskas, M., Ricci, F., and Schedl, M. (2013). Location-aware music recommendation using auto-tagging and hybrid matching. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys ’13, pages 17–24, New York, NY, USA. ACM.
- [Kannan et al., 2016] Kannan, R., Ballard, G., and Park, H. (2016). A high-performance parallel algorithm for nonnegative matrix factorization. In *Proceedings of the 21st ACM SIGPLAN*

- Symposium on Principles and Practice of Parallel Programming*, PPOPP '16, pages 9:1–9:11, New York, NY, USA. ACM.
- [Karimi et al., 2018] Karimi, M., Jannach, D., and Jugovac, M. (2018). News recommender systems – survey and roads ahead. *Information Processing & Management*, 54(6):1203 – 1227.
- [Kim and Park, 2008] Kim, J. and Park, H. (2008). Sparse Nonnegative Matrix Factorization for Clustering. Technical report, Georgia Institute of Technology.
- [Kodinariya and Makwana, 2013] Kodinariya, T. M. and Makwana, P. R. (2013). Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95.
- [Koren, 2009] Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation*, (August):1–10.
- [Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8).
- [Kumar et al., 2017] Kumar, V., Pujari, A. K., Sahu, S. K., Kagita, V. R., and Padmanabhan, V. (2017). Proximal maximum margin matrix factorization for collaborative filtering. *Pattern Recognition Letters*, 86(Supplement C):62 – 67.
- [Lakshminarayanan et al., 2011] Lakshminarayanan, B., Bouchard, G., and Archambeau, C. (2011). Robust bayesian matrix factorisation. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 425–433.
- [Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791.
- [Lee et al., 2002] Lee, M., Choi, P., and Woo, Y. (2002). *A Hybrid Recommender System Combining Collaborative Filtering with Neural Network*, pages 531–534. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Li et al., 2009] Li, T., Zhang, Y., and Sindhwani, V. (2009). A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 244–252, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Li et al., 2017] Li, Y., Wang, D., He, H., Jiao, L., and Xue, Y. (2017). Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems. *Neurocomputing*, 249(Supplement C):48 – 63.
- [Liao and Lee, 2016] Liao, C.-L. and Lee, S.-J. (2016). A clustering based approach to improving

- the efficiency of collaborative filtering recommendation. *Electronic Commerce Research and Applications*, 18(Supplement C):1 – 9.
- [Liu et al., 2010] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916.
- [Lu et al., 2015] Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems*, 74:12 – 32.
- [Lü et al., 2012] Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., and Zhou, T. (2012). Recommender systems. *Physics Reports*, 519(1):1–49.
- [Lu et al., 2014] Lu, M., Qin, Z., Cao, Y., Liu, Z., and Wang, M. (2014). Scalable news recommendation using multi-dimensional similarity and jaccard-kmeans clustering. *Journal of Systems and Software*, 95:242 – 251.
- [Lucas et al., 2013] Lucas, J. P., Luz, N., Moreno, M. N., Anacleto, R., Figueiredo, A. A., and Martins, C. (2013). A hybrid recommendation approach for a tourism system. *Expert Systems with Applications*, 40(9):3532 – 3550.
- [Luo et al., 2017] Luo, M., Nie, F., Chang, X., Yang, Y., Hauptmann, A. G., and Zheng, Q. (2017). Probabilistic non-negative matrix factorization and its robust extensions for topic modeling. In *AAAI*, pages 2308–2314.
- [Mahara et al., 2016] Mahara, T. et al. (2016). A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment. *Procedia Computer Science*, 89:450–456.
- [Maulik and Bandyopadhyay, 2002] Maulik, U. and Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654.
- [Mirzaei et al., 2016] Mirzaei, S., Hamme, H. V., and Norouzi, Y. (2016). Under-determined reverberant audio source separation using bayesian non-negative matrix factorization. *Speech Communication*, 81(Supplement C):129 – 137. Phase-Aware Signal Processing in Speech Communication.
- [Murtagh, 1983] Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal*, 26(4):354–359.
- [Najafabadi et al., 2017] Najafabadi, M. K., Mahrin, M. N., Chuprat, S., and Sarkan, H. M. (2017). Improving the accuracy of collaborative filtering recommendations using clustering and associa-

