

## Tesis de Maestría

# Inducción de preferencias a partir del contexto de elección del usuario en sistemas de recomendación

Abalde, Roberto

2014-02-28

Este documento forma parte de la colección de tesis doctorales y de maestría de la Biblioteca Central Dr. Luis Federico Leloir, disponible en [digital.bl.fcen.uba.ar](http://digital.bl.fcen.uba.ar). Su utilización debe ser acompañada por la cita bibliográfica con reconocimiento de la fuente.

This document is part of the doctoral theses collection of the Central Library Dr. Luis Federico Leloir, available in [digital.bl.fcen.uba.ar](http://digital.bl.fcen.uba.ar). It should be used accompanied by the corresponding citation acknowledging the source.

Cita tipo APA:

Abalde, Roberto. (2014-02-28). Inducción de preferencias a partir del contexto de elección del usuario en sistemas de recomendación. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires.

Cita tipo Chicago:

Abalde, Roberto. "Inducción de preferencias a partir del contexto de elección del usuario en sistemas de recomendación". Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires. 2014-02-28.

**EXACTAS** UBA

Facultad de Ciencias Exactas y Naturales



**UBA**

Universidad de Buenos Aires



Universidad de Buenos Aires  
Facultad de Ciencias Exactas y Naturales  
Departamento de Computación

# **Inducción de preferencias a partir del contexto de elección del usuario en sistemas de recomendación**

Tesis presentada para obtener el título de  
*Magíster en Explotación de Datos y Descubrimiento de Conocimiento*

Roberto Abalde

Director: Dr. Marcelo Armentano

Buenos Aires, 2013



# Resumen

La mayoría de los sistemas de recomendación actuales que usan técnicas de filtrado colaborativo basado en memoria predicen las preferencias de un usuario usando solamente la información de los ítems sobre los que el usuario expresó su preferencia. Sin embargo no utilizan el contexto de elección del usuario, esto es, el conjunto de ítems que fueron recomendados al usuario pero que fueron ignorados.

En este trabajo de tesis presentamos las modificaciones hechas a los algoritmos clásicos de filtrado colaborativo basado en memoria para que utilicen el contexto de elección del usuario al momento de predecir sus preferencias por nuevos ítems. Como no se conocen conjuntos de datos públicos que tengan el contexto de elección del usuario y que nos permitan probar los algoritmos modificados, implementamos dichas modificaciones en un sistema de recomendación real para poder recolectar los datos necesarios para la etapa de experimentación. Finalmente, realizamos una serie de experimentos sobre tres conjuntos de datos, que nos permitieron verificar que nuestra propuesta tiene un mejor desempeño que los sistemas de recomendación clásicos.



# Abstract

Most of existing recommendation systems use memory-based collaborative filtering techniques to predict the preferences of a user using only the information of the items for which the user expressed a preference. However, existent systems do not use the context of user choice, that is, the set of items that were recommended to the user but were ignored by her.

In this thesis we present the modifications made to the classic memory-based collaborative filtering algorithms to use the context of user choice in the prediction of the user preferences. Since there are no known public datasets including the context of user choice that could allow us to test the modified algorithms, we implemented these changes in a real recommender system that allowed us to build a dataset with the information required. Finally, we conducted a series of experiments on three datasets, which allowed us to verify that our proposal has better performance than the traditional recommendation systems.



# Índice

---

1.	Introducción.....	9
1.1	Los sistemas de recomendación.....	9
1.2	Objetivo del trabajo .....	10
1.3	Organización de la tesis .....	11
2.	Los sistemas de recomendación .....	12
2.1	La larga cola.....	12
2.2	Aplicaciones de sistemas de recomendación .....	12
2.3	Definición del problema .....	15
2.4	Información de las utilidades asignadas .....	15
2.5	Recomendaciones basadas en contenido .....	16
2.6	Filtrado colaborativo.....	18
2.7	Aproximaciones híbridas .....	23
3.	Inducción de preferencias a partir del contexto de elección del usuario..	25
3.1	La información de contexto .....	25
3.2	El contexto de elección del usuario .....	26
3.3	Adaptación de los métodos de filtrado colaborativo .....	27
3.4	Implementación .....	29
4.	Experimentos, resultados y discusión.....	33
4.1	Conjuntos de datos.....	33
4.2	Métricas de evaluación .....	35
4.3	Protocolo de evaluación.....	37
4.4	Resultados y discusión.....	38
5.	Conclusiones y trabajos futuros.....	59

5.1	Contribuciones realizadas .....	59
5.2	Trabajos futuros .....	60
6.	Bibliografía.....	61
7.	Agradecimientos.....	64

# 1. Introducción

---

En este capítulo introductorio presentamos los sistemas de recomendación, discutimos su importancia y cómo proponemos mejorar su desempeño con nuestro trabajo de investigación.

## 1.1 Los sistemas de recomendación

Obtener buenas recomendaciones se ha convertido en un componente crítico dentro del proceso de toma de decisiones ya que el crecimiento de internet ha permitido presentar a los usuarios una cantidad cada vez mayor de alternativas entre las que elegir.

Este problema ha impulsado la investigación y desarrollo de los sistemas de recomendación [1], un conjunto de técnicas y herramientas de software con el objetivo de asistir a un usuario durante la toma de decisiones, recomendándole ítems que le podrían ser interesantes, por ejemplo qué productos comprar, qué música escuchar o qué noticias leer.

Las técnicas de recuperación de la información también ayudan a un usuario a encontrar ítems interesantes; sin embargo, a diferencia de los sistemas de recomendación, lo hacen a partir de una consulta explícita que expresa las necesidades de información del usuario.

El problema de hacer recomendaciones puede ser pensado como un aprendizaje de preferencias [2] ya que los sistemas de recomendación realizan sus recomendaciones utilizando modelos predictivos de las preferencias que los usuarios tienen sobre los ítems. Dichas preferencias son inducidas a partir de la información del usuario, de los ítems y de las interacciones entre los usuarios y el sistema. Por ejemplo, un sitio de alquiler películas podría desarrollar un sistema de recomendación para ayudar a sus clientes a encontrar las próximas películas que estarían interesados en ver, basándose en la edad y el sexo de los clientes, el país de origen, año de estreno y género de las películas y de las opiniones expresadas por el usuario sobre las películas que conoce.

Las interacciones entre los usuarios y los sistemas de recomendación se desarrollan en dos partes: primero, el sistema le recomienda al usuario una lista de ítems

y luego el usuario expresa sus preferencias sobre los ítems recomendados seleccionando un conjunto (posiblemente vacío) de ítems.

Por ejemplo, supongamos que un usuario visita un sitio de alquiler de películas y que el sistema le recomienda la lista que se muestra en la Figura 1-1, en base a la información de las películas, del usuario y de los alquileres realizados anteriormente por el usuario. Después de analizar la lista, supongamos que el usuario selecciona las películas 1, 3 y 7 (marcadas con un corazón y numeradas de izquierda a derecha). En este caso, el sistema podría inferir que los ítems seleccionados son más relevantes que los ítems ignorados (2, 4, 5 y 6) para el usuario, ya que expresó su preferencia por ellos.



Figura 1-1 – Películas recomendadas por el sistema y las seleccionadas por el usuario

La mayoría de las aproximaciones actuales al problema de recomendación utilizan sólo la información relacionada con los ítems seleccionados para inducir modelos de preferencia, desaprovechando la información relativa a los ítems ignorados. En la siguiente sección definiremos como objetivo de este trabajo aprovechar dicha información con el fin de mejorar el desempeño de los sistemas de recomendación actuales.

## 1.2 Objetivo del trabajo

El objetivo de esta tesis es verificar que el *contexto de elección del usuario*, es decir el conjunto de ítems que le fueron recomendados al usuario pero no seleccionados, puede ser utilizado para mejorar el desempeño de los sistemas de recomendación.

Para cumplir con el objetivo propuesto modificamos y adaptamos las técnicas clásicas de filtrado colaborativo basado en memoria de forma que utilicen el contexto de elección del usuario en la predicción de preferencias.

Como no se conocen conjuntos de datos públicos que tengan el contexto de elección del usuario y que nos permitan probar los algoritmos modificadas, por un lado hemos adaptado conjuntos de datos existentes y por otro hemos desarrollado un sistema de recomendación que almacena los contextos de elección.

Por último, realizamos una serie de experimentos para comparar el desempeño entre las técnicas de recomendación que usan el contexto de elección del usuario y las técnicas clásicas.

### **1.3 Organización de la tesis**

Luego de haber introducido el problema a tratar, analizamos el marco teórico de los sistemas de recomendación en el Capítulo 2, describiendo el origen del problema y ejemplos de uso y analizando las técnicas desarrolladas hasta el momento. En el Capítulo 3 detallamos el aporte de este trabajo de tesis, analizamos los trabajos de investigación relacionados y explicamos la solución propuesta y su implementación. En el Capítulo 4 explicamos los experimentos realizados y discutimos los resultados obtenidos. Finalmente, en el Capítulo 5, presentamos nuestras conclusiones y describimos posibles trabajos futuros que pueden ser derivados a partir del presente trabajo.

## 2. Los sistemas de recomendación

---

En este capítulo describimos en detalle los sistemas de recomendación, definiendo el problema que resuelven, las fuentes de datos que utilizan y las técnicas que implementan.

### 2.1 La larga cola

Las tiendas físicas se ven limitadas por la escasez de espacio y pueden mostrar al cliente sólo una pequeña parte de todas las opciones de productos existentes. Por el contrario, en el mundo virtual es posible mostrar el catálogo completo de productos al cliente. Mientras que una librería física sólo puede tener varios miles de libros en sus estantes, Amazon puede ofrecer millones. O un periódico en papel sólo puede imprimir varias decenas de artículos diariamente, mientras que los servicios de noticias en línea pueden ofrecer miles. Esta distinción entre el mundo físico y el virtual se conoce como el fenómeno de la *larga cola* [3].

Este fenómeno hace que en el mundo físico no sea posible adaptar la tienda a cada cliente en particular, por lo que las recomendaciones son bastante simples: se recomiendan productos por su popularidad. Por ejemplo, una librería mostrará sólo los libros más populares según las ventas y un diario imprimirá sólo los artículos que su editor considera que le interesan a la mayoría de la gente.

En el mundo virtual, aunque es posible ofrecerle todos los productos al cliente, no hay forma de mostrárselos todos a la vez ni se puede suponer que conoce todos los productos que podrían interesarle. Estas razones motivaron el surgimiento de los sistemas de recomendación [1] que permiten adaptar la tienda a cada cliente al hacerle recomendaciones de aquellos productos en los que posiblemente va a estar interesado.

### 2.2 Aplicaciones de sistemas de recomendación

Algunos de los sitios más visitados en internet usan sistemas de recomendación para ayudar al usuario en los procesos de decisión.

Uno de los sistemas de recomendación más conocido es el desarrollado por Amazon<sup>1</sup> que lo utiliza para ofrecer productos relacionados a los que el cliente ha mostrado interés. La Figura 2-1 muestra, marcada en rojo, la recomendación de libros relacionados que se compraron junto con el que se está mostrando en detalle.

Click to **LOOK INSIDE!**

**RECOMMENDER SYSTEMS HANDBOOK**

Click to open expanded view

Share your own customer images  
Search inside this book

**Recommender Systems Handbook [Hardcover]**  
 Francesco Ricci (Editor), Lior Rokach (Editor), Bracha Shapira (Editor), Paul B. Kantor (Editor)  
 ★★★★★ (1 customer review)

List Price: ~~\$229.00~~  
 Price: **\$171.31** & **FREE Shipping**. [Details](#)  
 You Save: **\$57.69 (25%)**

**Only 8 left in stock (more on the way).**  
 Ships from and sold by Amazon.com. Gift-wrap available.

Want it Friday, Dec. 6? Order within **18 hrs 44 mins** and choose **One-Day Shipping** at checkout. [Details](#)  
**Ordering for Christmas?** To ensure delivery by December 24 choose **FREE Shipping** at checkout. [Read more about holiday shipping.](#)

37 new from \$166.23 21 used from \$168.35

**FREE TWO-DAY SHIPPING FOR COLLEGE STUDENTS**  
[Learn more](#) **amazonstudent**

Formats	Amazon Price	New from	Used from
Kindle Edition	Rent from \$49.50 \$23.96	--	--
<b>Hardcover</b>	<b>\$171.31</b>	<b>\$166.23</b>	<b>\$168.35</b>
Paperback	--	--	--

**Customers Who Bought This Item Also Bought**

<p><b>Recommender Systems: An Introduction</b>            Dietmar Jannach            ★★★★★ (4)            Hardcover            \$65.61 <b>Prime</b></p>	<p><b>Machine Learning: A Probabilistic ...</b>            Kevin P. Murphy            ★★★★★ (23)            Hardcover            \$81.00 <b>Prime</b></p>	<p><b>Algorithms of the Intelligent Web</b>            Haralambos Marmanis            ★★★★★ (16)            Paperback            \$26.78 <b>Prime</b></p>	<p><b>Data Mining: Practical Machine Learning ...</b>            Ian H. Witten            ★★★★★ (33)            Paperback            \$37.80 <b>Prime</b></p>	<p><b>Introduction to Information Retrieval</b>            Christopher D. Manning            ★★★★★ (18)            Hardcover            \$58.28 <b>Prime</b></p>	<p><b>Bayesian Reasoning and Machine Learning</b>            David Barber            ★★★★★ (9)            Hardcover            \$59.44 <b>Prime</b></p>
---	---	---	---	--	---

Figura 2-1 – El sistema de recomendación de productos de Amazon

Otro conocido caso de implementación de sistemas de recomendación es el de la empresa de alquiler de películas Netflix<sup>2</sup>. El sistema (Figura 2-2) se encarga de ayudar al usuario a elegir las próximas películas que estaría interesado en ver.

<sup>1</sup> <http://www.amazon.com>

<sup>2</sup> <http://www.netflix.com>

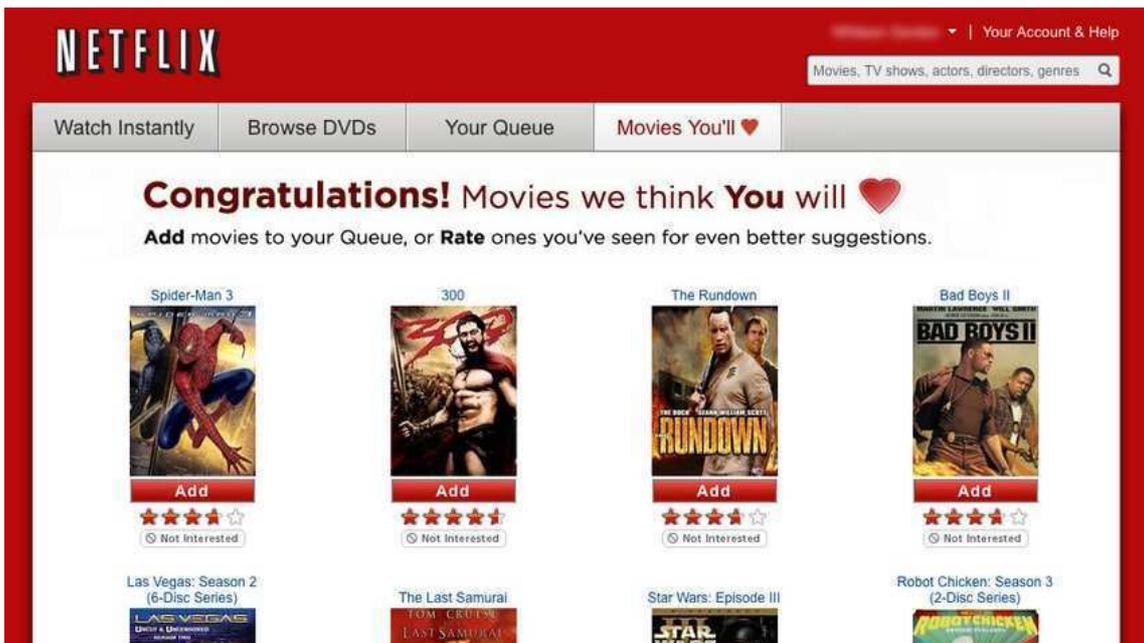


Figura 2-2 – El sistema de recomendación de películas de Netflix

Además de recomendar productos, los sistemas de recomendación han sido implementados para ayudar en la investigación.

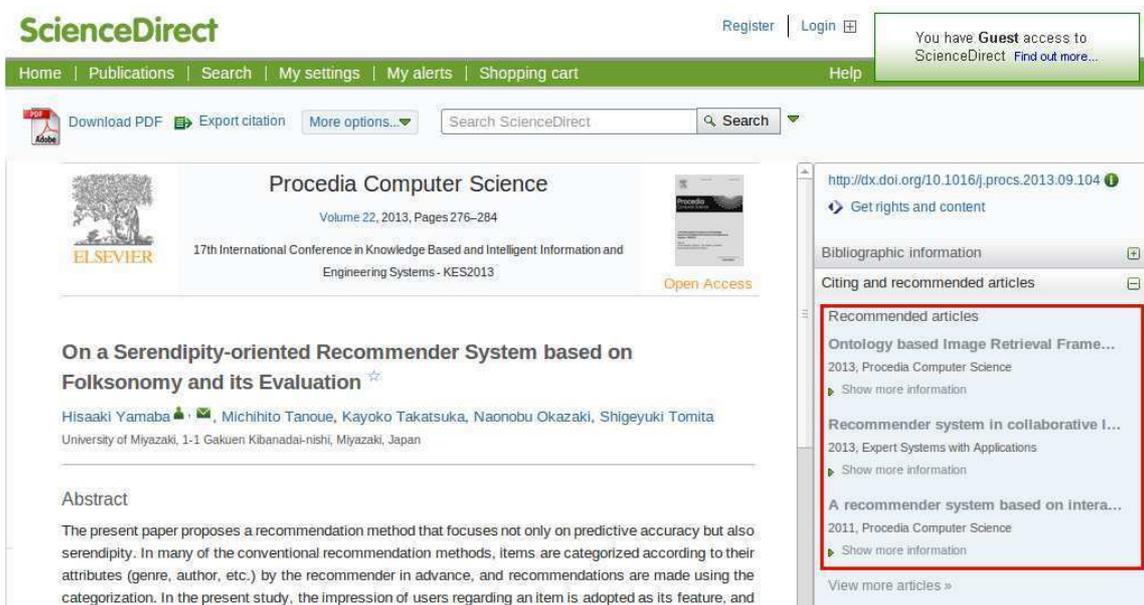


Figura 2-3 – El sistema de recomendación de artículos científicos de ScienceDirect

Por ejemplo, el sitio de publicaciones científicas ScienceDirect<sup>3</sup> recomienda artículos científicos (marcado en rojo en la Figura 2-3) relacionados con el que se está observando, ayudando en su trabajo a los investigadores.

