

UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación
e Inteligencia Artificial



**DISEÑO Y VALIDACIÓN DE MODELOS PARA
SISTEMAS DE RECOMENDACIÓN
MEMORIA QUE PRESENTA**

Sergio Cleger Tamayo

**PARA OPTAR AL GRADO DE
Doctor en Informática**

Directores:

**Prof. Dr. Juan Francisco Huete Guadix
Prof. Dr. Juan Manuel Fernández Luna**

Granada, Octubre de 2012

Editor: Editorial de la Universidad de Granada
Autor: Sergio Cleger Tamayo
D.L.: GR 2254-2013
ISBN: 978-84-9028-630-2

La memoria titulada *Diseño y validación de modelos para sistemas de recomendación*, que presenta D. Sergio Cleger Tamayo, profesor de la Universidad de Holguín (Cuba), para optar por el grado de Doctor, ha sido realizada bajo la dirección de Juan Manuel Fernández Luna y Juan Francisco Huete Guadix, Profesores Titulares del Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada (España).

Granada, octubre de 2012

Los directores

Fdo.: J. M. Fernández Luna Fdo.: J. F. Huete Guadix

El doctorando

Fdo.: Sergio Cleger Tamayo

*A mi familia, en especial a mis padres, María y Román,
a mis abuelos,
y a mi hermana.*

“El secreto del éxito es la constancia en el propósito”
Benjamín Disraeli

Agradecimientos

Cuando comencé esta etapa académica de mi vida, no imagine que estaría escribiendo los agradecimientos de esta memoria. Ciertamente es algo que todos anhelamos cuando iniciamos un camino tan complejo, pero cuesta tanto, que es mejor llegar al final y no pensar tanto en cuando llegará.

Sin proponerme contar toda la historia, bien es cierto que han sido más de cuatro años de gran dedicación, altas y bajas, pero total satisfacción de hacer algo que me place enormemente, investigar. A mediados del año 2008 y motivado por el área inicialmente por Raúl Menéndez, ya estaba sobre este barco, rodeado de un mar de conocimiento donde al mismo tiempo disfrutaba y me asustaba de saber que había muchísimo más que aprender.

En los cursos de doctorado recibidos en la Universidad Central de Las Villas, Cuba, conocí a mis tutores y directores de tesis, Dr. Juan Francisco Huete Guadix y Dr. Juan Manuel Fernández Luna. Los encuentros iniciales permitieron definir la línea de investigación a seguir, los Sistemas de Recomendación. Desde ese momento y ante disímiles situaciones, hemos obtenido los resultados que se presentan en esta memoria.

Lo primero es agradecer a mis directores de tesis todo el apoyo que me han dado. Sus reflexiones y análisis en todo momento han sido claves para llegar hasta aquí. Los encuentros con el doctor Huete, casi diarios, me aportaban un enorme conocimiento. Siempre recuerdo sus consejos científicos. Los intercambios con el doctor Fernández me ayudaron a no claudicar. Ambos directores me han hecho crecer y no tendré nunca palabras para describirlos.

Muchos doctorandos han dicho haber tenido los mejores directores de tesis, yo solo puedo decir que mis directores de tesis son únicos, y espero que la mayoría de sus expectativas sobre mí respecto a la investigación, se hayan cumplido satisfactoriamente. Agradecerles toda la paciencia ante las múltiples y en ocasiones reiteradas equivocaciones. Agradecerles por su atención y dedicación sin importar día ni hora.

Todos los resultados que en esta memoria se describen, se han obtenido en gran medida gracias a la profesionalidad y exigencias de mis directores, y más que directores, amigos, Juan y Juanma. Gracias por todo, sin ustedes no hubiese sido posible.

Otras personas en lo académico han sido fundamentales en toda la etapa de investigación, mencionarlos a todos es imposible, pero espero no olvidar los principales nombres, y de

ocurrir, mis disculpas. A Julio César Rodríguez Cano, mi compañero y amigo de mil batallas, con quien he transitado desde el 2008, muchas gracias amigo. A Leandro por su apoyo incondicional en todo momento, gracias.

Profesores que tanto en Granada como en Cuba han hecho posible la continuidad de este programa doctoral, Verdegay, Raúl, Rafael Bello, Rosa, a todos, muchas gracias.

A los profesores andaluces y cubanos que nos aportaron su saber; al grupo de doctorandos de SoftComputing que hemos compartido en todo este tiempo, en especial a Oscar, Angel, Pavel, Guanín, Carlos, Marvín, Isvani y Agustín; a Rosa Urquiza, María del Carmen, Rita, Carmen, Yanet, Luis Angel, Humberto, Scull, Félix, Pedro, Cuevas, Raúl Ernesto, Ramiro, Reynier, Diana, Daymi e Isis, amigos en los que he podido apoyarme para poder continuar y que en todo momento han mostrado interés por mí, gracias a todas y todos. Gracias a Kapil, Raj, Dinesh y Mark, con los que pude contar cuando los necesité.

A los compañeros del periódico ¡Ahora!, quienes me apoyaron en una parte de la investigación, especialmente a Jorge Luis y Karina, muchas gracias.