- tion rules mining on implicit data. *Computers in Human Behavior*, 67(Supplement C):113 – 128.
- [Noh et al., 2018] Noh, G., Oh, H., and Lee, J. (2018). Power users are not always powerful: The effect of social trust clusters in recommender systems. *Information Sciences*, 462:1 – 15.
- [Ortega et al., 2016] Ortega, F., Hernando, A., Bobadilla, J., and Kang, J. H. (2016). Recommending items to group of users using matrix factorization based collaborative filtering. *Information Sciences*, 345(Supplement C):313 – 324.
- [Park et al., 2012] Park, D. H., Kim, H. K., Choi, I. Y., and Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059 – 10072.
- [Patra et al., 2015] Patra, B. K., Launonen, R., Ollikainen, V., and Nandi, S. (2015). A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. *Knowledge-Based Systems*, 82:163–177.
- [Qu et al., 2013] Qu, W., Song, K.-S., Zhang, Y.-F., Feng, S., Wang, D.-L., and Yu, G. (2013). A novel approach based on multi-view content analysis and semi-supervised enrichment for movie recommendation. *Journal of Computer Science and Technology*, 28(5):776–787.
- [Rad and Jamzad, 2017] Rad, R. and Jamzad, M. (2017). Image annotation using multi-view non-negative matrix factorization with different number of basis vectors. *Journal of Visual Communication and Image Representation*, 46(Supplement C):1 – 12.
- [Rana and Jain, 2014] Rana, C. and Jain, S. K. (2014). An evolutionary clustering algorithm based on temporal features for dynamic recommender systems. *Swarm and Evolutionary Computation*, 14(Supplement C):21 – 30.
- [Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM.
- [Ricci et al., 2015a] Ricci, F., Rokach, L., and Shapira, B., editors (2015a). *Advances in collaborative Filtering*, chapter 3, pages 77–118. Springer US.
- [Ricci et al., 2015b] Ricci, F., Rokach, L., and Shapira, B., editors (2015b). *A Comprehensive Survey of Neighborhood-Based Recommendation Methods*, chapter 2, pages 37–76. Springer US.
- [Ricci et al., 2015c] Ricci, F., Rokach, L., and Shapira, B., editors (2015c). *Evaluating Recommender Systems*, chapter 8, pages 265–308. Springer US.
- [Ricci et al., 2015d] Ricci, F., Rokach, L., and Shapira, B., editors (2015d). *Recommender Systems: Introduction and Challenges*, chapter 1, pages 1–34. Springer US.

- [Salah et al., 2016] Salah, A., Rogovschi, N., and Nadif, M. (2016). A dynamic collaborative filtering system via a weighted clustering approach. *Neurocomputing*, 175(Part A):206 – 215.
- [Salakhutdinov and Mnih, 2007] Salakhutdinov, R. and Mnih, A. (2007). Probabilistic matrix factorization advances in neural information processing systems (nips 2007). *Vancouver, Canada*, pages 1–8.
- [Sanchez et al., 2008] Sanchez, J., Serradilla, F., Martinez, E., and Bobadilla, J. (2008). Choice of metrics used in collaborative filtering and their impact on recommender systems. In *Digital Ecosystems and Technologies, 2008. DEST 2008. 2nd IEEE International Conference on*, pages 432–436. IEEE.
- [Sarwar et al., 2002] Sarwar, B. M., Karypis, G., Konstan, J., and Riedl, J. (2002). Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324.
- [Saxena et al., 2017] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., and Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267:664 – 681.
- [Schachtner et al., 2014] Schachtner, R., Pöppel, G., Tomé, A., Puntonet, C., and Lang, E. (2014). A new bayesian approach to nonnegative matrix factorization: Uniqueness and model order selection. *Neurocomputing*, 138(Supplement C):142 – 156.
- [Schedl and Hauger, 2015] Schedl, M. and Hauger, D. (2015). Tailoring music recommendations to users by considering diversity, mainstreaminess, and novelty. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, pages 947–950, New York, NY, USA. ACM.
- [Schwartz, 2004] Schwartz, B. (2004). *The paradox of choice : why more is less*. HarperCollins Publishers Inc., New York.
- [Seret et al., 2014] Seret, A., Verbraken, T., and Baesens, B. (2014). A new knowledge-based constrained clustering approach: Theory and application in direct marketing. *Applied Soft Computing*, 24:316 – 327.
- [Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.
- [Shinde and Kulkarni, 2012] Shinde, S. K. and Kulkarni, U. (2012). Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems*