## 2.3 Definición del problema

Los sistemas de recomendación suelen tener dos clases de entidades: los ítems que van a ser recomendados y los usuarios que son los destinatarios de las recomendaciones.

Los usuarios tienen preferencias sobre los ítems y aquellas conocidas son representadas en una *matriz de utilidad* (Figura 2-4) que asigna a cada par usuario-ítem un valor que representa el grado de preferencia de ese usuario por ese ítem.

		Ítems					
		1	2	...	i	...	m
Usuarios	1	5	3		1	2	
	2		2				4
	:			5			
	u	3	4		2	1	
	:					4	
	n			3	2		

Figura 2-4- Matriz de utilidad representando calificaciones en una escala de 1 a 5

La matriz suele ser dispersa porque usualmente la mayoría de sus valores son desconocidos. Un valor desconocido implica que no tenemos ninguna información de la preferencia de un usuario por un ítem.

El objetivo de un sistema de recomendación es descubrir el subconjunto de preferencias desconocidas que tengan la mayor utilidad para cada usuario.

## 2.4 Información de las utilidades asignadas

En general, existen formas explícitas e implícitas de conocer la utilidad que le da un usuario a un ítem.

---

<sup>3</sup> <http://www.sciencedirect.com>

En la forma explícita se le pide al usuario que califique ítems en una escala, que puede ser numérica (por ejemplo, de 1 a 5), ordinal (por ejemplo, “de acuerdo”, “neutral” o “en desacuerdo”), binaria (por ejemplo, “me gusta” o “no me gusta”) o unaria (se conoce sólo cuando el usuario calificó positivamente un ítem). Los sistemas de recomendación de películas o libros suelen obtener las calificaciones de sus clientes de esta forma. La efectividad de esta aproximación es limitada porque usualmente los usuarios no desean calificar explícitamente (por el esfuerzo que requiere o por la pérdida de privacidad, por ejemplo) y la información de aquellos que las dieron puede estar sesgada por el hecho de provenir de gente que deseaba dar calificaciones.

Las calificaciones implícitas provienen de inferencias hechas a partir de la observación del comportamiento de los usuarios. Por ejemplo, cuando un usuario compra un libro, mira un video o lee una noticia se puede inferir que está expresando su interés por ese ítem. En este caso la escala de calificaciones tiene un único valor (usualmente 1) que significa que el usuario ha expresado su preferencia por el ítem. Este tipo de calificaciones no tienen los sesgos mencionados para las explícitas y requieren menos esfuerzo por parte del usuario. Sin embargo presenta el problema de que se infiere sólo calificaciones positiva, más allá de si finalmente al usuario le gustó el ítem que seleccionó.

A continuación presentaremos los dos enfoques generales clásicos que existen para diseñar sistemas de recomendación: Los sistemas basados en contenido y los sistemas basados en filtrado colaborativo que se enfocan en las relaciones entre los usuarios y los ítems.

## **2.5 Recomendaciones basadas en contenido**

Los sistemas basados en contenido realizan recomendaciones utilizando tanto las características del usuario como de los ítems que le han interesado al usuario.

Por ejemplo, algunas de las características que pueden ser relevantes para recomendar películas podrían ser el director, el género, el año de estreno o los actores que participan en ella [4], mientras que para el usuario podría ser su información demográfica [5]. En este caso, cuando un usuario ha dado una buena calificación a las películas “Blade Runner” y “Star Wars” se podría inferir que tiene preferencia por las películas de ciencia ficción.

Gran parte de la investigación de este tipo de sistemas de recomendación se ha enfocado en recomendar ítems con contenido textual, tales como páginas web, libros y películas. Por eso, muchas aproximaciones han tratado este problema como una tarea de recuperación de información, donde el contenido asociado con las preferencias del usuario es tratado como una consulta y los ítems no calificados son puntuados por su relevancia para dicha consulta [6].

Una alternativa a las aproximaciones de recuperación de la información, es tratar la recomendación como una tarea de clasificación, donde el contenido de un ítem representa un ejemplo y las calificaciones realizadas por el usuario son usadas como etiquetas para la clasificación [7][8]. Por ejemplo, en la recomendación de libros presentada en [9] se usa el título, el autor, la sinopsis y las críticas para entrenar un clasificador bayesiano ingenuo multinomial.

Este tipo de técnicas de recomendación presentan algunos problemas:

- Efecto túnel: La tendencia a recomendar ítems cada vez más parecidos a los gustos del usuario, sin considerar otros ítems novedosos que tal vez sean de su agrado pero que el usuario desconoce o a los nunca fue expuesto [10].
- Usuario nuevo: Cuando un usuario no ha calificado ítems, el sistema no tiene información suficiente para realizar recomendaciones [11].
- Dependencia del dominio: Los modelos aprendidos no suelen mostrar buen desempeño en dominios diferentes de donde fue desarrollado, por ejemplo un sistema de este tipo para recomendación de noticias no muestra buen desempeño para recomendar artículos científicos [12].

Existen variantes del filtrado basado en contenido que usan otros tipos de información o conocimiento para realizar recomendaciones, tales como el conocimiento específico sobre cómo las características de los ítems satisfacen las preferencias de los usuarios o los perfiles de los amigos de los usuarios en las redes sociales [13]. Estas aproximaciones no utilizan las calificaciones del usuario para realizar las recomendaciones por lo que pueden ayudar a evitar los problemas vistos anteriormente.

## 2.6 Filtrado colaborativo

Los sistemas basados en filtrado colaborativo, utilizan solamente la información contenida en la matriz de utilidades para realizar recomendaciones y sus modelos se basan en las similitudes que presentan las calificaciones entre pares de usuarios o de ítems.

Los métodos de filtrado colaborativo pueden ser subdivididos a su vez en *basados en memoria* y *basados en modelos*, según la forma en que calculan las recomendaciones.

### 2.6.1 Filtrado colaborativo basado en memoria

El *filtrado colaborativo basado en memoria* utiliza la similitud entre usuarios o entre ítems para predecir la utilidad que le va a dar un usuario a un ítem no calificado anteriormente.

Estas técnicas [14] se basan en el supuesto de que dos usuarios con preferencias similares van a calificar el mismo ítem en forma similar o que dos ítems calificados por los usuarios en forma similar tendrán calificaciones similares en el futuro.

Para poder realizar las predicciones de las utilidades, estas técnicas crean un conjunto de vecinos usando una medida de similitud entre usuarios o ítems, seleccionando los  $k$  más cercanos o con los que se encuentren por debajo de un umbral de similitud dado.

Una medida de similitud muy utilizada es el coeficiente de correlación de Pearson [15] que se calcula como:

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (\text{ecuación 1})$$

Donde:

- $w_{u,v}$  es una medida de similitud en los usuarios  $u$  y  $v$
- $I$  es el conjunto de ítems calificados por los usuarios  $u$  y  $v$
- $r_{u,i}$  es el valor de la utilidad dada al ítem  $i$  por el usuario  $u$
- $\bar{r}_u$  es la media de las calificaciones hechas por el usuario  $u$

Alternativamente, se pueden tratar a las calificaciones de dos usuarios como vectores en un espacio  $m$ -dimensional y calcular la similitud basándose en el coseno del ángulo entre ellos:

$$w_{u,v} = \cos(\vec{r}_u, \vec{r}_v) = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\|_2 \times \|\vec{r}_v\|_2} = \frac{\sum_{i=1}^m r_{u,i} r_{v,i}}{\sqrt{\sum_{i=1}^m r_{u,i}^2} \sqrt{\sum_{i=1}^m r_{v,i}^2}} \quad (\text{ecuación 2})$$

Estudios empíricos [14] han mostrado que generalmente la correlación de Pearson se comporta mejor que la similitud coseno.

Cuando sólo se toma en cuenta los conjuntos de ítems calificados y no sus calificaciones o cuando las calificaciones son unarias se suele usar como medida el coeficiente de similitud de Jaccard:

$$w_{u,v} = \frac{|U \cap V|}{|U \cup V|} \quad (\text{ecuación 3})$$

Donde:

- $U$  es el conjunto de ítems calificados por el usuario  $u$
- $V$  es el conjunto de ítems calificados por el usuario  $v$

Existen otras medidas de similitud en la literatura, incluyendo la correlación de Spearman, la correlación  $\tau$  de Kendall, la media de las diferencias al cuadrado, la entropía y la similitud coseno ajustada [16][17].

A partir de las similitudes, se calculan las predicciones de las utilidades como la media ponderada de las desviaciones sobre la media de los vecinos:

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in K} (r_{v,i} - \bar{r}_v) w_{u,v}}{\sum_{v \in K} w_{u,v}} \quad (\text{ecuación 4})$$

Donde:

- $p_{u,i}$  es la predicción de la utilidad del ítem  $i$  para el usuario  $u$
- $w_{u,v}$  es la similitud entre los usuarios  $u$  y  $v$
- $K$  es el conjunto de usuarios similares a  $u$ , seleccionados por una cantidad de usuarios o por un umbral de similitud

En el caso de calificaciones unarias, la predicción se calcula como la media de las similitudes de los vecinos que calificaron el ítem:

$$p_{u,i} = \frac{\sum_{v \in K} S_{v,i} w_{u,v}}{\sum_{v \in K} w_{u,v}} \quad (\text{ecuación 5})$$

Donde:

- $p_{u,i}$  es la predicción de la utilidad del ítem  $i$  para el usuario  $u$
- $w_{u,v}$  es la similitud entre los usuarios  $u$  y  $v$
- $K$  es el conjunto de usuarios similares a  $u$ , seleccionados por cantidad o por un umbral de similitud
- $S_{v,i}$  es un indicador cuyo valor es 1 cuando el usuario  $u$  calificó el ítem  $i$ , sino es 0

En muchos sistemas de recomendación, la cantidad de usuarios supera ampliamente la de ítems. En estos casos los algoritmos basados en la similitud de usuarios muestran problemas de escalabilidad debido la complejidad computacional de buscar usuarios similares.

Para evitar este problema se desarrolló el filtrado colaborativo basado en la similitud entre ítems [18] que busca ítems similares a los calificados por el usuario en vez de buscar usuarios similares. El supuesto detrás de este enfoque es que los usuarios van a preferir ítems similares a los que ya les gustaron. En la práctica, esta aproximación lleva a recomendaciones más rápidas y usualmente mejores aunque presenta el problema de no permitirle al usuario descubrir ítems novedosos [18][19].

Bajo esta aproximación, el coeficiente de correlación de Pearson se calcula como:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2 \sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}} \quad (\text{ecuación 6})$$

Donde:

- $U$  es el conjunto de todos los usuarios que calificaron ambos ítems  $i$  y  $j$
- $r_{u,i}$  es la clasificación del ítem  $i$  hecha por el usuario  $u$
- $\bar{r}_i$  es la media de las clasificaciones para el ítem  $i$  de todos los usuarios

Y el coeficiente de similitud de Jaccard se calcula como:

$$w_{i,j} = \frac{|I \cap J|}{|I \cup J|} \quad (\text{ecuación 7})$$

Donde:

- $I$  es el conjunto de usuarios que calificaron el ítem  $i$
- $J$  es el conjunto de usuarios que calificaron el ítem  $j$

En el filtrado colaborativo basado en ítems, la utilidad del ítem  $i$  para un usuario  $u$  puede ser predicha usando una media ponderada simple:

$$p_{u,i} = \frac{\sum_{j \in K} r_{u,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|} \quad (\text{ecuación 8})$$

Donde:

- $p_{u,i}$  es la predicción de la utilidad del ítem  $i$  para el usuario  $u$
- $w_{i,j}$  es la similitud entre los ítems  $i$  y  $j$
- $K$  es el conjunto de los  $k$  ítems similares al ítem  $i$ , seleccionados por cantidad o por un umbral de similitud

En el caso de calificaciones unarias, el índice que sirve para ordenar los ítems a recomendar se calcula como:

$$p_{u,i} = \frac{\sum_{j \in K} S_{u,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|} \quad (\text{ecuación 9})$$

Donde:

- $p_{u,i}$  es la predicción de la utilidad del ítem  $i$  para el usuario  $u$
- $w_{i,j}$  es la similitud entre los ítems  $i$  y  $j$
- $S_{u,j}$  es un indicador cuyo valor es 1 cuando el usuario  $u$  calificó el ítem  $j$  sino es 0
- $K$  es el conjunto de los  $k$  ítems calificados por  $u$  que son similares a  $i$ , seleccionados por cantidad o por un umbral de similitud

Los sistemas que usan la técnica de filtrado colaborativo basado en memoria suelen presentar tres problemas:

- Arranque en frío: Este tipo de aproximaciones no puede hacer buenas predicciones para usuarios nuevos (cuando usan similitud entre usuarios) o ítems nuevos (cuando usan similitud entre ítems) ya que se requiere una gran cantidad de calificaciones para tener buenas recomendaciones [20].
- Escalabilidad: Cuando se tienen millones de usuarios (o ítems) se requiere una gran potencia de cálculo para calcular los usuarios (o ítems) similares y poder realizar las recomendaciones [21].
- Matrices dispersas: Usualmente el número de ítems y/o de usuarios es muy grande y el número de calificaciones suele ser pequeño, por lo que se cuenta con poca información para realizar recomendaciones [21].

Por otro lado, este método tiene dos ventajas: permite explicar fácilmente al usuario cómo se calculó la predicción de una utilidad y además puede ser entrenado en línea con un bajo costo computacional e ir mejorando la precisión de las predicciones.

### **2.6.2 El filtrado colaborativo basado en modelos**

El *filtrado colaborativo basado en modelos* estima las calificaciones de los usuarios usando modelos estadísticos. Una aproximación de este tipo sería construir un clasificador para cada usuario activo que represente los ítems seleccionados como características y utilice las calificaciones disponibles como etiquetas [22].

Más recientemente han surgido los modelos de factores latentes y la factorización de matrices y han mostrado buenos resultados [23]. A diferencia de los métodos basados en memoria, los modelos de factores latentes asumen que la similitud entre los usuarios y los ítems es inducida a partir de una estructura de baja dimensionalidad oculta en los datos. Por ejemplo, la utilidad que un usuario le da a una película se podría pensar que depende de algunos factores implícitos tales como la preferencia del usuario por un género cinematográfico o por un actor. Las técnicas de factorización de matrices son una clase de modelos de factores latentes en los que los usuarios y los ítems son representados como vectores de características desconocidos (vectores columna)  $w_u, h_i \in R^k$  a lo largo de  $k$  dimensiones latentes. Estos vectores de

características son aprendidos de modo que los productos internos  $w_u^T h_i$  se aproximan a las calificaciones conocidas  $r_{u,i}$  con respecto a una función de pérdida minimizando la siguiente función objetivo:

$$J(W, H) = \sum_{(u,i) \in L} (r_{u,i} - w_u^T h_i)^2 \quad (\text{ecuación 10})$$

Donde:

- $r_{u,i}$  es la clasificación del ítem  $i$  hecha por el usuario  $u$
- $W = [w_1 \dots w_n]^T$  es una matriz de  $n \times k$
- $H = [h_1 \dots h_n]$  es una matriz de  $k \times n$
- $L$  es el conjunto de pares usuario-ítem para los cuales las calificaciones son conocidas.

Los procedimientos estándar de optimización de la función objetivo no convexa  $J(W, H)$  incluyen técnicas basadas en gradientes y procedimientos como los mínimos cuadrados alternados. Otra clase de técnicas es la factorización de matrices no negativas [24] donde restricciones de no negatividad son puestas sobre  $W$  y  $H$ .

Si bien estas técnicas muestran buenos desempeños, presenta dos problemas: la dificultad de explicarle al usuario la predicción realizada derivada de la imposibilidad de explicar los factores latentes y el alto costo computacional de entrenar el modelo en línea por tener que recalcularse los factores latentes ante cada nueva utilidad expresada por el usuario.

## 2.7 Aproximaciones híbridas

La Tabla 2-1 muestra un resumen de los problemas de los sistemas de recomendación basados en contenido y en filtrado colaborativo.

Para evitar estos problemas se han desarrollado métodos híbridos que combinan las dos estrategias. El problema de recomendar ítems que no sean parecidos a los que el usuario suele calificar positivamente puede ser solucionado con el filtrado colaborativo, ya que, basándonos en el comportamiento pasado de otros usuarios podemos encontrar ítems novedosos para el usuario actual. El problema del arranque en frío que tienen los sistemas de filtrado colaborativo puede ser solucionado parcialmente aplicando las

técnicas basadas en contenido utilizando, por ejemplo, la información demográfica del usuario para encontrar los ítems calificados por usuarios similares.

Tabla 2-1 - Resumen de los problemas que presenta cada modelo

Problema	Modelo		
	Basado en contenido	Filtrado colaborativo	
		Basado en memoria	Basado en modelos
Efecto túnel	Sí	No	No
Usuario/ítem nuevo	No	Sí	Sí
Dependencia del dominio	Sí	No	No
Arranque en frío	Sí	Sí	Sí
Escalabilidad	No	Sí	Sí
Matrices dispersas	No	Sí	Sí
No poder explicar recomendaciones	No	No	Sí
No poder actualizar en línea	No	No	Sí

Para mejorar la escalabilidad y la dispersión se suelen usar, por ejemplo, técnicas de reducción de dimensionalidad y el análisis de conglomerados (*clustering*), que permiten disminuir la cantidad de ítems o de usuarios en la matriz de calificaciones [25].