En lo familiar, mis padres, María y Román, quienes han sido faro durante toda mi vida y me han apoyado sin dudarlos en todas mis decisiones académicas y profesionales. Todo este tiempo en el que he estado lejos de ellos y han sufrido mi ausencia, no han dejado de animarme a continuar la investigación. Junto a ellos mi hermana Karina, quien igual me ha apoyado en todo durante este periodo. A ellos, mil gracias.

En Granada encontré un grupo de personas especiales, que me acogieron como hijo y amigo. Ellos lograron que este último año no fuese tan largo y que me sintiera como en casa. Sus nombres quedarán en esta tesis y también en mi memoria, por un lado Mary, Padilla, Angela y Antonio, por otro, Juan, Palmira y Palmi. A todos, muchas gracias. Gracias también a David por sus reflexiones y su sincera e incondicional amistad.

Agradecer a la Asociación Universitaria Iberoamericana de Postgrado (AUIP) y a la Agencia Española de Cooperación Internacional para el Desarrollo (AECID) por las becas concedidas, y que en gran parte facilitaron la realización y culminación de toda la investigación.

Finalmente, agradecer al grupo de investigación al que pertenezco, Tratamiento de la Incertidumbre en Inteligencia Artificial, y en especial a su director, Dr. Luis M. de Campos. La utilización del equipamiento, servidores y la financiación por parte de los proyectos asociados al grupo, TIN2008-06566-C04-01, P09-TIC-04526, TIN2011-28538-C02-02, TIC-04526 y MIPRCV (CSD2007-00018), han sido muy importantes para finalizar esta investigación.

En fin, a todos los que han hecho posible todo esto, familiares, amigos y colegas, los mencionados y los omitidos involuntariamente, muchísimas gracias.

Trabajos publicados

Artículos en Revistas

- “*Top-N news recommendations in digital newspapers*”. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete. Knowledge-Based Systems (KBS). 27: 180 - 189 (2012)
- “*Explanation in Recommender Systems*”. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete. Sometido a User Modeling and User-Adapted Interaction (UMUAI), the Journal of Personalization Research. (2012)
- “*Learning from Explanations in Recommender Systems*”. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete. Sometido a Knowledge and Information Systems (KAIS), an International Journal. (2012)

Congresos Internacionales

- “*Una propuesta para la recomendación de noticias basada en técnicas de agrupamiento*”. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete; Ramiro Pérez-Vázquez. Tendencias de Soft Computing, Editorial Feijóo, UCLV. 2009. ISBN-959250525-4
- “*Alternativa al proceso social de recomendación: Sistemas de Recomendación colaborativos*”. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete; Ramiro Pérez-Vázquez. IV Conferencia Científica Internacional de la Universidad de Holguín 2009 ISBN 978-959-16-1010-2

- *“Proposal for an Experimental Platform in collaborative Information Retrieval”*. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete; Ramiro Pérez-Vázquez. Proceedings of 2009 International Symposium on Collaborative Technologies and Systems. Maryland, USA, May 2009. 485 - 493, ISBN 978-1-4244-4586-8, William McQuay, Waheed W. Smari (Eds.)
- *“A Proposal for News Recommendation Based on Clustering Techniques”*. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete; Ramiro Pérez-Vázquez; Julio C. Rodríguez Cano. The Twenty Third International Conf on Industrial, Engineering & Other Applications of Applied Intelligent Systems. IEA/AIE (3) Junio 2010: 478 - 487. ISBN 0302-9743.
- *“Estudio de Semántica Latente en la Personalización del Semanario Digital ¡ahora!”*. V Conferencia Científica Internacional de la Universidad de Holguín, Abril 2011. ISSN 978-959-16-1329-5
- *“Explaining Neighborhood-based Recommendations”*. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete. 35th Annual SIGIR Conference. Proceedings of the ACM SIGIR 2012 Conference. Portland, Oregon, USA August 12–16, 2012. 1063 - 1064. ISBN 978-1-4503-1472-5. Association for Computing Machinery, ACM.
- *“On the use of Weighted Mean Absolute Error in Recommender Systems”*. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete. Association for Computing Machinery Recommender Systems, RecSys 2012 International Workshop on Recommendation Utility Evaluation: Beyond RMSE. RUE 2012. Dublin, Ireland, September, 9.

Congresos Nacionales

- *“A New Criteria for Selecting Neighborhood in Memory-Based Recommender Systems”*. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete. 14th Conference of the Spanish Association for Artificial Intelligence, CAEPIA. La Laguna, Spain, November 7-11, 2011: 423 - 432. Lecture Notes in Computer Science. Volumen 7023. ISBN 978-3-642-25273-0.

- *“Propuesta de Interfaz de Explicación para la recomendación de noticias”*. Sergio Cleger-Tamayo; Juan M. Fernández-Luna; Juan F. Huete; Roi Blanco. II Congreso Español de Recuperación de Información. 145 - 150, 2012 (ISBN: 978-84-8021-860-32). Rafael Berlanga y Paolo Rosso (Editores).