- with Applications*, 39(1):1381 – 1387.
- [Siddiqi and Sait, 2017] Siddiqi, U. F. and Sait, S. M. (2017). A new heuristic for the data clustering problem. *IEEE Access*, 5:6801–6812.
- [Su et al., 2016] Su, K., Ma, L., Xiao, B., and Zhang, H. (2016). Web service qos prediction by neighbor information combined non-negative matrix factorization. *Journal of Intelligent and Fuzzy Systems*, 30(6):3593–3604.
- [Su and Khoshgoftaar, 2009] Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2.
- [Sun and Dong, 2017] Sun, B. and Dong, L. (2017). Dynamic model adaptive to user interest drift based on cluster and nearest neighbors. *IEEE Access*, 5:1682–1691.
- [Sun et al., 2015] Sun, Q., Lu, J., Wu, Y., Qiao, H., Huang, X., and Hu, F. (2015). Non-informative hierarchical bayesian inference for non-negative matrix factorization. *Signal Processing*, 108(Supplement C):309 – 321.
- [Tan et al., 2011] Tan, S., Bu, J., Chen, C., Xu, B., Wang, C., and He, X. (2011). Using rich social media information for music recommendation via hypergraph model. *ACM Trans. Multimedia Comput. Commun. Appl.*, 7S(1):22:1–22:22.
- [Terán et al., 2018] Terán, L., Mensah, A. O., and Estorelli, A. (2018). A literature review for recommender systems techniques used in microblogs. *Expert Systems with Applications*, 103:63 – 73.
- [The Netflix Prize, 2006] The Netflix Prize (2006). Netflix dataset. <https://www.netflixprize.com/>.
- [Turcotte et al., 2015] Turcotte, J., York, C., Irving, J., Scholl, R. M., and Pingree, R. J. (2015). News recommendations from social media opinion leaders: Effects on media trust and information seeking. *Journal of Computer-Mediated Communication*, 20(5):520–535.
- [Vilar et al., 2018] Vilar, J. A., Lafuente-Rego, B., and D’Urso, P. (2018). Quantile autocovariances: A powerful tool for hard and soft partitional clustering of time series. *Fuzzy Sets and Systems*, 340:38 – 72. Theme: Clustering.
- [Virtanen et al., 2008] Virtanen, T., Cemgil, A. T., and Godsill, S. (2008). Bayesian extensions to non-negative matrix factorisation for audio signal modelling. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1825–1828.
- [Vishnuvarthanan et al., 2016] Vishnuvarthanan, G., Rajasekaran, M. P., Subbaraj, P., and Vishnuvarthanan, A. (2016). An unsupervised learning method with a clustering approach for tumor identification and tissue segmentation in magnetic resonance brain images. *Applied Soft Computing*, 38:190 – 212.