## 3. Inducción de preferencias a partir del contexto de elección del usuario

---

En este capítulo presentamos nuestra propuesta de utilizar el contexto de elección de usuario para mejorar la predicción de las preferencias en los sistemas de recomendación que usan técnicas de filtrado colaborativo basado en memoria.

### 3.1 La información de contexto

En muchas aplicaciones de sistemas de recomendación no es suficiente tener en cuenta sólo los usuarios e ítems a la hora de dar recomendaciones, también es importante incorporar la *información de contexto* en el proceso de recomendación con el fin de recomendar ítems a usuarios bajo ciertas circunstancias [26].

Por ejemplo, saber el lugar donde se encuentra el usuario es importante para recomendarle películas que estén en la cartelera de los cines cercanos. Del mismo modo, las recomendaciones de restaurantes deberían variar dependiendo de la hora del día en que se quiere ir o de si el usuario desea ir a comer solo o acompañado.

Para representar la información de contexto se suele usar un espacio multidimensional que reemplaza la matriz de utilidades. Este espacio multidimensional tiene al usuario y al ítem como primeras dos dimensiones y luego un conjunto de dimensiones contextuales, donde cada dimensión captura un tipo particular de contexto [27] [28]. Por ejemplo, podrían tenerse dimensiones contextuales para el clima, la estación del año, el lugar (latitud y longitud) o el dispositivo que se usó cuando expresó la utilidad en el sistema de recomendación.

La información de contexto puede ser utilizada en tres etapas del proceso de recomendación:

- Pre-filtrado contextual: La información del contexto del usuario se usa para guiar la selección y construcción de los datos relevantes para la recomendación que luego son utilizados por un sistema de recomendación clásico [29].
- Post-filtrado contextual: Las recomendaciones se calculan con un sistema de recomendación clásico usando la matriz de utilidades e ignorando la información

de contexto y luego son ajustadas para cada usuario usando la información contextual [29].

- Modelado contextual: La información de contexto es usada directamente por la técnica de modelado como parte de la estimación de las preferencias [30].

La investigación sobre sistemas de recomendación contextuales suele tomar en cuenta usualmente el contexto de los usuarios o de los ítems a recomendar, pero no han utilizado la información del contexto en que el usuario realiza la elección efectiva de los ítems recomendados, puntualmente qué ítems le fueron recomendados e ignoró, lo que llamaremos el *contexto de elección del usuario*.

## 3.2 El contexto de elección del usuario

Una importante fuente de información contextual son las interacciones entre los usuarios y el sistema de recomendación.

Estas interacciones tienen dos pasos: primero el sistema le recomienda a un usuario una lista de ítems ordenada según la predicción de sus preferencias y luego, el usuario la analiza y selecciona un conjunto (posiblemente vacío) de ítems, expresando de esta manera sus preferencias acerca de los ítems recomendados.

En estas interacciones el usuario no está expresando una preferencia absoluta, esto es, que prefiere los ítems seleccionados sobre todo el resto de los existentes [31] porque no los conoce todos. En realidad está expresando que prefiere los ítems seleccionados sobre el resto de los recomendados, esto es, una preferencia relativa.

Las preferencias relativas se definen entre dos conjuntos de ítems recomendados: los que fueron seleccionados por el usuario y los que fueron ignorados. A este último conjunto lo denominaremos *contexto de elección del usuario*.

El contexto de elección del usuario es un tipo de información contextual no utilizada hasta hace poco tiempo en los sistemas de recomendación mientras que la información de los ítems seleccionados es usada por la mayoría.

En el ámbito de recuperación de la información se ha utilizado una idea similar al contexto de elección del usuario [32] aplicada a los buscadores: Cuando un buscador presenta una lista de enlaces a un usuario luego de realizar una consulta y el usuario

selecciona uno de ellos, se induce que el usuario está diciendo implícitamente que los enlaces anteriores al seleccionado no son importantes.

El primer uso conocido del contexto de elección del usuario en sistemas de recomendación [33] fue presentado como parte de un modelo de factores latentes (filtrado colaborativo basado en modelos) que deriva en un modelo *logit* multinomial.

El presente trabajo hace uso del contexto de elección del usuario para mejorar los sistemas de recomendación pero, a diferencia de los trabajos anteriores, se enfoca en adoptar las técnicas de filtrado colaborativo basado en memoria.

### 3.3 Adaptación de los métodos de filtrado colaborativo

En el filtrado colaborativo basado en usuarios clásico, la creación de la lista de ítems a recomendar comienza con la creación de un conjunto de ítems candidatos, formado por todos aquellos ítems existentes que el usuario no seleccionó en ninguna de sus interacciones con el sistema. Luego, para cada ítem del conjunto de candidatos, se calcula la predicción de su utilidad como la suma de las distancias de los usuarios similares que seleccionaron dicho ítem. Por último, se crea la lista de ítems a recomendar ordenando los ítems del conjunto de candidatos por la utilidad calculada.

Veamos este proceso en un ejemplo. Supongamos que todos los ítems en el sistema de recomendación son  $I = \{i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10}\}$  y que el usuario  $u$  seleccionó en interacciones anteriores los ítems  $S = \{i_3, i_5, i_7\}$ . Así, los ítems candidatos para  $u$  son  $C = I - S = \{i_1, i_2, i_4, i_6, i_8, i_9, i_{10}\}$ . Ahora, supongamos que la predicción de las utilidades los ítems candidatos es  $P = (0.2, 0.03, 0.37, 0.16, 0.1, 0.14, 0.0)$  donde  $P_j$  es la utilidad del ítem  $j$  en  $C$ . La predicción de utilidad de un ítem puede ser 0.0, como el caso de  $i_{10}$ , porque los vecinos más cercanos del usuario actual no seleccionaron nunca ese ítem. Luego, en base a las utilidades de  $P$ , ordenamos  $C$  y obtenemos la lista de ítems a recomendar  $R = (i_4, i_1, i_6, i_9, i_8, i_2, i_{10})$ .

Ahora, supongamos que los ítems ignorados por el usuario son  $T = \{i_1, i_9\}$ . Estos ítems deberían estar cerca de los últimos puestos de la lista de recomendaciones ya que si fueron ignorados por el usuario, podemos suponer que tienen menor o igual utilidad que los ítems nunca vistos y esto nos podría permitir hacer un mejor ordenamiento de  $C$ .

Entonces, para calcular la predicción de utilidad de un ítem  $i$  para un usuario  $u$ , tomando en cuenta los ítems ignorados, primero sumaremos las similitudes para los ítems seleccionados por los vecinos cercanos a  $u$ . A esta suma la ponderaremos con un parámetro  $\beta$  y si  $i$  fue ignorado por  $u$  le restaremos  $(1-\beta)$ .

El parámetro  $\beta$  es un valor entre 0 y 1 que sirve para balancear el peso de los ítems seleccionados sobre los ignorados.

Volviendo al ejemplo anterior, si tomamos  $\beta=0.9$  tenemos que  $P' = \beta P = (0.18, 0.027, 0.333, 0.144, 0.09, 0.126, 0.0)$ . Luego, si le restamos  $(1-\beta)$  a los ítems ignorados, tenemos  $P'' = (0.08, 0.027, 0.333, 0.144, 0.09, 0.026, 0.0)$  con lo que el ordenamiento de  $C$  sería  $R' = (i_4, i_6, i_8, i_2, i_1, i_9, i_{10})$  teniendo los ítems ignorados al final de la lista.

Esta modificación traducida al cálculo de la predicción de utilidad de un ítem para un usuario (ecuación 5) puede calcularse como:

$$p_{u,i} = \beta \frac{(\sum_{v \in K} S_{v,i} w_{u,v})}{\sum_{u \in K} w_{u,v}} - (1 - \beta) T_{u,i} \quad (\text{ecuación 11})$$

Donde:

- $p_{u,i}$  es la predicción de la utilidad del ítem  $i$  para el usuario  $u$
- $w_{u,v}$  es la similitud entre los usuarios  $u$  y  $v$
- $K$  es el conjunto de usuarios similares a  $u$ , seleccionados por cantidad o por un umbral de similitud
- $S_{v,i}$  es un indicador cuyo valor es 1 cuando el usuario  $v$  calificó el ítem  $i$ , sino es 0
- $T_{u,i}$  es un indicador cuyo valor es 1 cuando el usuario  $u$  ignoró el ítem  $i$ , sino es 0
- $\beta$  es un valor entre 0 y 1 que pondera los ítems seleccionados contra los ignorados

De forma análoga, modificamos la forma en que el filtrado colaborativo basado en ítems calcula la predicción de la utilidad (ecuación 9) de la siguiente manera:

$$p_{u,i} = \beta \frac{\sum_{j \in K} S_{u,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|} - (1 - \beta) T_{u,j} \quad (\text{ecuación 12})$$

Donde:

- $p_{u,i}$  es la predicción de la utilidad del ítem  $i$  para el usuario  $u$
- $w_{i,j}$  es la similitud entre los usuarios  $i$  y  $j$
- $S_{u,j}$  es un indicador cuyo valor es 1 cuando el usuario  $v$  calificó el ítem  $i$ , sino es 0
- $T_{v,i}$  es un indicador cuyo valor es 1 cuando el usuario  $u$  ignoró el ítem  $i$ , sino es 0
- $K$  es el conjunto de los  $k$  usuarios que calificaron  $i$  y que son similares a  $u$ , seleccionados por cantidad o por un umbral de similitud
- $\beta$  es el peso que se le da a los ítems seleccionados y a los ignorados

Cuando  $\beta=1.0$  no se realiza la penalización de los intereses seleccionados porque se anula el término que está restando, por lo que las ecuaciones modificadas son iguales a las originales. A estos casos los llamaremos “base” o “BL”.

### 3.4 Implementación

Para poder probar las modificaciones propuestas en la sección anterior en un entorno real, desarrollamos y pusimos en línea un sistema de recomendación de películas<sup>4</sup> utilizando Python 2.7<sup>5</sup>.

El sistema desarrollado (Figura 3-1) genera recomendaciones a partir de 8048 películas argentinas, españolas y estadounidenses, estrenadas entre 1970 y 2012. Esta selección responde a que los usuarios que probaron el sistema voluntariamente son, en su gran mayoría, alumnos argentinos de grado y posgrado y era necesario filtrar aquellas películas que ninguno de ellos iba a seleccionar para tener menos dispersión en la matriz de utilidades.

El sistema de recomendación implementa dos algoritmos de filtrado colaborativo por similitud de usuarios: el modelo clásico y el que refleja las modificaciones propuestas en esta tesis. Cuando un usuario se registra en el sistema, se le asigna automáticamente y al azar uno de los dos algoritmos implementados. De esta forma disponemos de un grupo de prueba y un grupo de control.

---

<sup>4</sup> <http://pelisquemegustan.inductia.com>

<sup>5</sup> <http://www.python.org>

La escala de las calificaciones utilizada por el sistema implementado es unaria debido al poco esfuerzo que dicha escala requiere del usuario para expresar sus preferencias. Además, usamos la distancia Jaccard para calcular los vecinos más cercanos.

En cada interacción el sistema le recomienda 8 películas al usuario. Se eligió esta cantidad en base a la teoría de que la memoria de trabajo de una persona puede albergar  $7 \pm 2$  objetos al mismo tiempo [34]. Entonces, podemos suponer que el usuario observa todos los ítems presentados, selecciona aquellos que le interesan e ignora el resto. Así, el contexto de elección del usuario va a tener entre 1 y 8 películas en cada interacción.

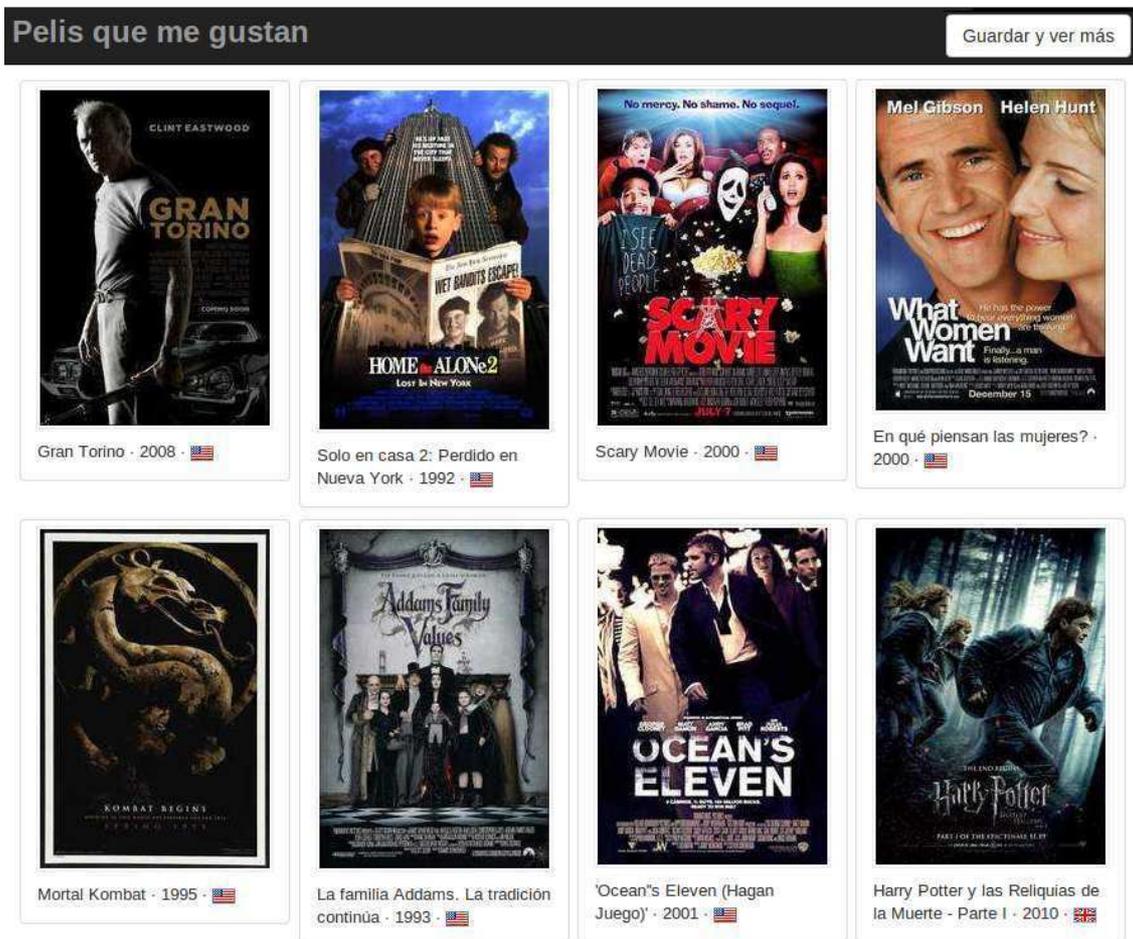


Figura 3-1 - Sistema de recomendación de películas desarrollado

Los ítems recomendados se muestran en la pantalla del sistema de acuerdo al siguiente orden: los primeros cuatro se muestran en la primera fila y los últimos cuatro en la segunda, manteniendo el orden de izquierda a derecha. El orden obedece a un estudio [35] que muestra que el movimiento de los ojos de los usuarios en los sitios web sigue un patrón en forma de F: primero miran dos bandas horizontales de izquierda a derecha, y luego una vertical a la izquierda de la pantalla de arriba a abajo.

Para seleccionar algunas de las películas recomendadas, el usuario debe hacer click sobre ellas y las selecciones se marcarán con un corazón (Figura 3-2). Luego debe presionar el botón “Guardar y ver más” para guardar sus selecciones (si hubo) y obtener más recomendaciones.

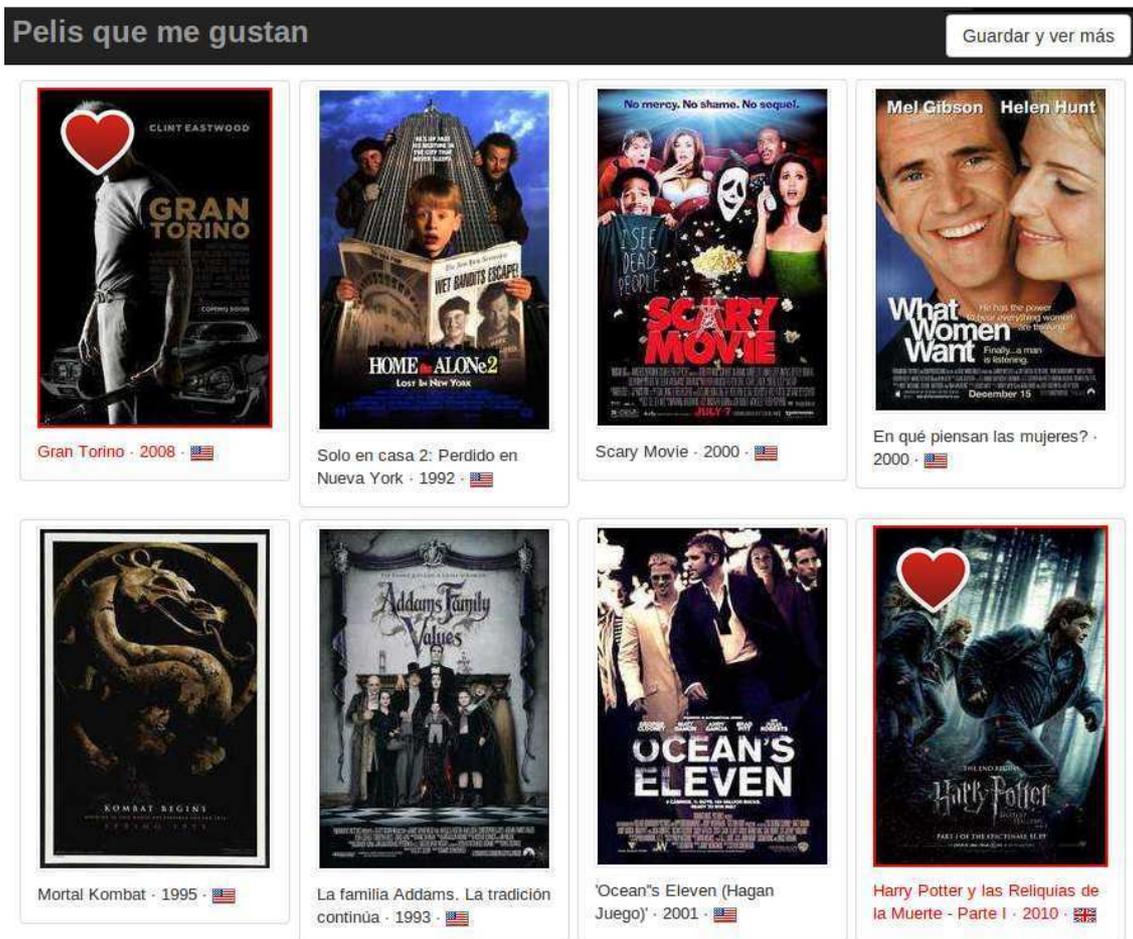


Figura 3-2- Sistema de recomendación de películas mostrando dos películas seleccionadas

Después de cada interacción con un usuario, el sistema almacena la identificación del usuario, la lista de ítems recomendados, la lista de los ítems seleccionados y la fecha y la hora en que se realizó la interacción.