Capítulos de Libros

- *“Motores de Búsqueda de Código Abierto”*. Sergio Cleger-Tamayo; Julio C. Rodríguez Cano. Capítulo del libro “Recuperación de Información. Un enfoque práctico y multidisciplinar”. Fidel Cacheda Sijo, Juan Manuel Fernández Luna, Juan Francisco Huete Guadix (Editores). ISBN:978-84-9964-112-6, Editorial RA-MA, impreso en España en septiembre de 2011. pp 814.

Índice general

Índice general	XVII
Índice de figuras	XXI
Índice de tablas	XXV
1. Introducción y motivación	1
2. Introducción a los sistemas de recomendación	7
2.1. Introducción a los sistemas de recomendación	7
2.1.1. Principales modelos de recomendación	12
2.2. Sistemas de recomendación de filtrado colaborativo	18
2.2.1. Etapas del filtrado colaborativo	19
2.2.2. Métodos de filtrado colaborativo	23
2.2.2.1. Algoritmo <i>Slope One</i>	23
2.2.2.2. Técnica de Factorización de Matrices	24
2.3. Sistemas de recomendación basados en contenido	24
2.3.1. Algoritmos de <i>Clustering</i> o Agrupamiento	27
2.3.1.1. Algoritmo K-Medias	30
2.3.1.2. Algoritmo Aspect Model	31
2.3.1.3. Extensión de Aspect Model	33
2.3.2. Otros algoritmos	34
2.4. Explicación de las recomendaciones	35
2.4.1. Valoración de las explicaciones	42
2.5. Evaluación de los SR	43
2.5.1. Colecciones de datos	43
2.5.2. Métricas de evaluación	44

I	Recomendación basada en filtrado colaborativo	49
3.	Modelos de filtrado colaborativo	55
3.1.	Selección de vecinos con alta calidad predictiva	56
3.1.1.	Impacto de características en la recomendación	60
3.2.	Experimentación	61
3.2.1.	Metodología de experimentación	61
3.2.2.	Resultados y análisis	62
3.2.2.1.	Evaluación de la calidad del vecindario seleccionado para la predicción	62
3.2.2.2.	Evaluación de las dependencias del error en la predicción . .	65
3.3.	Conclusiones	72
4.	Explicación en filtrado colaborativo	77
4.1.	Características para la explicación	78
4.2.	Propuesta de explicación	86
4.3.	Interfaz de explicación para un SR de películas	91
4.3.1.	Ejemplos de uso de la interfaz de explicación	94
4.4.	Experimentación	97
4.4.1.	Interacción de usuarios	98
4.4.2.	Resultados y análisis	101
4.4.2.1.	Eficacia de la explicación	102
4.4.2.2.	Eficiencia y persuasión de la explicación	108
4.4.2.3.	Satisfacción de los usuarios con las explicaciones	114
4.5.	Conclusiones	116
5.	Aprendizaje basado en la explicación	119
5.1.	Evidencias empíricas de las explicaciones	119
5.2.	Selección de características	125
5.3.	Aprendizaje de la explicación	127
5.3.1.	Modelo propuesto para predecir el error	129
5.3.2.	Consideraciones sobre eficiencia	133
5.4.	Experimentación	134
5.4.1.	Resultados y análisis	135
5.4.1.1.	Corrección del error en las predicciones para todos los usuarios	135
5.4.1.2.	Desempeño de las predicciones en usuarios candidatos a co- rregir el error	137
5.4.1.3.	Análisis de la importancia de las características de la interfaz de explicación	143

5.5. Conclusiones	151
II Recomendación basada en contenido	155
6. Modelos basados en contenido	161
6.1. Recomendación en entornos noticiosos	162
6.2. Modelos propuestos	165
6.2.1. Modelo Dyadic	166
6.2.2. Modelo Triadic	168
6.3. Experimentación	170
6.3.1. Colección de prueba	171
6.3.2. Metodología de experimentación	175
6.3.3. Resultados y análisis	178
6.3.3.1. Predicción de <i>ranking</i> global	179
6.3.3.2. Predicción de <i>ranking</i> basado en las categorías de noticias .	180
6.3.3.3. Análisis comparativo entre las categorías	183
6.3.3.4. Análisis de comportamiento de preferencias en el tiempo . .	184
6.4. Conclusiones	187
7. Explicación en filtrado cognitivo	191
7.1. Propuesta de explicación	192
7.1.1. Ejemplos de uso de la interfaz de explicación	197
7.2. Evaluación de la propuesta	200
7.3. Conclusiones	203
8. Conclusiones y trabajos futuros	207
8.1. Conclusiones	207
8.2. Trabajos futuros	210
Apendice A	213
Bibliografía	217