- [Wang et al., 2016] Wang, X., Cao, X., Jin, D., Cao, Y., and He, D. (2016). The (un)supervised nmf methods for discovering overlapping communities as well as hubs and outliers in networks. *Physica A: Statistical Mechanics and its Applications*, 446(Supplement C):22 – 34.
- [Wang and Zhang, 2014] Wang, X. and Zhang, J. (2014). Using incremental clustering technique in collaborative filtering data update. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pages 420–427.
- [Wen and Celebi, 2011] Wen, Q. and Celebi, M. E. (2011). Hard versus fuzzy c-means clustering for color quantization. *EURASIP Journal on Advances in Signal Processing*, 2011(1):118.
- [Winoto and Tang, 2010] Winoto, P. and Tang, T. Y. (2010). The role of user mood in movie recommendations. *Expert Systems with Applications*, 37(8):6086 – 6092.
- [Witten et al., 2016] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.
- [Wu et al., 2009] Wu, J., Xiong, H., and Chen, J. (2009). Adapting the right measures for k-means clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09*, pages 877–886, New York, NY, USA. ACM.
- [Wu and Yang, 2002] Wu, K.-L. and Yang, M.-S. (2002). Alternative c-means clustering algorithms. *Pattern Recognition*, 35(10):2267 – 2278.
- [Yang et al., 2002] Yang, J., Wang, W., Wang, H., and Yu, P. (2002). delta;-clusters: capturing subspace correlation in a large data set. In *Proceedings 18th International Conference on Data Engineering*, pages 517–528.
- [Yoo and Choi, 2009] Yoo, J. and Choi, S. (2009). *Weighted Nonnegative Matrix Co-Tri-Factorization for Collaborative Prediction*, pages 396–411. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Zafari and Moser, 2017] Zafari, F. and Moser, I. (2017). Modelling socially-influenced conditional preferences over feature values in recommender systems based on factorised collaborative filtering. *Expert Systems with Applications*, 87(Supplement C):98 – 117.
- [Zahra et al., 2015] Zahra, S., Ghazanfar, M. A., Khalid, A., Azam, M. A., Naeem, U., and Prugel-Bennett, A. (2015). Novel centroid selection approaches for kmeans-clustering based recommender systems. *Information sciences*, 320:156–189.
- [Zarzour et al., 2018] Zarzour, H., Maazouzi, F., Soltani, M., and Chemam, C. (2018). An improved collaborative filtering recommendation algorithm for big data. In *Computational Intelligence and Its Applications: 6th IFIP TC 5 International Conference, CIIA 2018, Oran,*

- Algeria, May 8-10, 2018, Proceedings 6*, pages 660–668. Springer.
- [Zhang et al., 2016a] Zhang, C., Wang, H., Yang, S., and Gao, Y. (2016a). Incremental nonnegative matrix factorization based on matrix sketching and k-means clustering. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 426–435. Springer.
- [Zhang et al., 2014] Zhang, C.-X., Zhang, Z.-K., Yu, L., Liu, C., Liu, H., and Yan, X.-Y. (2014). Information filtering via collaborative user clustering modeling. *Physica A: Statistical Mechanics and its Applications*, 396(Supplement C):195 – 203.
- [Zhang et al., 2016b] Zhang, F., Gong, T., Lee, V. E., Zhao, G., Rong, C., and Qu, G. (2016b). Fast algorithms to evaluate collaborative filtering recommender systems. *Knowledge-Based Systems*, 96:96 – 103.
- [Zhang et al., 2017a] Zhang, F., Lu, Y., Chen, J., Liu, S., and Ling, Z. (2017a). Robust collaborative filtering based on non-negative matrix factorization and r_1 -norm. *Knowledge-Based Systems*, 118(Supplement C):177 – 190.
- [Zhang et al., 2017b] Zhang, F., Lu, Y., Chen, J., Liu, S., and Ling, Z. (2017b). Robust collaborative filtering based on non-negative matrix factorization and r_1 -norm. *Knowledge-Based Systems*, 118:177 – 190.
- [Zhang et al., 2016c] Zhang, G., He, M., Wu, H., Cai, G., and Ge, J. (2016c). Non-negative multiple matrix factorization with social similarity for recommender systems. In *2016 IEEE/ACM 3rd International Conference on Big Data Computing Applications and Technologies (BDCAT)*, pages 280–286.
- [Zhang et al., 2016d] Zhang, X., Zong, L., Liu, X., and Luo, J. (2016d). Constrained clustering with nonnegative matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 27(7):1514–1526.
- [Zhang et al., 2017c] Zhang, Y., Chen, M., Huang, D., Wu, D., and Li, Y. (2017c). idoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization. *Future Generation Computer Systems*, 66(Supplement C):30 – 35.
- [Zhao et al., 2009] Zhao, Q., Xu, M., and Fränti, P. (2009). Sum-of-squares based cluster validity index and significance analysis. In *ICANNGA*, volume 5495, pages 313–322. Springer.