Por ejemplo, cuando se guarden las selecciones que se muestran en la Figura 3-2, el sistema almacenará la siguiente tupla:

`(1234, [1, 2, 3, 4, 5, 6, 7, 8], [1, 8], '2013-12-13 18:45')`

Usando el conjunto de datos generados por el sistema de recomendación desarrollado y otros dos conjuntos de datos públicos, probamos el sistema modificado para verificar que mejora el desempeño del sistema de recomendación. En el siguiente capítulo describimos las pruebas realizadas y los resultados obtenidos.

## 4. Experimentos, resultados y discusión

---

En este capítulo describiremos los conjuntos de datos usados, las métricas de evaluación utilizadas y los experimentos realizados para probar los modelos modificados. Luego mostraremos y analizaremos los resultados obtenidos.

### 4.1 Conjuntos de datos

Para evaluar las modificaciones propuestas se utilizaron tres conjuntos de datos. Los conjuntos de datos *movielens* y *webscope* son públicos y son ampliamente utilizados para evaluar sistemas de recomendación. Estos conjuntos de datos no tienen el contexto de elección del usuario por lo que tuvimos que simularlo. El tercer conjunto de datos fue recolectado durante la interacción de los usuarios de prueba del sistema de recomendación desarrollado y por lo tanto contiene el contexto de elección del usuario por nosotros utilizando el sistema de recomendación que desarrollamos y tiene el contexto de elección del usuario.

#### 4.1.1 Conjunto de datos *movielens*

El conjunto de datos *movielens*<sup>6</sup> fue creado por el laboratorio GroupLens de la Universidad de Minnesota y es uno de los más conocidos y utilizados por la comunidad de investigación de sistemas de recomendación. *Movielens* tiene las preferencias de usuarios anónimos sobre un conjunto de películas y la fecha y hora de la interacción. Las preferencias están expresadas en una escala numérica de 1 a 5.

Para simular el contexto de elección del usuario utilizamos la misma política descrita en [33]: tomamos las calificaciones con valor '5' como los ítems seleccionados y el resto de las calificaciones como ítems ignorados. Los ítems que no tenían una valoración en el conjunto de datos original tampoco la van a tener en el conjunto modificado.

En este trabajo de tesis utilizamos los conjuntos de datos *movielens-1M* y *movielens-100k*. *Movielens-1M* que tiene 1000209 calificaciones de 3706 películas hechas por 6040 usuarios (esto es, 4.47% de las calificaciones posibles). Los usuarios

---

<sup>6</sup> <http://www.grouplens.org/datasets/movielens/>

en el conjunto de datos calificaron por lo menos 20 películas. Movielens-100k tiene 100000 calificaciones de 1682 películas hechas por 943 usuarios (6.3% de las calificaciones posibles).

El conjunto de datos movielens-1M con el contexto de elección simulado tiene 226310 ítems seleccionados (1.2% de los ítems seleccionados posibles) y 773899 ítems ignorados (3,4 ítems ignorados por cada seleccionado).

Con el contexto de elección simulado, el conjunto de datos movielens-100k tiene 21201 ítems seleccionados (1.34% del total) y 78799 ítems ignorados (3.7 ítems ignorados por cada seleccionado).

#### **4.1.2 Conjunto de datos webscope**

El conjunto de datos webscope<sup>7</sup> contiene 11557943 calificaciones (en una escala de 0 a 100) de 98211 artistas hechas por 1948882 usuarios anónimos.

Como la cantidad de calificaciones es demasiado grande tomamos una muestra al azar de 1000000 de opiniones de 16681 usuarios y 13928 ítems. (0.43% de calificaciones sobre el total).

Para simular el contexto de elección del usuario, se tomaron las calificaciones de 90 puntos o más como los ítems seleccionados dejando el resto de los calificados como ignorados. Se seleccionó este valor para obtener una proporción entre ítems seleccionados e ignorados similar al conjunto de datos movielens. Los ítems no calificados quedan igual. Como resultado obtuvimos 380379 calificaciones unarias (1.6% del total) y 619621 ítems ignorados (1.6 ignorados por cada seleccionado).

#### **4.1.3 Conjunto de datos “pelis que me gustan”**

El tercer conjunto de datos usado para las pruebas es el que creamos para este trabajo de investigación a partir del sistema de recomendación que implementamos.

El sistema de recomendación desarrollado asigna al azar uno de los algoritmos de recomendación a cada usuario cuando se registra en el sistema. Entonces, este conjunto de datos tiene por un lado los datos generados por el algoritmo de filtrado

---

<sup>7</sup> <http://webscope.sandbox.yahoo.com/catalog.php?datatype=r>

colaborativo de usuarios basado en memoria clásico y por otro los dos generados por el algoritmo que modificamos.

El sistema almacena por cada interacción, el ID del usuario participante, las 8 películas que fueron recomendadas, las películas que fueron seleccionadas y la fecha y hora.

El conjunto de datos del algoritmo clásico tiene 115 usuarios que seleccionaron la suma de 2975 películas (0.32% del total disponible) en 9107 interacciones con el sistema. Este conjunto de datos no tiene películas ignoradas ya que no era necesario para el algoritmo registrarlas.

El conjunto del algoritmo modificado tiene 114 usuarios que realizaron 7057 interacciones con el sistema, seleccionando 5339 películas (0.58 % del total) e ignorando 50274. En total se tienen 8314 películas seleccionadas y 120941 ignoradas, esto es, 14.4 películas ignoradas por cada seleccionada.

Estos datos están disponibles públicamente <sup>8</sup> y son usados en el siguiente capítulo para verificar el desempeño de los algoritmos modificados.

## 4.2 Métricas de evaluación

La evaluación de un sistema de recomendación puede ser realizada comparando la relevancia de los ítems de la lista de ítems recomendados y el orden en que se encuentran dentro de la lista. Dos métricas muy utilizadas por la comunidad de recuperación de la información son la precisión (*precision*) definida como:

$$P(A, B) = \frac{|A \cap B|}{|B|} \quad (\text{ecuación 13})$$

y el de recuperación (*recall*) definido como:

$$R(A, B) = \frac{|A \cap B|}{|A|} \quad (\text{ecuación 14})$$

Donde:

- *A* es la lista de ítems relevantes para el usuario
- *B* es la lista ordenada de ítems recomendados

---

<sup>8</sup> <http://pelisquemegustan.inductia.com/dataset/>

Estas dos métricas no toman en cuenta el orden en que los ítems fueron recomendados. Una métrica que sí lo tiene en cuenta es la Ganancia Acumulada Descontada (DCG, *Discounted Cumulative Gain*) que mide la ganancia de un ítem dentro de la lista recomendada de acuerdo a su posición. DCG penaliza aquellos ítems relevantes que aparecen lejos de las primeras posiciones de la lista, reduciendo en forma logarítmica su grado de relevancia de acuerdo a su posición.

La Ganancia Descontada Acumulada en una posición particular  $p$  de la lista de recomendaciones se define como:

$$DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(i + 1)} \quad (15)$$

Donde:

- $rel_i$  es un indicador cuyo valor es 1 si el ítem que se encuentra en la posición  $i$  es relevante, sino es 0

Dado que la cantidad de ítems en las listas de recomendación puede variar de usuario en usuario, se define la Ganancia Descontada Acumulada Normalizada (NDCG, *Normalized Discounted Cumulative Gain*) como:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad (16)$$

Donde:

- $IDCG_p$  ( $DCG_p$  ideal) es el máximo posible  $DCG_p$  hasta la posición  $p$  y se obtiene ordenando los ítems de una lista por relevancia

Es posible calcular estas tres métricas sólo para los primeros  $n$  ítems recomendados de una serie de recomendaciones, en cuyo caso hablamos de precisión media en  $n$  ( $AP@n$ ), recuperación media en  $n$  ( $AR@n$ ) y NDCG media en  $n$  ( $NCDG@n$ ).

Para evaluar el desempeño de los modelos fuera de línea tomamos las  $n$  primeras recomendaciones realizadas por cada uno de ellos, con  $n \in \{1, 5, 10\}$ . Para las pruebas

en línea con el sistema de recomendación desarrollado tomamos  $n = 8$  ya que esta es la cantidad de películas recomendadas por el sistema en cada interacción.

### 4.3 Protocolo de evaluación

Comparamos los algoritmos modificados con las versiones originales en distintas pruebas para verificar que el uso de la información del contexto de elección del usuario mejora el desempeño del filtrado colaborativo basado en memoria.

Se pueden realizar dos tipos de prueba de acuerdo a las características del conjunto de datos: pruebas fuera de línea y pruebas en línea [36].

Las pruebas fuera de línea se realizan usando un conjunto de datos recogido previamente que tiene las calificaciones de los usuarios. El protocolo de evaluación que usualmente se sigue en estos casos es el de validación cruzada: se divide el conjunto de calificaciones en dos partes y se usa una de ellas para entrenar los algoritmos de predicción de preferencias y la otra para evaluar las predicciones hechas. En la evaluación se predice, para cada usuario, una lista de preferencias a partir de los datos de entrenamiento y se la compara con las preferencias que están en los datos de prueba.

Realizamos este tipo de pruebas para comparar los algoritmos clásicos con los modificados por nosotros usando los conjuntos de datos movielens-1M, webscope y “pelis que me gustan”, haciendo una validación cruzada dividiendo cada conjunto de datos en cinco partes y calculando las métricas  $AP@n$ ,  $AR@n$  y  $NCDG@n$  para  $n \in \{1, 5, 10\}$ .

Las pruebas en línea simulan cada una de las interacciones que los usuarios tienen con el sistema de recomendación. El protocolo de evaluación que se sigue es, para cada interacción del usuario con el sistema siguiendo el orden temporal, se comparan las preferencias expresadas por el usuario en la interacción actual con las predicciones calculadas usando las interacciones anteriores. Evidentemente, para poder seguir este protocolo el conjunto de datos tiene que tener la fecha y hora de cada transacción. Se debe tener en cuenta que en este tipo de pruebas se entrena el modelo de predicción de preferencias por cada interacción, lo que redundaría en mucho tiempo de procesamiento.

Este tipo de pruebas las hicimos usando los conjuntos de datos movielens-100k y “pelis que me gustan” ya que cuentan con la fecha y hora de cada transacción. En el caso del conjunto de datos movielens.100k, cada registro del conjunto de datos tiene un ítem que fue seleccionado o ignorado por un usuario. Si el ítem actual fue seleccionado, se le pide al sistema una lista de ítems recomendados, se la compara con el ítem actual y se calculan las métricas de evaluación presentadas en la sección 4.2. Luego se utiliza el registro actual, seleccionado o ignorado, para entrenar el modelo de predicción de preferencias.

Por otra parte, en el conjunto de datos “pelis que me gustan”, cada registro tiene todos los ítems mostrados y seleccionados en cada interacción, con lo que podemos calcular las métricas de evaluación directamente.

En todos los experimentos utilizaremos la prueba  $t$  de Student de una cola para dos muestras apareadas, para verificar su validez estadística.

## **4.4 Resultados y discusión**

En esta sección presentamos los resultados de las pruebas realizadas. Se hicieron pruebas en línea y fuera de línea para los conjuntos de datos movielens-1M, webscope y “pelis que me gustan”. En cada caso se evaluaron dos variantes de los algoritmos de filtrado colaborativo: el basado en usuarios (CF-U) y el basado en ítems clásico (CF-I).

### **4.4.1 Pruebas fuera de línea**

A continuación mostramos y discutimos los resultados de nuestras pruebas fuera de línea para el conjunto de datos movielens-1M, webscope y “pelis que me gustan”.

Los algoritmos usados tienen un parámetro  $K$  que indica la cantidad de vecinos y se usa durante el cálculo de las preferencias. Este parámetro debe ser fijado antes de realizar las pruebas para cada uno de los conjuntos de datos y afecta a la performance de los algoritmos porque, cuanto más grande es, se toman en cuenta más vecinos para predecir las preferencias y consecuentemente tardan más tiempo. Entonces, tomaremos el menor valor de  $K$  que maximice las métricas vistas en la sección 4.2 para los algoritmos clásicos ( $\beta=1.0$ ).

#### 4.4.1.1 Conjunto de datos movielens-1M, algoritmo basado en usuarios

Para el algoritmo basado en usuarios clásico ( $\beta=1.0$ ), los valores de las métricas para los distintos valores de  $K$  pueden observarse en Tabla 4-1.

Tabla 4-1 - Métricas para distintos valores de  $K$ , conjunto de datos movielens-1M, CF-U,  $\beta=1.0$

<b>K</b>	<b>NDCG@1</b>	<b>NDCG@5</b>	<b>NDCG@10</b>	<b>AP@5</b>	<b>AP@10</b>	<b>AR@5</b>	<b>AR@10</b>
10	0,176	0,136	0,114	0,125	0,099	0,099	0,150
20	0,205	0,156	0,131	0,143	0,113	0,116	0,174
30	0,215	0,163	0,136	0,150	0,118	0,123	0,184
40	0,221	0,167	0,139	0,152	0,120	0,125	0,189
50	0,225	0,169	0,141	0,154	0,121	0,127	0,192
60	0,224	0,169	0,141	0,154	0,122	0,126	0,193
70	0,225	0,169	0,141	0,154	0,122	0,127	0,193
80	0,227	0,170	0,142	0,154	0,122	0,127	0,193
<b>90</b>	<b>0,228</b>	<b>0,170</b>	<b>0,142</b>	<b>0,155</b>	<b>0,121</b>	<b>0,127</b>	<b>0,193</b>
100	0,227	0,170	0,141	0,154	0,121	0,127	0,192
110	0,227	0,170	0,141	0,154	0,121	0,127	0,192
120	0,227	0,170	0,141	0,154	0,120	0,127	0,191

Usaremos  $K=90$  para las pruebas porque es el menor valor de  $K$  para el cual se maximizan las métricas. Luego calculamos las métricas de evaluación para el algoritmo basado en usuarios, para distintos valores del parámetro  $\beta$  y obtuvimos los resultados mostrados en la Tabla 4-2.

Tabla 4-2 – Valores de las métricas por valor de  $\beta$ , conjunto de datos movielens-1M, CF-U,  $K=90$

<b><math>\beta</math></b>	<b>NDCG@1</b>	<b>NDCG@5</b>	<b>NDCG@10</b>	<b>AP@5</b>	<b>AP@10</b>	<b>AR@5</b>	<b>AR@10</b>
1,0	0,2261	0,1690	0,1404	0,1533	0,1201	0,1262	0,1911
0,9	0,2352	0,1739	0,1440	0,1572	0,1227	0,1288	0,1946
0,8	0,2445	0,1791	0,1479	0,1614	0,1255	0,1321	0,1982
0,7	0,2547	0,1854	0,1526	0,1668	0,1290	0,1356	0,2027
0,6	0,2674	0,1926	0,1581	0,1727	0,1332	0,1393	0,2076
0,5	0,2792	0,2005	0,1642	0,1795	0,1379	0,1436	0,2132
0,4	0,2903	0,2088	0,1706	0,1871	0,1432	0,1484	0,2190
0,3	0,2984	0,2156	0,1765	0,1937	0,1485	0,1524	0,2246
0,2	0,3007	0,2191	0,1797	0,1973	0,1517	0,1550	0,2277
<b>0,1</b>	<b>0,3010</b>	<b>0,2194</b>	<b>0,1801</b>	<b>0,1977</b>	<b>0,1521</b>	<b>0,1553</b>	<b>0,2282</b>

Valores estadísticamente significativos ( $p < 0.001$ )

Podemos observar que hay diferencias importantes entre el algoritmo clásico ( $\beta=1,0$ ) y el modificado. Las máximas diferencias se presentan cuando  $\beta=0,1$  donde el

algoritmo modificado supera al clásico en 33.1%, 29.8% y 28.2% para NDCG, 28.9% y 26.6% para AP y 23.1% y 19.4% para AR.

En la Figura 4-1 podemos ver la evolución de todas las métricas para los diferentes valores de  $\beta$ . Se observa que la diferencia entre NDCG@1 y NDCG@5 es bastante mayor que la de NDCG@5 con NDCG@10. Además, mientras que NDCG@1 y NDCG@5 superan a NDCG@10 y AP@5 a AP@10, AR@10 supera a AR@5.