Índice de figuras

2.1.	SR de películas Netflix.	11
2.2.	Esquema general del proceso de recomendación.	13
2.3.	Tareas en los sistemas de filtrado colaborativo.	14
2.4.	Representación de la información contenida en los perfiles de usuarios.	25
2.5.	Ejemplo donde al aplicar un algoritmo de agrupamiento se obtienen dos o cuatro <i>clusters</i>	29
2.6.	Ejemplo al aplicar el algoritmo k-medias para k igual seis.	31
2.7.	Representación gráfica asimétrica de Aspect Model.	32
2.8.	Representación gráfica simétrica de Aspect Model.	32
2.9.	Interfaz de explicación. Histograma con los votos del vecindario.	37
2.10.	Interfaz de explicación en Amazon, basada en contenido de ítems valorados previamente.	39
2.11.	Histograma que agrupa los votos de los vecinos.	40
2.12.	MoviExplain, interfaz de explicación basada en autores presentes en películas vistas previamente.	41
2.13.	Newsmap, interfaz de explicación de un sistema de recomendación noticioso.	42
2.14.	Distribución de votos en el conjunto de datos MovieLens.	44
2.15.	Popularidad de los ítems en el conjunto de datos MovieLens.	45
3.1.	Predicciones de los usuarios para las valoraciones de Ann.	58
3.2.	Escala de colores para representar el voto.	58
3.3.	Pesos con correlación positiva en la $wMAE$	67
3.4.	Pesos con correlación negativa en la $wMAE$	68
3.5.	Relación entre la $wMAE$ y la MAE	69
3.6.	Relación entre el peso y el error utilizando $rU+$	70
3.7.	Relación entre el peso y el error utilizando $rI+$	71
3.8.	Comportamiento de modelos respecto a la métrica $wMAPE$	72
4.1.	Relación entre los votos del usuario activo y el vecindario (ítems en común).	80

4.2.	Escala de colores para representar: i) el voto del usuario activo y ii) la diferencia de votos.	80
4.3.	Representación del comportamiento del vecindario al ordenar por el voto de Ann.	82
4.4.	Representación del comportamiento del vecindario al agrupar los votos de Ann y ordenar dentro de cada grupo por la predicción del vecindario.	84
4.5.	Representación al ordenar según el error de las predicciones de los ítems por el vecindario.	85
4.6.	Representación al ordenar según la similitud en las predicciones respecto al ítem objetivo.	86
4.7.	Criterio del vecindario acerca de los votos previos del usuario activo.	88
4.8.	Propuesta de explicación. Error promedio de votación vs Similitud en las predicciones.	89
4.9.	Propuesta de explicación. Se utiliza la diferencia del voto sugerido por el vecindario y el voto dado por el usuario para colorear cada figura.	90
4.10.	Ejemplo de explicación para la película “Desperado”.	91
4.11.	Ejemplo de explicación para la película “Scarface”.	92
4.12.	Propuesta de interfaz de explicación para SR de películas.	92
4.13.	Histogramas de predicción de errores. Subestimación (Izquierda), Acierto (Centro) y Sobreestimación (Derecha).	94
4.14.	Ejemplo de interfaz de explicación para SR de películas. Caso donde la recomendación se aceptaría.	95
4.15.	Ejemplo de interfaz de explicación para SR de películas. Caso donde la recomendación se pudiera modificar.	96
4.16.	Interfaz de presentación del prototipo desarrollado.	99
4.17.	Ejemplo de interfaz de la primera etapa, Basada en Vecindario.	101
4.18.	Ejemplo de interfaz de la segunda etapa, Basada en Historial.	102
4.19.	Ejemplo de interfaz de la tercera etapa, Pantalla Basada en Vecindario+Historial.	104
4.20.	Ejemplo de interfaz de Cuestionario. Primera pregunta.	105
5.1.	Ejemplo de histograma con las votos del vecindario.	121
5.2.	Ejemplo de correlación del error de la recomendación vs número de ítems en la parte izquierda de la interfaz de explicación propuesta en el Capítulo 4.	124
5.3.	Ejemplo de interfaz de explicación.	127
5.4.	Visualización del modelo de aprendizaje del error.	128
5.5.	Árbol de modelo $M5'$ para el usuario 507	132
5.6.	Mejora de la MAE al aplicar un simple algoritmo de regresión lineal.	136
5.7.	Mejoras en la MAE para ($\#I = 20$).	138
5.8.	Número de reglas lanzadas para cada uno de los conjuntos de datos considerados.	149

5.9.	Ganancia absoluta de la MAE para cada conjunto de datos según las características consideradas.	150
6.1.	Representación gráfica del modelo Dyadic.	167
6.2.	Representación del modelo para usuarios que acceden a noticias.	167
6.3.	Representación gráfica del modelo Triadic.	168
6.4.	Número de accesos por categorías	174
6.5.	Totales de direcciones IP que acceden por categorías	174
6.6.	P@ac para las 16 categorías más frecuentes.	184
7.1.	Representación del perfil general del usuario y relevancia en este de la noticia recomendada.	194
7.2.	Representación del perfil específico de la categoría Deporte y relevancia en este de la noticia recomendada.	195
7.3.	Selección de categorías para mostrar noticias.	195
7.4.	Noticias que soportan la explicación.	196
7.5.	Ejemplo de nicho de noticias similares que soportan la explicación.	197
7.6.	Resumen emergente de noticia seleccionada.	197
7.7.	Interfaz de explicación para sistemas de recomendación basados en contenido.	198
7.8.	Ejemplo de interfaz de explicación propuesta, para una noticia de la categoría Economía. Selección de noticias de todas las categorías.	199
7.9.	Ejemplo de interfaz de explicación propuesta, para una noticia de la categoría Economía. Selección de noticias de las categorías Economía y Política.	200
7.10.	Ejemplo de interfaz de explicación propuesta, para una noticia de la categoría Economía. Selección de noticias de la categoría Economía.	201