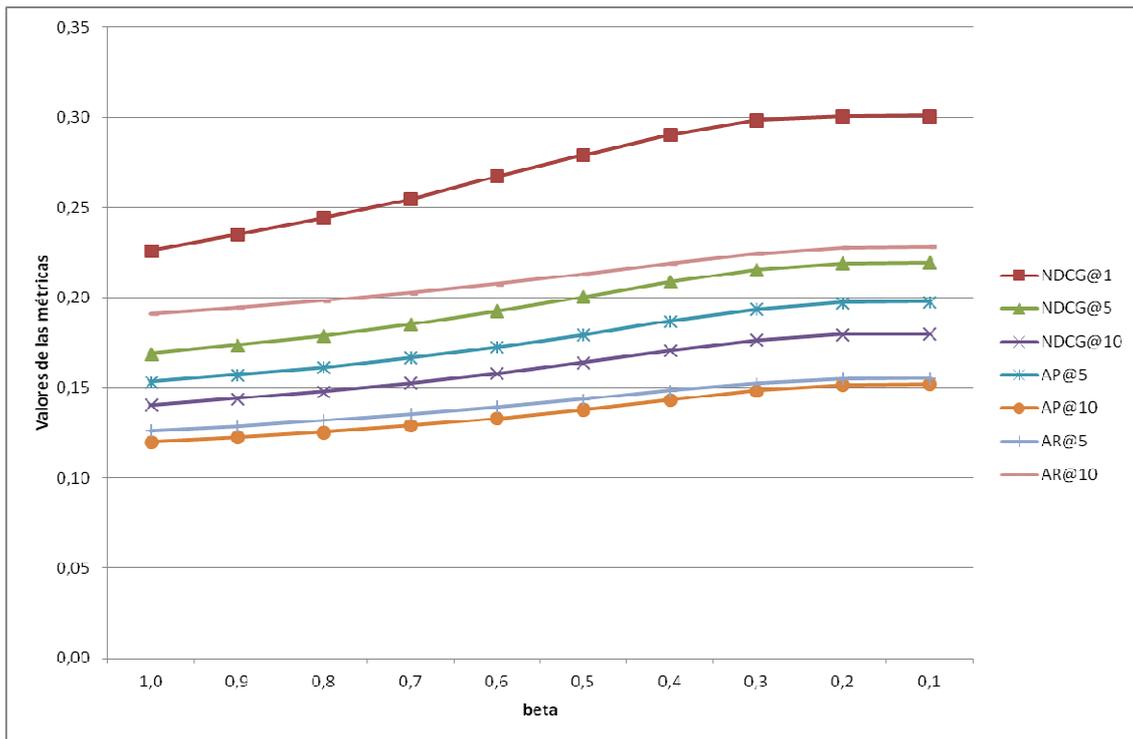


Figura 4-1 – Evolución de las métricas para conjunto de datos movielens-1M, CF-U

#### 4.4.1.2 Conjunto de datos movielens-1M, algoritmo basado en ítems

Para los algoritmos basados en ítems volvimos a calcular el valor de  $K$  y los resultados son los mostrados en la tabla Tabla 4-3. Al igual que en caso del algoritmo basado en usuarios, el menor valor de  $K$  que maximiza las métricas es  $K=90$ .

Tabla 4-3 - Métricas para distintos valores de  $K$ , conjunto de datos movielens-1M, modelo CF-I,  $\beta=1.0$

<b>K</b>	<b>NDCG@1</b>	<b>NDCG@5</b>	<b>NDCG@10</b>	<b>AP@5</b>	<b>AP@10</b>	<b>AR@5</b>	<b>AR@10</b>
10	0,1756	0,1356	0,1142	0,1246	0,0994	0,0988	0,1502

20	0,2048	0,1560	0,1310	0,1430	0,1134	0,1164	0,1744
30	0,2148	0,1634	0,1364	0,1496	0,1176	0,1226	0,1844
40	0,2212	0,1666	0,1392	0,1518	0,1200	0,1248	0,1886
50	0,2250	0,1686	0,1408	0,1536	0,1212	0,1266	0,1918
60	0,2240	0,1688	0,1414	0,1538	0,1218	0,1262	0,1930
70	0,2254	0,1692	0,1414	0,1544	0,1216	0,1266	0,1932
80	0,2274	0,1698	0,1416	0,1542	0,1218	0,1266	0,1934
<b>90</b>	<b>0,2282</b>	<b>0,1700</b>	<b>0,1416</b>	<b>0,1548</b>	<b>0,1214</b>	<b>0,1270</b>	<b>0,1926</b>
100	0,2274	0,1696	0,1414	0,1542	0,1212	0,1266	0,1920
110	0,2270	0,1700	0,1410	0,1540	0,1210	0,1270	0,1920
120	0,2270	0,1700	0,1410	0,1540	0,1200	0,1270	0,1910

La Tabla 4-4 muestra los resultados obtenidos para el algoritmo basado en entre ítems y puede observarse que son bastante menores que en el caso de similitud entre usuarios.

Tabla 4-4 - Valores de las métricas por valor de  $\beta$ , conjunto de datos movielens-1M, CF-I, K=90

$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
1,0	0,2192	0,1647	0,1377	0,1503	0,1187	0,1170	0,1822
0,9	**0,2199	0,1651	0,1379	*0,1506	*0,1188	0,1174	*0,1825
0,8	*0,2206	0,1654	0,1381	0,1509	0,1189	0,1180	*0,1828
0,7	0,2213	0,1658	0,1384	0,1512	0,1191	0,1184	0,1835
0,6	0,2225	0,1665	0,1387	0,1517	0,1193	0,1192	0,1840
0,5	0,2238	0,1671	0,1392	0,1522	0,1196	0,1200	0,1848
0,4	0,2256	0,1681	0,1398	0,1529	0,1199	0,1209	0,1856
0,3	0,2283	0,1696	0,1408	0,1541	0,1206	0,1225	0,1873
0,2	0,2324	0,1721	0,1425	0,1562	0,1216	0,1246	0,1894
<b>0,1</b>	<b>0,2406</b>	<b>0,1770</b>	<b>0,1458</b>	<b>0,1603</b>	<b>0,1239</b>	<b>0,1286</b>	<b>0,1936</b>

\*\* Estadísticamente no significativos ( $p < 0.05$ )

\* Estadísticamente no significativos ( $p < 0.01$ )

El resto son valores estadísticamente significativos ( $p < 0.001$ )

Las mayores diferencias con el algoritmo clásico ( $\beta=1,0$ ) se vuelven a dar para  $\beta=0,1$  pero son bastante menos importantes que en caso anterior: 9,7%, 7.5% y 5.9% para NDCG, 6,6% y 4,4% para AP y 9,9% y 6,2% para AR. Cabe aclarar que en algunos casos las diferencias no son estadísticamente significativas.

La Figura 4-2 muestra que, para los diferentes valores de  $\beta$ , la evolución de las métricas es similar al caso de los algoritmos basados en usuarios, aunque los valores son menores.

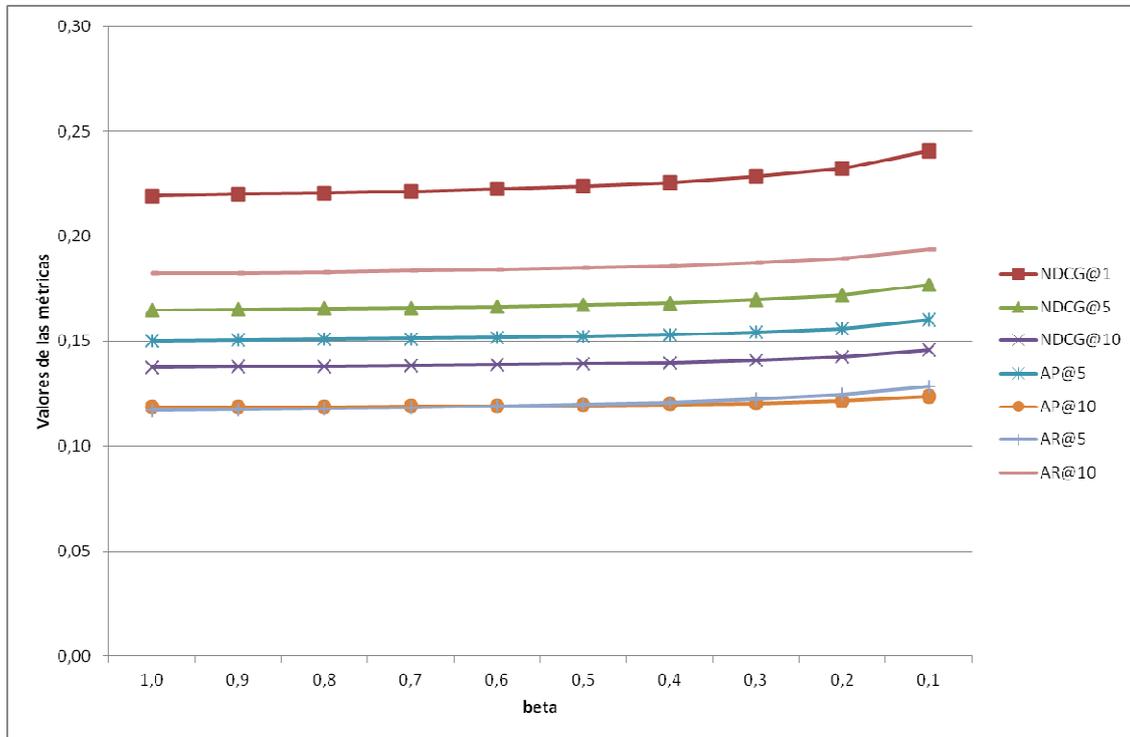


Figura 4-2 - Evolución de las métricas para conjunto de datos movielens-1M, CF-I

#### 4.4.1.3 Conjunto de datos webscope, algoritmo basado en usuarios

Calculamos nuevamente el mínimo valor del parámetro  $K$  que maximiza las métricas de evaluación para el conjunto de datos webscope.

Tabla 4-5 - Métricas para distintos valores de  $K$ , conjunto de datos webscope, modelo CF-U,  $\beta=1.0$

K	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
10	0,248	0,197	0,165	0,183	0,144	0,189	0,280
20	0,277	0,217	0,181	0,201	0,157	0,209	0,309
30	0,288	0,225	0,187	0,207	0,162	0,217	0,321
40	0,295	0,229	0,190	0,211	0,164	0,222	0,327
50	0,301	0,231	0,192	0,212	0,165	0,225	0,330
60	0,303	0,233	0,193	0,214	0,166	0,226	0,333
70	0,304	0,233	0,193	0,214	0,166	0,227	0,334
80	0,307	0,234	0,193	0,214	0,166	0,228	0,335

90	0,306	0,234	0,194	0,215	0,166	0,229	0,336
<b>100</b>	<b>0,308</b>	<b>0,234</b>	<b>0,194</b>	<b>0,215</b>	<b>0,166</b>	<b>0,229</b>	<b>0,337</b>
110	0,307	0,234	0,194	0,214	0,166	0,229	0,337
120	0,308	0,234	0,194	0,214	0,166	0,229	0,337
130	0,308	0,234	0,194	0,214	0,166	0,229	0,337
140	0,308	0,234	0,193	0,214	0,165	0,229	0,337
150	0,308	0,234	0,193	0,213	0,165	0,229	0,336

Valores estadísticamente significativos ( $p < 0.001$ )

En la Tabla 4-5 podemos observar que debemos seleccionar  $K=100$ . Los valores de las métricas obtenidas para los diferentes valores de  $\beta$  usando  $K=100$  pueden observarse en la Tabla 4-6.

Tabla 4-6 - Valores de las métricas por valor de  $\beta$ , conjunto de datos webscope, CF-U,  $K=100$

$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
0,1	0,3062	0,2344	0,1937	0,2147	0,1662	0,2296	0,3355
0,9	0,3105	0,2368	0,1954	0,2166	0,1672	0,2314	0,3376
0,8	0,3152	0,2395	0,1973	0,2188	0,1685	0,2334	0,3400
0,7	0,3210	0,2429	0,1996	0,2215	0,1701	0,2360	0,3429
0,6	0,3287	0,2469	0,2025	0,2245	0,1720	0,2386	0,3461
0,5	0,3358	0,2516	0,2058	0,2285	0,1744	0,2420	0,3499
0,4	0,3437	0,2570	0,2098	0,2331	0,1773	0,2455	0,3543
0,3	0,3521	0,2630	0,2143	0,2384	0,1808	0,2498	0,3592
0,2	0,3583	0,2683	0,2187	0,2434	0,1846	0,2542	0,3645
<b>0,1</b>	<b>0,3599</b>	<b>0,2704</b>	<b>0,2208</b>	<b>0,2457</b>	<b>0,1868</b>	<b>0,2564</b>	<b>0,3680</b>

Valores estadísticamente significativos ( $p < 0.001$ )

En esta prueba todos los valores son estadísticamente significativos y como en el caso del conjunto de datos movielens-1M, las mayores diferencias con el algoritmo clásico se alcanzan cuando  $\beta=0,1$ : 17,5%, 15,4% y 14,0% para NDCG, 14,4% y 12,4% para AP y 11,7% y 9,7% para AR.

La Figura 4-3 muestra que la evolución de las métricas para los distintos valores de  $\beta$  tienen un comportamiento similar al que tuvieron con el conjunto de datos movielens-1M. La única diferencia destacable es que los valores de AR@10 están por encima de NCDG@1.

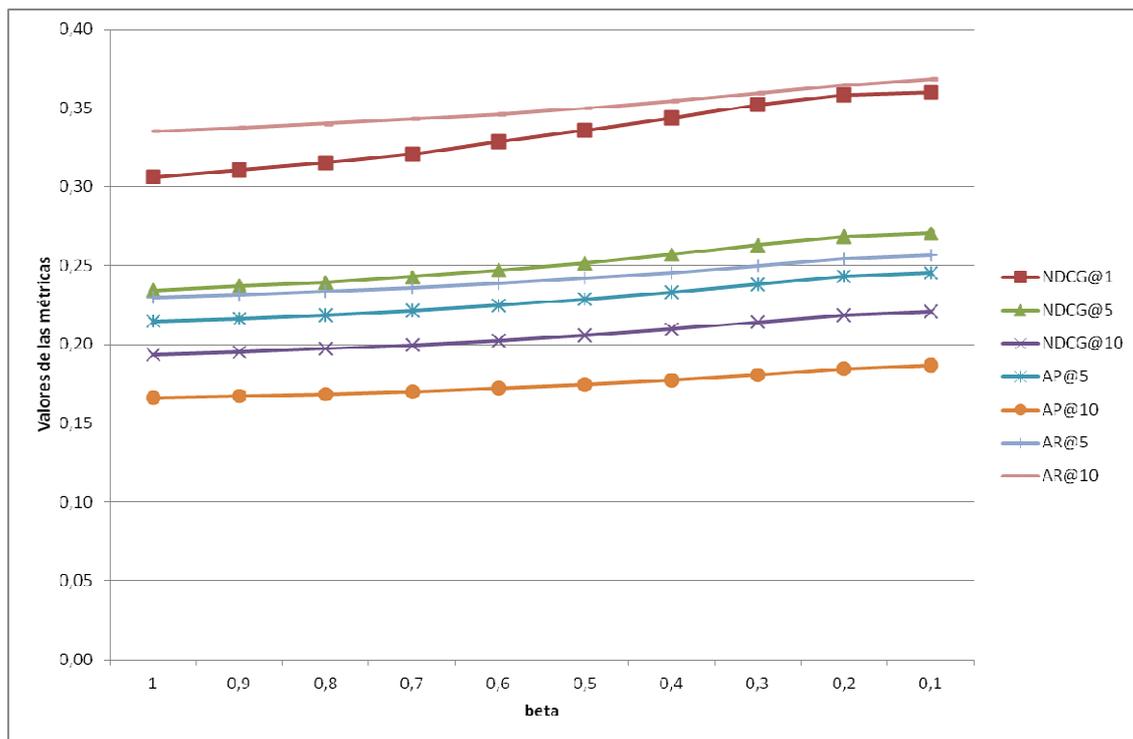


Figura 4-3- Evolución de las métricas para conjunto de datos webscope, CF-U

#### 4.4.1.4 Conjunto de datos webscope, algoritmo basado en ítems

Los resultados del cálculo del parámetro de la cantidad de vecinos  $K$  muestran (Tabla 4-7) que no existe un mínimo de  $K$  que maximice los valores de las métricas. Entonces, no usamos un valor de  $K$  durante la ejecución del algoritmo, lo que implicó que el conjunto de vecinos fueron todos los ítems de conjunto de datos.

Tabla 4-7 - Métricas para distintos valores de  $K$ , conjunto de datos webscope, modelo CF-I,  $\beta=1.0$

<b>K</b>	<b>NDCG@1</b>	<b>NDCG@5</b>	<b>NDCG@10</b>	<b>PREC@5</b>	<b>AP@10</b>	<b>AR@5</b>	<b>AR@10</b>
10	0,1088	0,0804	0,0698	0,0732	0,0618	0,0888	0,1396
100	0,2160	0,1650	0,1356	0,1504	0,1160	0,1782	0,2640
200	0,2368	0,1830	0,1504	0,1676	0,1288	0,1894	0,2812
300	0,2454	0,1902	0,1574	0,1748	0,1354	0,1944	0,2898
400	0,2518	0,1952	0,1614	0,1798	0,1390	0,1974	0,2932
500	0,2574	0,1986	0,1648	0,1822	0,1418	0,1990	0,2966
600	0,2616	0,2010	0,1668	0,1844	0,1434	0,2000	0,2986
700	0,2622	0,2032	0,1680	0,1864	0,1448	0,2016	0,3000
1000	0,2680	0,2066	0,1714	0,1894	0,1476	0,2036	0,3032
<b>max</b>	<b>0,2828</b>	<b>0,2184</b>	<b>0,1820</b>	<b>0,2004</b>	<b>0,1574</b>	<b>0,2088</b>	<b>0,3118</b>

Entonces, tomando  $K=\max$  calculamos los valores de las métricas de evaluación para los distintos valores de  $\beta$  y los resultados se muestran en la Tabla 4-8.