Índice de tablas

2.1.	Tus vecinos valoraron la película “X”.	41
3.1.	Votos de usuarios. El objetivo es proveer una predicción a Ann para el ítem I_7	56
3.2.	Modelo de recomendación. Selección de vecinos con buena calidad predictiva.	60
3.3.	Desempeño de la métrica RMSE con al menos 10 votos en común.	63
3.4.	Desempeño de la métrica Hamming con al menos 10 votos en común.	63
3.5.	Desempeño de la métrica MAE con al menos 10 votos en común.	64
3.6.	Desempeño de la métrica RMSE con al menos 1 voto en común.	64
3.7.	Desempeño de la métrica Hamming con al menos 1 voto en común.	64
3.8.	Desempeño de la métrica MAE con al menos 1 voto en común.	65
4.1.	Votos de usuarios. El objetivo es proveer a Ann de una explicación para el ítem $I_t = I_7$	78
4.2.	Ejemplo de Interfaz de Explicación para Ann. Voto de los vecinos para el ítem I_7	79
4.3.	Resultados generales de la primera etapa.	103
4.4.	Resultados generales de la segunda etapa.	106
4.5.	Beneficios al evaluar las explicaciones en las dos primeras etapas.	107
4.6.	Resultados generales de la tercera etapa.	109
4.7.	Resultados de las recomendaciones comunes entre las tres primeras etapas.	110
4.8.	Resultados generales de la cuarta etapa. Elección de las recomendaciones recibidas.	111
4.9.	Resultados para diferentes valores de recomendaciones. Primera y segunda etapa.	112
4.10.	Resultados para diferentes valores de recomendaciones. Tercera etapa.	113
5.1.	Información de las recomendaciones en ítems previos para un usuario.	122
5.2.	Comportamiento positivo de la propuesta para nuevas recomendaciones del usuario U	123
5.3.	Resultados de la MAE y $rMAE$ sobre las predicciones en todos los usuarios, del modelo basado en las predicciones pasadas, LM.	137
5.4.	Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=15$, en modelo LM.	139

5.5.	Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=20$, en modelo LM.	139
5.6.	Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=25$, en modelo LM.	140
5.7.	Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo LM para $\#I=15$	141
5.8.	Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo LM para $\#I=20$	141
5.9.	Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo LM para $\#I=25$	142
5.10.	Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=15$, en el modelo BL.	142
5.11.	Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=20$, en el modelo BL.	143
5.12.	Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=25$, en el modelo BL.	143
5.13.	Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo BL para $\#I=15$	144
5.14.	Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo BL para $\#I=20$	144
5.15.	Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo BL para $\#I=25$	145
5.16.	Valores de RMSE sobre las predicciones en usuarios a los que se le aplica la propuesta, para el modelo LM.	145
5.17.	Valores de RMSE sobre las predicciones en usuarios a los que se le aplica la propuesta, para el modelo BL.	146
5.18.	Resultados generales al comprobar la importancia de las características al predecir el error de las predicciones.	147
6.1.	Ejemplo de una matriz de accesos realizados por los usuarios a las categorías de noticias	172
6.2.	Categorías y sus identificadores (Id).	173
6.3.	Predicción de un <i>ranking</i> global para todas las noticias publicadas en un día determinado, \mathcal{N}	179
6.4.	Macro y micro precisión: filtrado por categorías de noticias del modelo general.	181
6.5.	Macro y micro precisión: modelos independientes basados en categorías de noticias.	183
6.6.	$P@ac$ cuando empleamos perfiles para corto (ST) y largo (LT) plazo.	186

Capítulo 1

Introducción y motivación

Hoy en día, con la elevada cantidad de información disponible, es muy difícil no encontrarnos en Internet con sitios que hagan recomendaciones de sus productos o servicios. La popularidad con la que cuentan los Sistemas de Recomendación (SR), en gran medida por la posibilidad que brindan a los usuarios de descubrir contenido relevante [75, 78], es notable en muchísimas aplicaciones de comercio electrónico [34, 48, 75]. Con el auge de las redes sociales en la actualidad, su utilización es más prominente.

Un SR es un tipo específico de aplicación de filtrado de información, muy utilizado para presentar a los usuarios ítems que les resulten de interés. Estos sistemas se basan en las preferencias de los usuarios para recomendar ítems en concreto [34, 93]. Entre otras, se destacan como principales aproximaciones las basadas en contenido y las de filtrado colaborativo. Las primeras suelen ser utilizadas en aplicaciones donde fácilmente se pueda disponer del contenido de los ítems [6]. Mientras el filtrado colaborativo se apoya en el criterio de colaboradores con gustos similares [59, 110]. Dependiendo del entorno en el que se quieran aplicar, se emplean estos u otros tipos de SR, que en muchos casos se combinan logrando sistemas híbridos [7, 21, 22, 28, 35, 99, 124].

Aunque la utilización de los SR se incrementa, debemos destacar algunos problemas que estos sistemas presentan y en los que se realizan continuas investigaciones [10, 47, 58, 64, 67, 75, 81, 105, 118]. Por ejemplo, los usuarios nuevos de los que no se conocen sus preferencias y resulta muy difícil ofrecerles recomendaciones, los nuevos productos que inicialmente no han sido valorados, y los usuarios que falsean información [1]. La desconfianza de un gran número de usuarios en los SR es otro tema que mantiene activa a la comunidad científica, por un lado, los usuarios en la Web no les gusta que utilicen sus datos públicamente, y estos sistemas necesariamente los utilizan en sus mecanismos de recomendación [2, 41, 77], mientras por otro lado, muchas veces los usuarios no comprenden la razón de la recomendación y prefieren desecharla.

Sin lugar a dudas, en un mundo digital donde cada día cientos de personas se incorporan a la red y disfrutan de sus servicios, es necesario buscar soluciones a los problemas anteriores, que además incrementen la precisión y escalabilidad de los algoritmos [105]. Siendo por tanto las investigaciones de los SR un campo abierto de investigación, en donde enmarcamos el trabajo que hemos desarrollado.

Objetivos

El objetivo principal de esta investigación es el diseño de modelos de recomendación que incrementen la eficiencia de la recomendación, su posterior implantación en los SR y su validación. Este objetivo se puede descomponer en los siguientes específicos:

- O1.** Mejorar la calidad de la recomendación de algoritmos de filtrado colaborativo, con una estrategia de selección de los vecinos con mejor características predictivas.
- O2.** Determinar el rendimiento que se puede alcanzar con la estrategia mediante su evaluación con conjuntos de datos. Comparar dicho rendimiento con otros modelos clásicos de filtrado colaborativo.
- O3.** Diseñar algoritmos para SR basados en contenido, que permitan mejorar la calidad de la recomendación y lograr una adecuada personalización.
- O4.** Determinar el rendimiento que se puede alcanzar con el algoritmo mediante su evaluación con conjuntos de datos reales de un entorno determinado. Comparar dicho rendimiento con otros modelos.
- O5.** Diseñar propuestas de explicación para SR, que permitan incrementar la confianza de los usuarios y la reutilización del sistema.
- O6.** Evaluar la calidad de las propuestas mediante la experimentación y evaluación con usuarios reales.

Descripción de la memoria por capítulos

Con la finalidad de describir cómo alcanzamos los objetivos anteriores y para una mejor comprensión de la investigación, hemos dividido esta memoria en siete capítulos. Un capítulo de introducción a los SR, cinco capítulos destinados a describir como se alcanzaron los objetivos y un capítulo final de conclusiones y trabajos futuros.

Capítulo 2. En este primer capítulo exponemos los conceptos generales y básicos de los SR, necesarios para comprender el resto de la memoria. Inicialmente se da una introducción al problema de la recomendación y se describen los diferentes tipos de sistemas existentes. Posteriormente se describen en detalles las principales técnicas que se utilizan en las dos partes en las que se divide el contenido de la memoria, los SR basados en contenido y los de filtrado colaborativo. Al concluir la descripción de estas dos aproximaciones, se aborda el tema de las explicaciones en los SR y se comentan los principales trabajos en el área. A lo largo del capítulo se muestran ejemplos de SR que podemos encontrar. Finalmente se puntualizan los detalles más importantes que se deben conocer para comprender las evaluaciones experimentales que se desarrollan.

Capítulo 3. Este capítulo, junto con los dos siguientes, conforman la parte relacionada con los SR basados en filtrado colaborativo. Se propone un método de selección de los mejores vecinos para realizar la recomendación, abordando los dos primeros objetivos declarados. Este método se describe y ejemplifica detalladamente antes de pasar a comprobar los resultados en la tarea de predicción de votos. En esta experimentación además comprobamos el impacto en la recomendación de determinadas características del conjunto de datos.

Capítulo 4. En este capítulo realizamos una propuesta de explicación para SR de filtrado colaborativo, tratando el quinto de los objetivos específicos señalados. Inicialmente se describe toda la información que se utiliza en la propuesta y se muestran diferentes aproximaciones de la misma. Finalmente, insertamos la explicación dentro de un SR de películas y desarrollamos una evaluación experimental con usuarios reales, dándole respuesta al sexto de los objetivos de esta memoria.

Capítulo 5. La experimentación del capítulo anterior nos permitió comprobar el comportamiento de los usuarios al recibir las explicaciones. A lo largo de este capítulo se intenta detectar el error de las predicciones, basándonos en las explicaciones para mejorar las recomendaciones del SR. En la última parte se exponen los resultados de la experimentación.