Tabla 4-8 - Valores de las métricas por valor de  $\beta$ , conjunto de datos webscope, CF-I,  $K=\max$

$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
1,0	0,2828	0,2183	0,1821	0,2006	0,1574	0,2088	0,3120
0,9	0,2837	0,2187	0,1823	0,2008	0,1575	0,2093	0,3125
0,8	0,2845	0,2191	0,1826	0,2011	0,1576	0,2098	0,3130
0,7	0,2855	0,2195	0,1829	0,2014	0,1578	0,2104	0,3136
0,6	0,2870	0,2201	0,1832	0,2017	0,1580	0,2109	0,3143
0,5	0,2847	0,2191	0,1826	0,2011	0,1577	0,2098	0,3131
0,4	0,2847	0,2191	0,1826	0,2011	0,1577	0,2098	0,3131
<b>0,3</b>	<b>0,2926</b>	<b>0,2230</b>	<b>0,1851</b>	<b>0,2039</b>	<b>0,1590</b>	<b>0,2145</b>	<b>0,3177</b>
0,2	0,2853	0,2194	0,1828	0,2013	0,1577	0,2102	0,3134
0,1	0,2869	0,2202	0,1833	0,2018	0,1580	0,2111	0,3143

Valores estadísticamente significativos ( $p < 0.01$ )

Se observa que todos los valores son estadísticamente significativos y que las diferencias con el algoritmo clásico ( $\beta=1.0$ ) no son muy importantes y se maximizan cuando  $\beta=0,3$ : 3,5% , 2,2% y 1,7% para NDCG, 1,7% y 1,0% para AP y 2,8% y 1,8% para AR.

La Figura 4-4 muestra la evolución de las métricas de evaluación para distintos valores de  $\beta$ . A diferencia de las pruebas anteriores, las métricas no crecen a medida que  $\beta$  decrece, mostrando un pico cuando  $\beta=0,3$ . De la misma forma que en el caso del algoritmo basado en usuarios, AR@10 está por encima de NCDG@1.

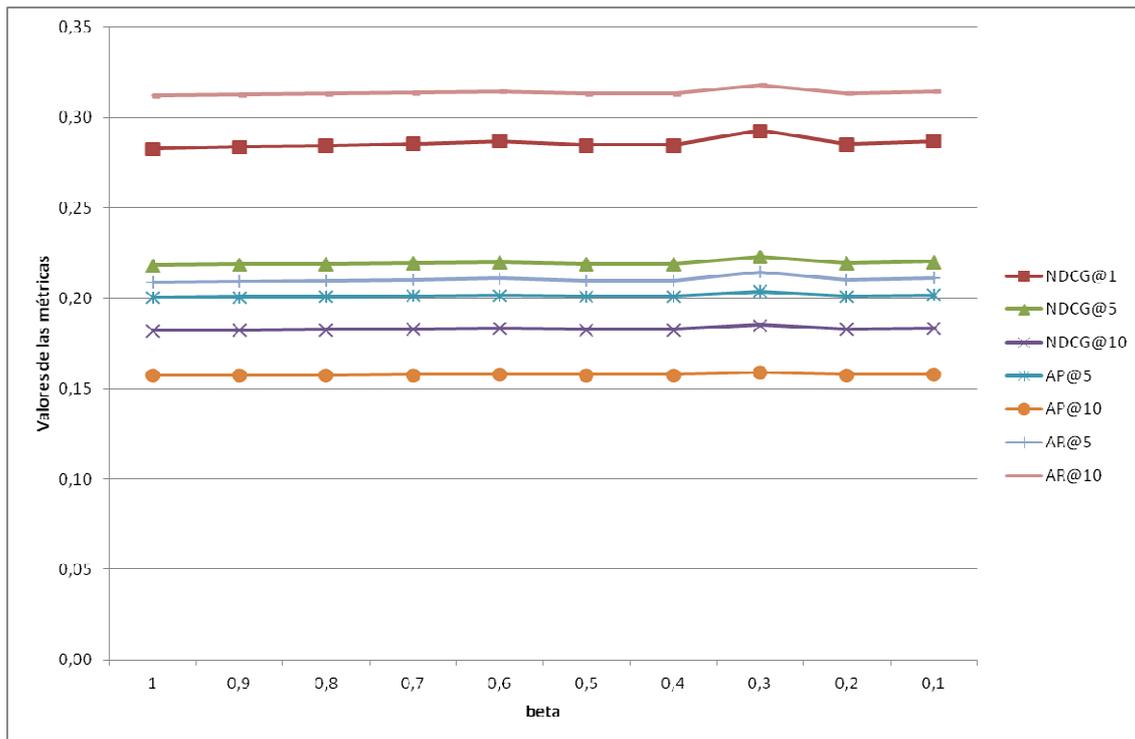


Figura 4-4 - Evolución de las métricas para conjunto de datos webscope, CF-I

#### 4.4.1.5 Conjunto de datos “pelis que me gustan”, algoritmo basado en usuarios

Para el conjunto de datos “pelis que me gustan”, el mínimo valor de  $K$  que maximiza las métricas de evaluación es  $K=20$  como puede observarse en la Tabla 4-9.

Tabla 4-9 - Métricas para distintos valores de  $K$ , conjunto de datos “pelis que me gustan”, modelo CF-U,  $\beta=1.0$

K	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
10	0,2218	0,1944	0,1792	0,1884	0,1690	0,1510	0,2622
<b>20</b>	<b>0,2356</b>	<b>0,2022</b>	<b>0,1880</b>	<b>0,1938</b>	<b>0,1776</b>	<b>0,1510</b>	<b>0,2772</b>
30	0,2246	0,2014	0,1854	0,1954	0,1752	0,1518	0,2680
40	0,2208	0,1974	0,1832	0,1916	0,1736	0,1514	0,2684
50	0,2134	0,1974	0,1824	0,1930	0,1732	0,1526	0,2674
60	0,2134	0,1952	0,1802	0,1908	0,1712	0,1480	0,2636
70	0,2124	0,1930	0,1792	0,1882	0,1708	0,1480	0,2646
80	0,2134	0,1906	0,1790	0,1856	0,1708	0,1442	0,2676
90	0,2124	0,1902	0,1778	0,1852	0,1698	0,1446	0,2656
100	0,2090	0,1892	0,1762	0,1850	0,1684	0,1450	0,2626
110	0,2144	0,1880	0,1756	0,1826	0,1674	0,1420	0,2600
120	0,2116	0,1876	0,1742	0,1824	0,1654	0,1426	0,2572
130	0,2026	0,1858	0,1736	0,1812	0,1656	0,1410	0,2576
140	0,2016	0,1862	0,1728	0,1820	0,1654	0,1412	0,2564
150	0,2024	0,1860	0,1726	0,1812	0,1646	0,1406	0,2564

Entonces, tomando  $K=20$  calculamos los valores de las métricas de evaluación para los distintos valores de  $\beta$  y los resultados se muestran en la Tabla 4-10.

Tabla 4-10 - Valores de las métricas por valor de  $\beta$ , conjunto de datos "pelis que me gustan", CF-U,  $K=20$

$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
1,0	0,2348	0,2033	0,1871	0,1960	0,1763	0,1536	0,2736
0,9	0,2702	0,2323	0,2084	0,2233	0,1938	0,1735	0,3004
0,8	0,3159	0,2655	0,2370	0,2515	0,2175	0,1975	0,3361
0,7	0,3839	0,3108	0,2719	0,2910	0,2447	0,2303	0,3783
0,6	0,4406	0,3595	0,3082	0,3376	0,2747	0,2705	0,4280
0,5	0,5135	0,4155	0,3493	0,3881	0,3068	0,3213	0,4803
0,4	0,5843	0,4723	0,3893	0,4387	0,3362	0,3778	0,5331
0,3	0,6571	0,5144	0,4152	0,4730	0,3513	0,4131	0,5583
<b>0,2</b>	<b>0,6748</b>	0,5228	<b>0,4191</b>	0,4782	<b>0,3514</b>	0,4174	<b>0,5549</b>
<b>0,1</b>	<b>0,6748</b>	<b>0,5230</b>	0,4190	<b>0,4784</b>	0,3513	<b>0,4176</b>	<b>0,5549</b>

Valores estadísticamente significativos ( $p < 0.001$ )

Podemos observar que los valores más altos de las métricas se tienen cuando  $\beta=0,1$  o  $\beta=0,2$  según la métrica, aunque la diferencia es menor al 0.02% muy poca.

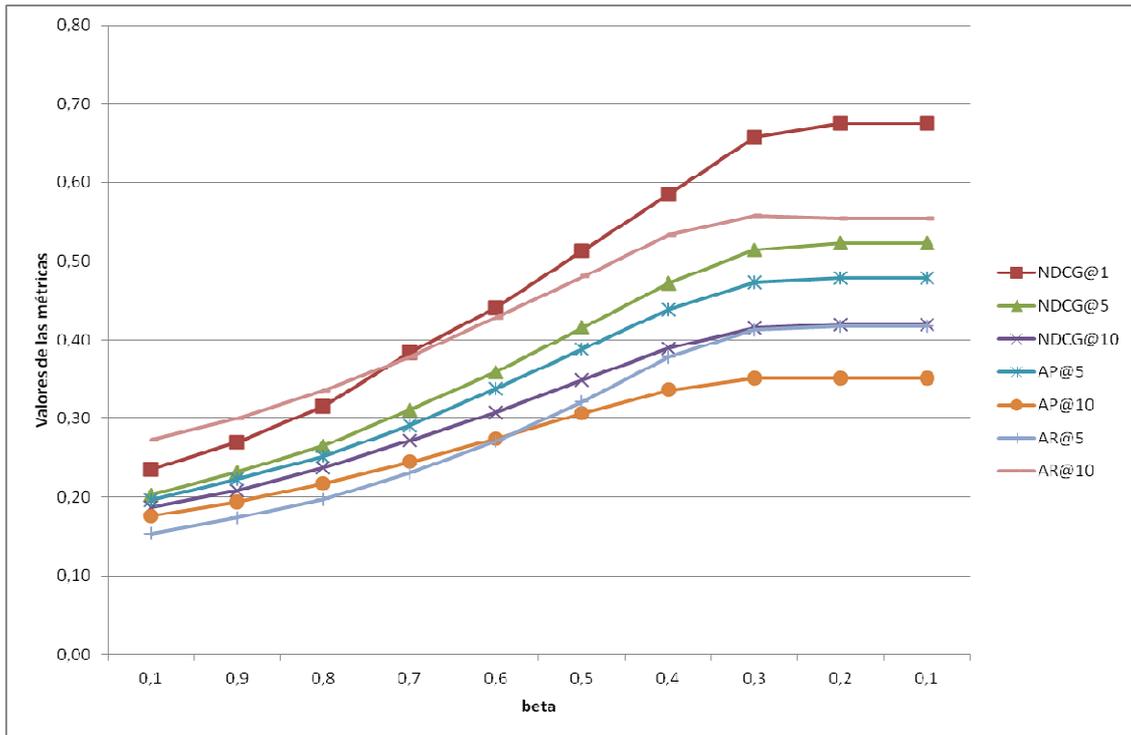


Figura 4-5 - Evolución de las métricas para conjunto de datos “pelis que me gustan”, CF-U

Las diferencias de los valores de  $\beta=0,1$  con los obtenidos cuando de  $\beta=1,0$  son muy grandes: 187,4%, 157,3% y 124,0% para NCDG, 144,1% y 99,3% para AR y 171,9% y 102,8% para AP.

La evolución de las métricas que se observa en la Figura 4-5 muestra que los valores aumentan más rápidamente que en las pruebas anteriores y que el ordenamiento de las métricas es el usual.

#### 4.4.1.6 Conjunto de datos “pelis que me gustan”, algoritmo basado en ítems

Como en las pruebas anteriores, en el caso del algoritmo basado en ítems volvemos a obtener que no existen un valor del parámetro  $K$  que maximice las métricas de evaluación (Tabla 4-11).

Tabla 4-11 - Métricas para distintos valores de  $K$ , conjunto de datos “pelis que me gustan”, modelo CF-I,  $\beta=1,0$

<b>K</b>	<b>NDCG@1</b>	<b>NDCG@5</b>	<b>NDCG@10</b>	<b>PREC@5</b>	<b>AP@10</b>	<b>AR@5</b>	<b>AR@10</b>
10	0,1378	0,1188	0,1138	0,1136	0,1094	0,0996	0,1944
100	0,1640	0,1488	0,1352	0,1436	0,1260	0,1236	0,2192
200	0,1642	0,1542	0,1426	0,151	0,1356	0,1282	0,2334

300	0,169	0,1538	0,1446	0,1486	0,1376	0,1284	0,2324
400	0,1754	0,1612	0,1504	0,1558	0,1428	0,1346	0,2392
500	0,1826	0,1672	0,1534	0,1622	0,1448	0,1376	0,2450
600	0,1878	0,1692	0,1538	0,1630	0,1438	0,1358	0,2408
700	0,1912	0,1694	0,1562	0,1622	0,1466	0,1362	0,2434
1000	0,1960	0,1736	0,1594	0,1672	0,1500	0,1388	0,2486
<b>max</b>	<b>0,2026</b>	<b>0,1844</b>	<b>0,1698</b>	<b>0,1798</b>	<b>0,1612</b>	<b>0,1446</b>	<b>0,2562</b>

Entonces, tomamos  $K=\text{máx}$  para calcular el valor de las métricas para los distintos valores de  $\beta$  que se muestra en la Tabla 4-12.

Varios valores de las métricas en dicha tabla no son estadísticamente significativos pero cuando las métricas se maximizan en  $\beta=0.1$  sí lo son.

Las diferencias entre los valores de  $\beta=1.0$  y  $\beta=0.1$  son altas: 84,0%, 58,6% y 46,7% para NDCG, 50,7% y 35,8% para AP y 58,3% y 38,7% para AR.

Tabla 4-12 - Valores de las métricas por valor de  $\beta$ , conjunto de datos "pelis que me gustan", CF-I,  $K=\text{máx}$

$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
1,0	0,2022	0,1844	0,1698	0,1795	0,1610	0,1445	0,2558
0,9	**0,2059	**0,1862	0,1715	**0,1809	**0,1624	**0,1464	**0,2582
0,8	**0,2090	0,1886	0,1729	**0,1827	**0,1631	**0,1485	**0,2590
0,7	**0,2109	0,1912	0,1745	*0,1855	*0,1645	**0,1515	**0,2610
0,6	**0,2165	0,1942	0,1771	0,1881	0,1666	*0,1533	*0,2644
0,5	**0,2203	0,1987	0,1801	0,1928	0,1692	0,1574	**0,2694
0,4	*0,2371	0,2061	0,1854	0,1980	0,1722	0,1630	*0,2753
0,3	0,2567	0,2174	0,1935	0,2062	0,1773	0,1695	0,2830
0,2	0,2950	0,2419	0,2106	0,2276	0,1897	0,1897	0,3048
<b>0,1</b>	<b>0,3722</b>	<b>0,2925</b>	<b>0,2490</b>	<b>0,2706</b>	<b>0,2186</b>	<b>0,2288</b>	<b>0,3550</b>

\*\* Estadísticamente no significativos ( $p < 0.05$ )

\* Estadísticamente no significativos ( $p < 0.01$ )

El resto son valores estadísticamente significativos ( $p < 0.001$ )

Puede observarse en la Figura 4-6 que las métricas casi no crecen para los mayores valores de  $\beta$  y recién después de  $\beta=0.3$  crecen rápidamente.

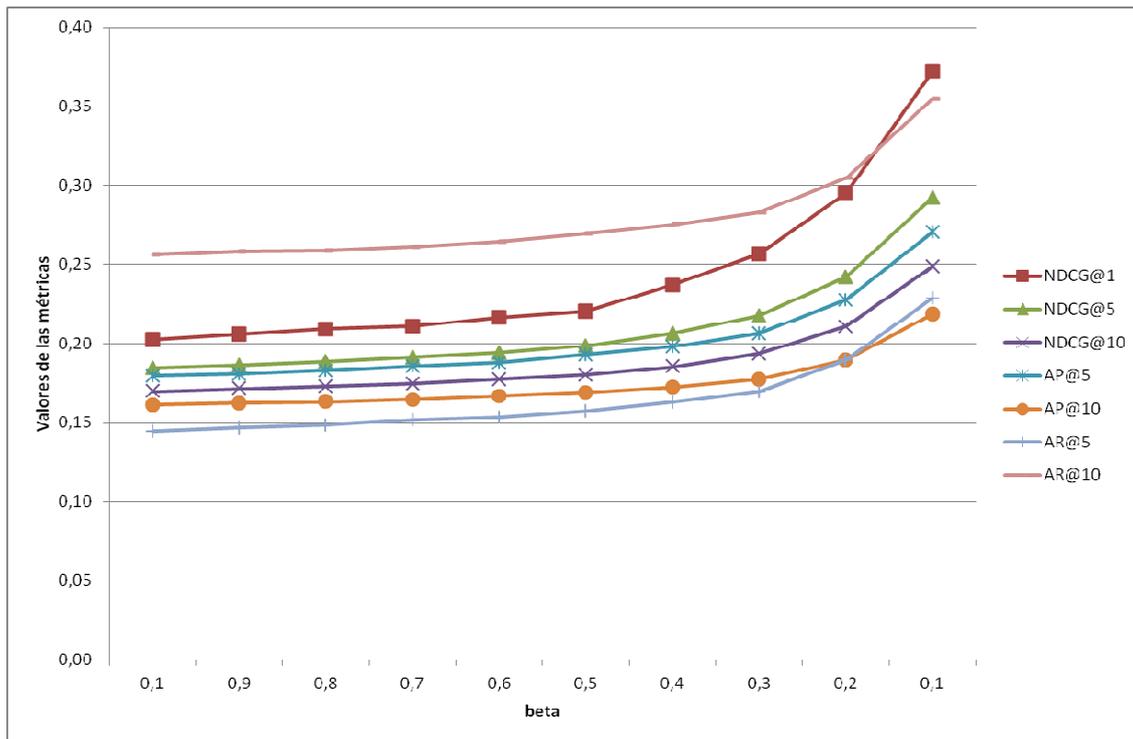


Figura 4-6 - Evolución de las métricas para conjunto de datos “pelis que me gustan”, CF-I

Las métricas tienen el orden usual visto en las otras pruebas, aunque esta vez el valor de AR@10 supera al de NCDG@1 solamente cuando  $\beta=0.1$ .