Capítulo 6. Este y el siguiente capítulo tratan temas relacionados con los SR basados en contenido. En especial tratamos la recomendación de noticias en un entorno digital y su personalización. Se proponen dos modelos probabilísticos para lograr el tercero de los objetivos trazados. En la evaluación experimental se comprueban finalmente los resultados

que se obtienen para la tarea de encontrar un *ranking* con los mejores ítems, dando respuesta al cuarto de los objetivos.

Capítulo 7. Este es el último capítulo donde se expone contenido relacionado con los objetivos de la investigación. Para dar respuesta al quinto de los objetivos de esta memoria, realizamos una propuesta de explicación para SR basados en contenido. La propuesta se describe detalladamente y mostramos ejemplos de su utilización.

Capítulo 8. Por último, se exponen las conclusiones que se obtienen en el desarrollo de la investigación y se comentan las principales líneas de trabajo futuro.

Capítulo 2

Introducción a los sistemas de recomendación

En el capítulo se abordan las principales características de los SR, necesarias para comprender la totalidad de la memoria. En primer lugar se describen los principales tipos de SR, centrándonos en los que se emplean en el resto de la investigación: los basados en contenido y los de filtrado colaborativo.

Cuando describimos en detalles los SR basados en contenido y los de filtrado colaborativo, profundizamos en las técnicas más relevantes para nuestra investigación.

En este capítulo además, ofrecemos al lector una revisión sobre los principales elementos a considerar en las evaluaciones con los SR. Resaltamos además las métricas empleadas a lo largo de la investigación, tanto en la tarea de predicción de voto, como para la tarea de encontrar un *ranking* los mejores ítems.

Finalmente se incluye un sección donde puntualizamos los avances más destacados respecto al diseño de explicaciones para los SR.

2.1. Introducción a los sistemas de recomendación

Según la Real Academia Española (RAE), recomendar es “aconsejar algo a alguien para bien suyo”. Partiendo de este concepto, se puede decir que un SR es una aplicación que aconseja ítems¹ a los usuarios, consejo informático que ha de resultar beneficioso: en un entorno de libros, aquellos libros que deban ser leídos primero que otros; en una aplicación de información sobre destinos turísticos, los lugares que según las preferencias del usuario le resultarán mas atractivos; en un periódico digital, aquellas noticias que satisfagan los intereses de información; en un sistema fílmico, el orden de películas según una cartelera disponible.

Los SR se definen como un campo de la Inteligencia Artificial en donde los investigadores se centran explícitamente en evaluar lo relevante de aquellos elementos que el usuario no

¹A lo largo de la memoria será empleado el término para referirnos a artículos o productos.

conoce [26]. Se destacan dos tipos fundamentalmente, los SR basados en contenido, en los que se recomienda según lo seleccionado en el pasado por el usuario, y los colaborativos, en los que se recomienda basándose en selecciones de clientes con gustos y preferencias similares [28, 78, 97]. En estos últimos se concibe la idea de una inteligencia colectiva para mostrar información al cliente adaptada a sus gustos, que en muchas ocasiones no estaba esperando encontrar y le puede resultar útil [35, 103].

Estos sistemas desde su surgimiento en la última década del siglo XX [45, 96] hasta la actualidad, han incrementado su presencia en diversos entornos, destacándose los dominios de comercio electrónico que tan populares son en la actualidad. Su habilidad para aprender del comportamiento de los usuarios y descubrir sus preferencias, ha dotado a los propios usuarios de una herramienta útil ante el incremento continuo de información al que se encuentran expuestos. Se pueden encontrar en entornos de libros, música, redes sociales, destinos turísticos, películas o simplemente noticias [112].

De forma general los SR suelen ser muy útiles pero pocos usuarios contribuyen a ellos directamente, pues de manera general se cumple la idea de la “regla del 1%”, la que plantea que del total de usuarios que acceden a una aplicación, sólo un pequeño porcentaje llega a realizar contribuciones reales en un sitio colaborativo (por ejemplo, subir un video a YouTube), mientras que alrededor del 10% interactúa con el contenido (realizar valoraciones, recomendaciones). El resto de los usuarios simplemente consulta los contenidos que se le muestran.

De forma general, los objetivos fundamentales de los SR en la actualidad [1, 59, 107], la solución a problemas de sobrecarga de información, mostrar información a usuarios según sus gustos y preferencias, facilitar relaciones entre clientes, protección de información (Sistemas de Censura), así como conocer a los clientes para satisfacer mejor sus necesidades.

Para muchos administradores de negocios de comercio electrónico sus metas están claras en cuanto a estos tipos de sistemas, pues le garantizan incrementar las ventas de sus productos y hacerlos más competitivos. Podemos destacar las siguientes funcionalidades de estos sistemas:

- Ayudan a nuevos e infrecuentes visitantes mediante amplias listas de recomendación.
- Fomentan la credibilidad a través de una comunidad.
- Intentan fidelizar a la clientela a través de servicios de notificación, envío de información con nuevos productos y (o) descuentos.
- Realizan ventas cruzadas por medio de recomendaciones asociadas por producto.
- Construyen relaciones prolongadas garantizando una personalización profunda con ello.