#### 4.4.1.7 Resumen de las pruebas fuera de línea

En las pruebas fuera de línea realizamos una validación cruzada en cinco partes de forma de poder comparar los algoritmos basados en usuarios e ítems clásicos con nuestra propuesta de modificación para los conjuntos de datos movielens-1M, webscope y “pelis que me gustan”.

La Tabla 4-13 muestra un resumen general de las diferencias de las pruebas fuera de línea de los algoritmos basados en usuarios, como un porcentaje de la diferencia de las métricas obtenidas cuando  $\beta=1.0$  contra el  $\beta$  que dio el mayor valor.

Tabla 4-13 - Diferencias en porcentaje entre métricas para  $\beta=1.0$  y el mejor  $\beta$ , según el conjunto de datos, CF-U

Conjunto de datos	$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
movielens-1M	0,1	33,1%	29,8%	28,2%	28,9%	26,6%	23,1%	19,4%

webscope	0,1	17,5%	15,4%	14,0%	14,4%	12,4%	11,7%	9,7%
pelis que me gustan	0,1	187,4%	157,3%	124,0%	144,1%	99,3%	171,9%	102,8%

Podemos observar que mientras que las diferencias para los conjuntos de datos movielens-1M y webscope están por debajo del 35%, en el caso de “pelis que me gustan” las diferencias son mucho más importantes, superando el 100% en casi todos los casos.

La Tabla 4-14 resume las diferencias para los algoritmos basados en usuarios, por conjunto de datos y métrica.

Tabla 4-14 - Diferencias en porcentaje entre métricas para  $\beta=1.0$  y el mejor  $\beta$ , según el conjunto de datos, CF-I

Conjunto de datos	$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10	AR@5	AR@10
movielens-1M	0,1	9,7%	7,5%	5,9%	6,7%	4,4%	9,9%	6,2%
webscope	0,3	3,5%	2,2%	1,7%	1,7%	1,0%	2,8%	1,8%
pelis que me gustan	0,1	84,0%	58,6%	46,7%	50,7%	35,8%	58,3%	38,7%

Las grandes diferencias que se observan entre los conjunto de datos se deben a la diferentes proporciones de películas seleccionadas contra ignoradas que presentan. Cuanto más grande sea la proporción de ignoradas con respecto a las seleccionadas, más información se va a tener en el contexto de elección del usuario para ser usado en la predicción de las preferencias. El conjunto de datos webscope tiene 1.62 películas ignoradas por cada seleccionada, mientras que movielens-1M tiene 3.41 y “pelis que me gustan” 14.5.

#### 4.4.2 Pruebas en línea

Como dijimos en la sección 4.3, las pruebas en línea simulan cada una de las interacciones que tiene cada usuario con el sistema, siguiendo la línea de tiempo presente en el conjunto de datos. Realizamos las pruebas en línea usando los conjuntos de datos movielens-100k y “pelis que me gustan” porque tienen la fecha y hora de cada

interacción. Utilizamos movieles-100k en vez de movielens-1M porque la cantidad de registros de ese conjunto de datos no permite procesarlo en un tiempo razonable.

A diferencia de las pruebas anteriores, en este caso no se conoce la totalidad de los ítems relevantes para el usuario. Esto hace que no tenga sentido calcular la métrica AR, por lo que mediremos el desempeño de los algoritmos usando NCDG y AP.

Comparamos el algoritmo de filtrado colaborativo basado en usuarios clásico con el modificado ( $\beta=0.1$ ), usando  $K=90$ . El algoritmo basado en usuarios es el que implementamos en el sistema de recomendación desarrollado y por eso lo usaremos en la comparación. Tomamos ese valor de  $K$  y de  $\beta$  porque fueron los valores que mostraron mejor desempeños en las pruebas fuera de línea presentadas en el apartado anterior.

#### 4.4.2.1 Conjunto de datos movielens-100k

Los resultados de la prueba en línea usando el conjunto de datos movielens-100k se muestran en la Tabla 4-15.

*Tabla 4-15 - Comparación de las medias de las métricas para los distintos valores de  $\beta$ , movielens-100k, CF-U*

$\beta$	NDCG@1	NDCG@5	NDCG@10	AP@5	AP@10
1,0	0,0218	0,0172	0,0149	0,0160	0,0132
0,1	0,0338	0,0242	0,0202	0,0218	0,0172

Valores estadísticamente significativos ( $p < 0.001$ )

Puede observarse que el algoritmo modificado supera los valores del algoritmo clásico para todas las métricas.

Si observamos en detalle la evolución de las métricas NCDG por la cantidad de usuarios que entran en el sistema (Figura 4-7) podemos verificar que después de un período inicial las métricas comienzan a estabilizarse.

Este período inicial se debe a que el sistema tiene poca información de entrenamiento por la poca cantidad de usuarios y esto provoca que sus recomendaciones sean de mala calidad para todos los algoritmos.

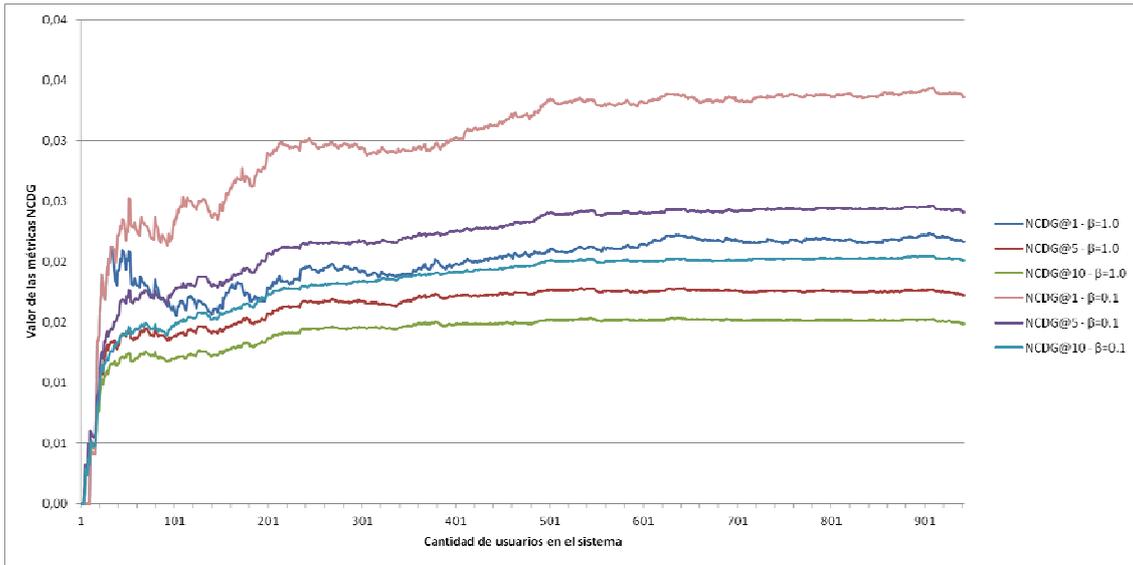


Figura 4-7 - NDCG@1, NDCG@5 y NDCG@10 por cantidad de usuarios, para  $\beta=1.0$  y  $\beta=0.1$ , movielens-100k

En todos los casos las métricas del algoritmo modificado están por encima de las del algoritmo clásico. Para las métricas AP sucede algo muy similar (Figura 4-8) al caso anterior.

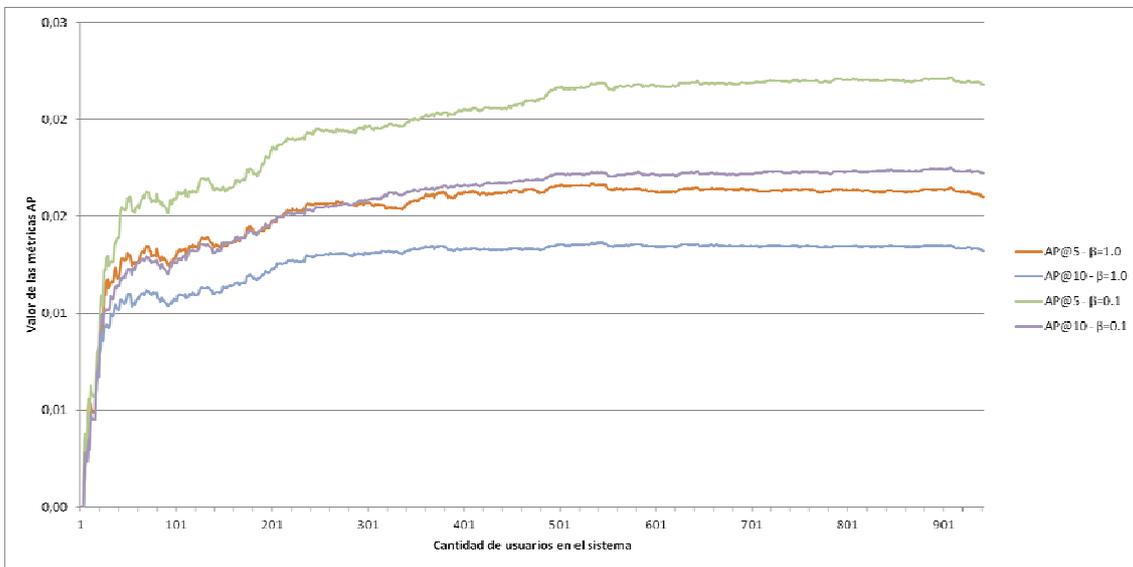


Figura 4-8 - AP@5 y AP@10 por cantidad de usuarios, para  $\beta=1.0$  y  $\beta=0.1$ , movielens-100k

La evolución de las métricas NCDG vista por la cantidad de interacciones del sistema (Figura 4-9) presenta un comportamiento similar al caso de cantidad de usuarios: un período inicial inestable y estabilización de todas las métricas.

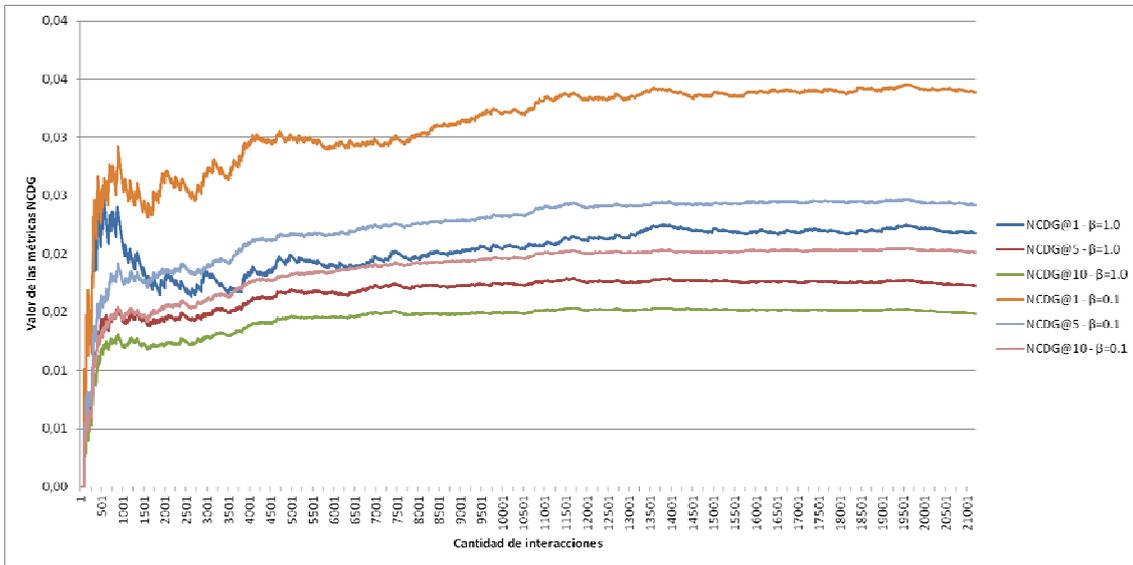


Figura 4-9 - NDCG@1, NDCG@5 y NDCG@10 por cantidad de interacciones, para  $\beta=1.0$  y  $\beta=0.1$ , movielens-100k

El mismo comportamiento que NCDG puede observarse en la Figura 4-10 con las métricas AP.

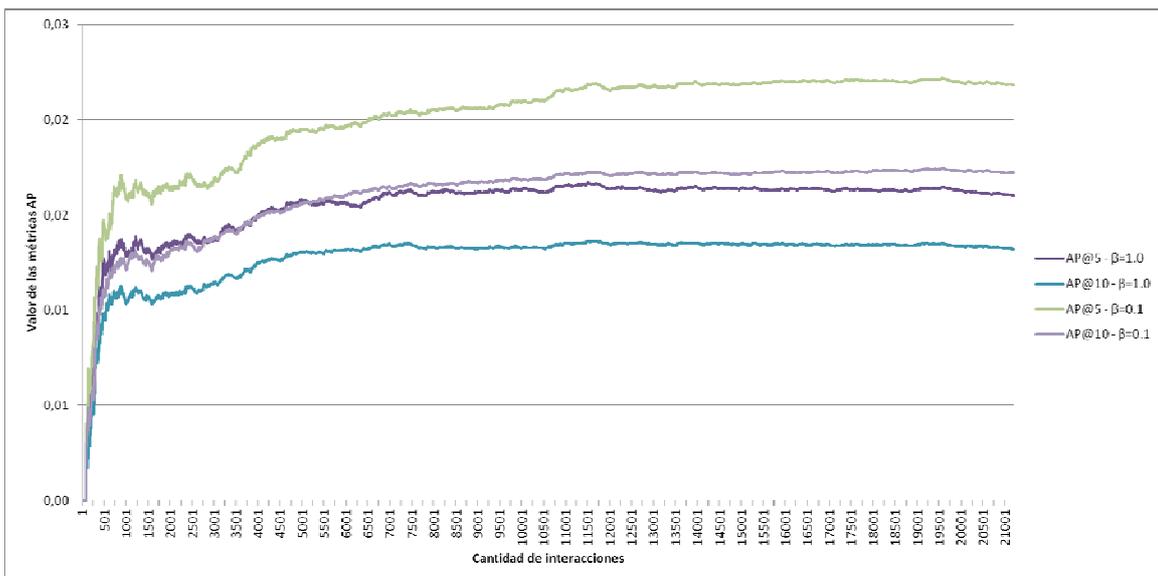


Figura 4-10 – AP@5 y AP@10 por cantidad de interacciones, para  $\beta=1.0$  y  $\beta=0.1$ , movielens-100k

#### 4.4.2.2 Conjunto de datos “pelis que me gustan”

Para la segunda prueba en línea, usamos el conjunto de datos obtenido del sistema de recomendación que desarrollamos. En este caso usaremos las métricas NDCG@8 y AP@8 ya que el sistema le recomienda al usuario 8 ítems en cada interacción.

Tabla 4-16 - Comparación de las medias de las métricas para los distintos valores de  $\beta$ , “pelis que me gustan”, CF-U

$\beta$	NDCG@8	AP@8
1,0	0,0386	0,0411
0,1	0,0976	0,0957

Valores estadísticamente significativos ( $p < 0.001$ )

La Figura 4-11 muestra la evolución de la métrica NDCG@8 a medida que el sistema tiene mayor cantidad de usuarios.

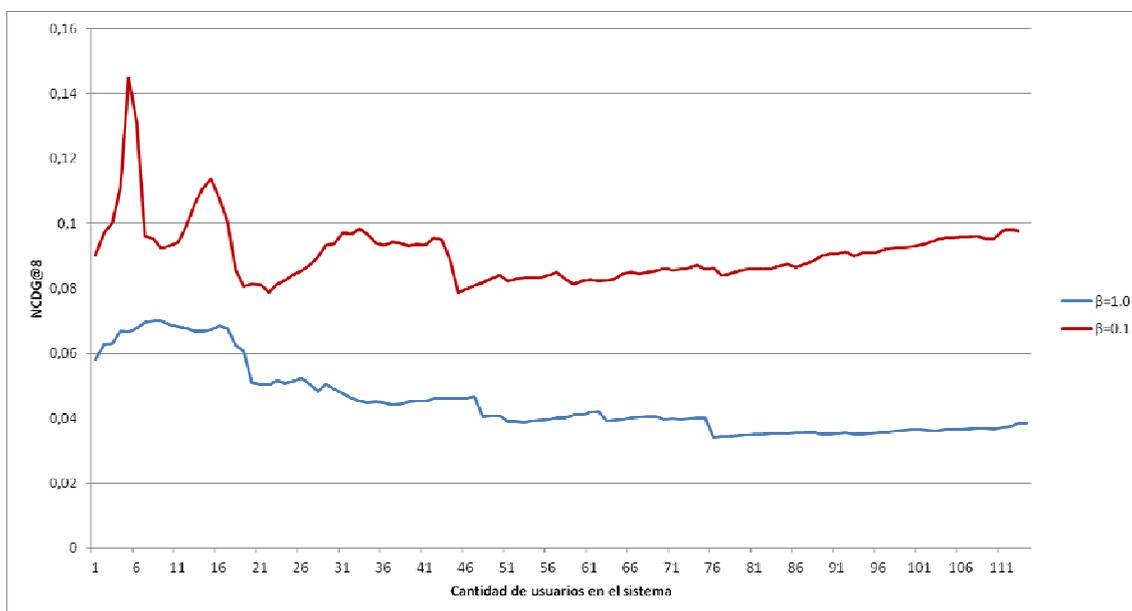


Figura 4-11 - NDCG@8 por cantidad de usuarios en el sistema

Podemos observar que el modelo modificado ( $\beta=0,1$ ) domina al modelo clásico ( $\beta=1,0$ ) en toda la serie y que, luego de un período inicial de inestabilidad, los valores comienzan a crecer y se acrecienta la diferencia entre los dos modelos.

El período inicial de inestabilidad se debe a que el sistema se encuentra en su fase de arranque en frío, esto es, no tiene suficientes datos para hacer buenas predicciones. La estabilidad comienza cuando la cantidad de usuarios en el sistema se acerca al valor de  $K=90$ , la cantidad de vecinos a usar en el cálculo de las predicciones de preferencias.

La Figura 4-12 muestra que el comportamiento de AP@8 es similar al de NCDG@8.

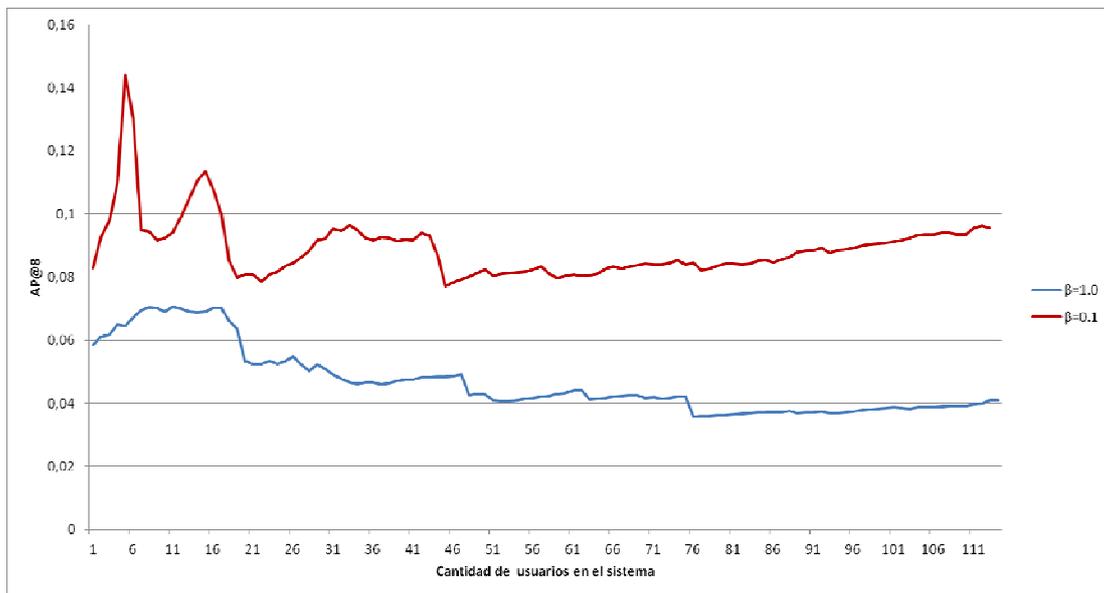


Figura 4-12 - AP@8 por cantidad de usuarios en el sistema

Si analizamos las mismas métricas por la cantidad de interacciones entre el usuario y el sistema de recomendación podemos ver que en este caso también el modelo modificado mejora la performance del modelo clásico.

La Figura 4-13 muestra que la evolución de la métrica NDCG a medida que se tienen registradas más interacciones usuario-sistema es similar a la evolución por cantidad de usuarios: el algoritmo modificado domina en toda la serie al clásico y luego

de un período inicial de inestabilidad, las métricas comienzan a crecer y la diferencia se hace cada vez más grande.

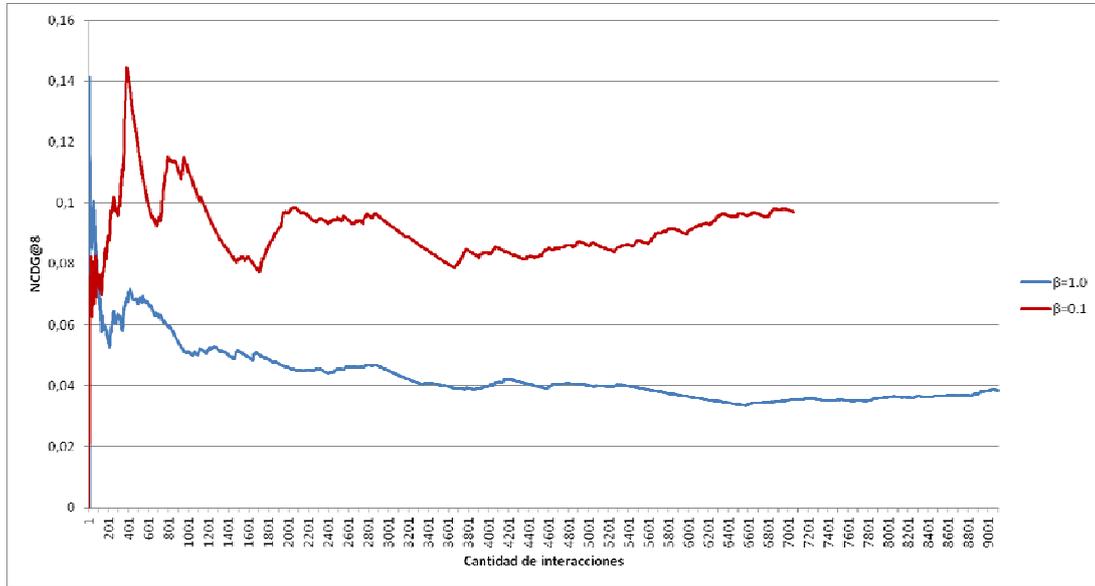


Figura 4-13 - NDCG@8 por cantidad de interacciones

La métrica AP@8 muestra un comportamiento muy similar a NDCG@8 (Figura 4-14).

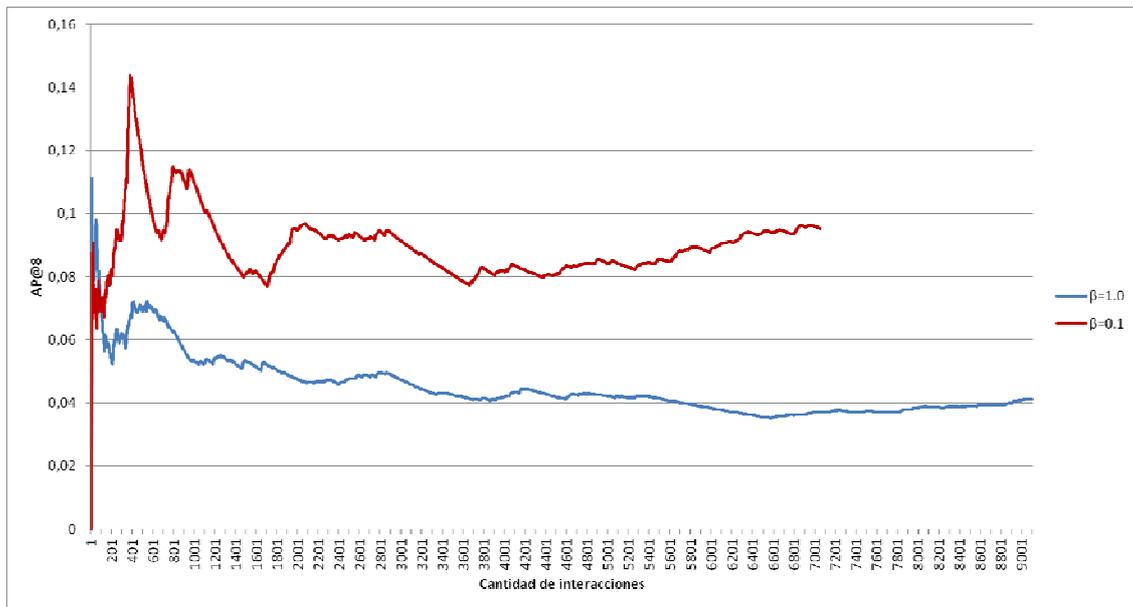


Figura 4-14 - AP@8 por cantidad de interacciones

#### **4.4.2.3 Resumen de las pruebas en línea**

Las dos pruebas en línea realizadas mostraron que los algoritmos modificados tienen un desempeño mucho mejor que los algoritmos clásicos.

Las diferencias en porcentaje en el caso del conjunto de datos movielens-100k son mayores que las de la prueba fuera de línea para el mismo conjunto de datos: 55,34%, 40,19% y 35,61% para NCDG y 36,13% y 30,31% para AP.

Para el conjunto de datos “pelis que me gustan”, las diferencias en porcentaje entre los valores de las métricas también son importantes: 152,8% para NCDG y 132,8% para AP.

## 5. Conclusiones y trabajos futuros

---

En este capítulo presentamos las conclusiones de esta tesis y los posibles trabajos futuros que pueden ser desarrollados para extender lo presentado.

### 5.1 Contribuciones realizadas

En este trabajo de tesis presentamos modificaciones a los algoritmos clásicos de filtrado colaborativo basado en memoria para que utilicen el contexto de elección del usuario en sus predicciones de las preferencias de los usuarios. Las modificaciones propuestas son simples y de fácil implementación en los sistemas de recomendación en producción.

Además, desarrollamos un sistema de recomendación que implementa dichas modificaciones para poder crear un conjunto de datos que contenga el contexto de elección del usuario, ya que no se encuentran publicados conjuntos de datos que lo contengan. Finalmente, realizamos una serie de experimentos sobre tres conjuntos de datos, dos conjuntos clásicos y el creado por nosotros, para verificar que nuestra propuesta mejora el rendimiento de los algoritmos clásicos.

La principal contribución de este trabajo es haber verificado que las modificaciones propuestas mejoran significativamente el desempeño de los algoritmos clásicos de filtrado colaborativo basados en memoria. En las pruebas fuera de línea las mejoras van desde el 10% en los algoritmos en el filtrado colaborativo de ítems hasta el 30% en los de usuarios y superan el 100% en las pruebas en línea.

La segunda contribución de esta tesis es la creación de un conjunto de datos que contiene el contexto de elección del usuario que permitirá la experimentación con otros algoritmos que utilicen esta información. Nuestra contribución final es el código fuente de nuestra implementación de los algoritmos modificados<sup>9</sup> que puede ser utilizado como plataforma para nuevos trabajos de investigación.

---

<sup>9</sup> <http://pelisquemegustan.inductia.com/fuentes>

## 5.2 Trabajos futuros

Este trabajo podría ser extendido de diversas maneras en el futuro. En principio se podría verificar si tomar como contexto de elección del usuario sólo aquellos ítems ignorados que se encuentran antes que el último seleccionado en la lista de ítems recomendados mejora el desempeño de los algoritmos de recomendación.

Además, las relaciones entre los ítems seleccionados y los ignorados podrían tratarse como relaciones de preferencia y verificar si el orden en que son seleccionados los ítems por el usuario incide significativamente en la performance de los algoritmos. Por ejemplo, si el usuario seleccionó el ítem 4, luego el 2 y por último el 1, significa que su preferencia por el ítem 1 es mayor que la de los ignorados, la del ítem 2 es más que la del 1 y que la del ítem 4 más que la del 2.

Otra posible extensión es usar los ítems ignorados que se encuentran en el contexto de elección en el cálculo de las funciones de similitud entre usuarios. Esto es, definir que dos usuarios van a ser similares si seleccionan e ignoran los mismos ítems.

También podrían modificarse los modelos basados en contenido para que utilicen la información de contexto de elección. Por ejemplo, si en un sistema de recomendación de películas el usuario usualmente ignora las comedias románticas, no deberían ser recomendadas películas de este género.

Finalmente podría extenderse el modelo propuesto para abarcar otros grados de calificación además del unario, por ejemplo calificaciones ordinales o numéricas.

## 6. Bibliografía

---

- [1] F. Ricci, L. Rokach, B. Shapira, F. Ricci, L. Rokach, B. Shapira y P. B. Kantor, «Recommender Systems Handbook,» Springer US, 2011, pp. 1--35.
- [2] M. D. Gemmis, L. Iaquinta, P. Lops, C. Musto, F. Narducci y G. Semeraro, «Preference learning in recommender systems,» 2009.
- [3] R. Anand, J. Ullman y J. Leskovec, *Mining of Massive Datasets*, Cambridge University Press, 2011, p. 307.
- [4] P. Melville, R. J. Mooney y R. Nagarajan, «Content-boosted collaborative filtering for improved recommendations,» 2002.
- [5] M. J. Pazzani, «A Framework for Collaborative, Content-Based and Demographic Filtering,» *Artif. Intell. Rev.*, vol. 13, pp. 393--408, dec 1999.
- [6] M. Balabanovic y Y. Shoham, «Fab: content-based, collaborative recommendation,» *Communications of ACM*, vol. 40, pp. 66--72, mar 1997.
- [7] P. Cotter y B. Smyth, «PTV: Intelligent Personalised TV Guides,» 2000.
- [8] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes y M. Sartin, *Combining Content-Based and Collaborative Filters in an Online Newspaper*,
- [9] R. J. Mooney y L. Roy, «Content-based book recommending using learning for text categorization,» 2000.
- [10] V. Maccatrozzo, «Burst the filter bubble: using semantic web to enable serendipity,» pp. 391--398, 2012.
- [11] V. Formoso, D. Fernández, F. Cacheda y V. Carneiro, «Using profile expansion techniques to alleviate the new user problem,» *Information Processing & Management*, 2012.
- [12] I. Fernández-Tobías, I. Cantador, M. Kaminskis y F. Ricci, «Cross-domain recommender systems: A survey of the state of the art,» 2012.
- [13] X. Liu y K. Aberer, «SoCo: a social network aided context-aware recommender system,» 2013.
- [14] J. S. Breese, D. Heckerman y C. Kadie, «Empirical analysis of predictive

- algorithms for collaborative filtering,» 1998.
- [15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom y J. Riedl, «GroupLens: an open architecture for collaborative filtering of netnews,» 1994.
- [16] J. L. Herlocker, J. A. Konstan, L. G. Terveen y J. T. Riedl, «Evaluating collaborative filtering recommender systems,» *ACM Trans. Inf. Syst.*, vol. 22, pp. 5--53, jan 2004.
- [17] X. Su y T. M. Khoshgoftaar, «A survey of collaborative filtering techniques,» *Adv. in Artif. Intell.*, vol. 2009, pp. 4:2--4:2, jan 2009.
- [18] G. Linden, B. Smith y J. York, «Amazon.com Recommendations: Item-to-Item Collaborative Filtering,» *IEEE Internet Computing*, vol. 7, pp. 76--80, jan 2003.
- [19] B. Sarwar, G. Karypis, J. Konstan y J. Riedl, «Item-based collaborative filtering recommendation algorithms,» 2001.
- [20] J. Bobadilla, F. Ortega, A. Hernando y J. Bernal, «A collaborative filtering approach to mitigate the new user cold start problem,» *Knowledge-Based Systems*, vol. 26, pp. 225--238, 2012.
- [21] A. Kumar y A. Sharma, «Alleviating Sparsity and Scalability Issues in Collaborative Filtering Based Recommender Systems,» 2013.
- [22] D. Billsus y M. J. Pazzani, «Learning Collaborative Information Filters,» de *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.
- [23] Y. Koren, R. Bell y C. Volinsky, «Matrix Factorization Techniques for Recommender Systems,» *Computer*, vol. 42, pp. 30--37, aug 2009.
- [24] M. H. Aghdam, M. Analoui y P. Kabiri, «Application of nonnegative matrix factorization in recommender systems,» 2012.
- [25] S. Gong, «A collaborative filtering recommendation algorithm based on user clustering and item clustering,» *Journal of Software*, vol. 5, pp. 745--752, 2010.
- [26] G. Adomavicius, A. Tuzhilin, S. Berkovsky, E. W. De Luca y A. Said, «Context-Aware Recommender Systems: Research Workshop and Movie Recommendation Challenge,» 2010.
- [27] G. Adomavicius, R. Sankaranarayanan, S. Sen y A. Tuzhilin, «Incorporating contextual information in recommender systems using a multidimensional

- approach,» *ACM Transactions on Information Systems (TOIS)*, vol. 23, pp. 103--145, 2005.
- [28] C. Palmisano, A. Tuzhilin y M. Gorgoglione, «Using context to improve predictive modeling of customers in personalization applications,» *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, pp. 1535--1549, 2008.
- [29] U. Panniello, A. Tuzhilin y M. Gorgoglione, «Comparing context-aware recommender systems in terms of accuracy and diversity: which contextual modeling, pre-filtering and post-filtering methods perform the best,» 2012.
- [30] A. Karatzoglou, X. Amatriain, L. Baltrunas y N. Oliver, «Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering,» 2010.
- [31] S. Rendle, C. Freudenthaler, Z. Gantner y L. Schmidt-Thieme, «BPR: Bayesian personalized ranking from implicit feedback,» 2009.
- [32] T. Joachims, «Optimizing search engines using clickthrough data,» 2002.
- [33] S.-H. Yang, B. Long, A. J. Smola, H. Zha y Z. Zheng, «Collaborative competitive filtering: learning recommender using context of user choice,» 2011.
- [34] G. A. Miller, «The magical number seven, plus or minus two: some limits on our capacity for processing information.,» *Psychological Review*, vol. 63, nº 2, pp. 81-97, Mar 1956.
- [35] J. Nielsen, "F-Shaped Pattern For Reading Web Content," [Online]. Available: <http://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>. [Accessed Dec 2013].
- [36] G. Shani y A. Gunawardana, «Evaluating recommendation systems,» pp. 257--297, 2011.
- [37] M. Pazzani y D. Billsus, «Learning and Revising User Profiles: The Identification of Interesting Web Sites,» *Mach. Learn.*, vol. 27, pp. 313--331, jun 1997.

## 7. Agradecimientos

---

En primer lugar me gustaría agradecer a mi director, Dr. Marcelo Armentano, por su ayuda, paciencia y orientación durante este año de trabajo. Además, me gustaría reconocer a mis compañeros de clase y profesores por su compromiso y apoyo durante la cursada de la maestría y a los voluntarios que usaron el sistema de recomendación desarrollado. Por último, quiero expresar mi agradecimiento a las autoridades de la maestría por haberme brindado la oportunidad de cursarla.