La recomendación es una conducta social que existe en la sociedad desde hace muchísimo tiempo, pero los SR son relativamente jóvenes cuando se habla del entorno digital. Al que reconocen como el primero de ellos surge en el año 1992, “Tapestry” de Xerox PARC [45]. Este sistema ofrece al usuario recomendaciones de correos electrónicos o noticias y se diseñó para soportar tanto un sistema basado en contenido como uno de filtrado colaborativo.

El surgimiento de Tapestry fue el motor para que múltiples técnicas y herramientas surgieran y se introdujeran en numerosos negocios o entornos. GroupLens fue otro de los sistemas pioneros y recomienda al usuario los artículos que mejor fueron valorados por usuarios que tienen una alta similitud en sus valoraciones [67, 94].

Otros múltiples sistemas y para diversos sectores se diseñaron:

- Fab, por ejemplo, sugiere direcciones de documentos u otros recursos digitales (*Uniform Resource Locator*, URLs) relevantes a los usuarios, combinando las semejanzas de la valoración de los usuarios y de las páginas Web (parte del proyecto de biblioteca digital de Stanford).
- Entrée, mientras tanto, recomienda restaurantes adecuados a los intereses y el presupuesto del usuario. Este último sistema integra técnicas basadas en conocimiento y filtrado colaborativo utilizando valoraciones semánticas en las que un sistema obtiene las razones que hay detrás de una preferencia.

Empresas y compañías actualmente hacen uso de SR como una forma que les permite alcanzar personalización con sus clientes a través de la Web. Es muy común en entornos digitales la recomendación de qué productos comprar o adquirir, así como la ayuda ante una incertidumbre, lo que en un alto número tiende a convertir a esos usuarios beneficiados en activos visitantes y posibles clientes.

Negocios de comercio electrónico como Barnes&Noble y Sleeper, que se dedican a la venta de libros, utilizan un SR partiendo de aquellos productos que fueron adquiridos y han interesado a los usuarios. Otros ejemplos a destacar son Mangarate, E!Online, Finnkino y Netflix para películas; CdNow, Firefly y Ringo [110] en las preferencias musicales; B2B en lo relacionado a Software - Hardware; Jester [48] para recomendar chistes, así como Ebay [34], ShopMatcher, TripMatcher, Macromedia, Doubleclick en temáticas generales.

MyStrands y LastFm son otros ejemplos de SR musicales, basados en la propia realización de los usuarios, asociando a cada usuario un perfil formado por las experiencias del resto de usuarios con gustos similares para luego recomendar música o videos que están también asociados al mencionado perfil.

El SR 11870.com, surgió con la intención de dar forma a una comunidad de usuarios que recomiendan servicios cotidianos variados (desde un estilista hasta un restaurante) a otros

usuarios. Este sistema realiza recomendaciones a los usuarios registrados en función del perfil creado, de su localización y (o) de los negocios/servicios que él ha recomendado.

Indudablemente el uso de los SR se ha extendido rápidamente, hoy se pueden encontrar en bibliotecas digitales [93], brindando ayuda a los usuarios para encontrar libros que se ajusten a las preferencias de cada uno y con la ayuda colaborativa. El uso en el sector de la protección, fundamentalmente dirigido a evitar el acceso de niños a material inadecuado en Internet, se puede ver en Cyberpatrol y Cybersitte. Mientras, referente al contexto noticioso, además de los mencionados, pueden verse otros ejemplos de recomendadores en Net Perceptions, Shift, Infoscan, Le Monde, entre otros.

Un entorno Web que puede considerarse un SR es Facebook: una aplicación Web formada por muchas redes sociales relacionadas con una escuela, universidad, trabajo, región u otros factores definidos por los propios usuarios. Dicho sistema reconoce al usuario y, enfocado en las relaciones que se van produciendo con otros usuarios y en las características demográficas, recomienda servicios y (o) usuarios que suelen resultar de interés.

El interés por los SR ha sido tal, que la inmensa mayoría de los más importantes sitios de comercio electrónico no conciben su negocio sin estos. Por ejemplo, Amazon.com [75], donde se realizan recomendaciones de libros, CDs y otros productos, y en cuyo negocio Jeff Bezos, alegando a la excelencia dada la implementación de los SR, expresó: “Cada uno de nuestros clientes tendrá una tienda a su medida”. Además se tiene a MovieLens.com donde se recomiendan películas, o AdaptiveInfo.com, un SR de noticias.

En la Figura 2.1 se muestra una interfaz de Netflix, que en el sector de las películas es uno de los SR que mayor auge ha tenido desde su surgimiento en el año 1997, incrementando sus índices de alquiler no solo de las películas nuevas, sino en las más antiguas y menos comerciales, mostrando que estos sistemas animan a los usuarios a adquirir más productos, por un lado, y por otro, les muestra aquellos productos que quizás nunca conocerían.

De manera general, los SR pueden ser un hecho en cualquier negocio o sistema que tenga muchas opciones u opiniones para ayudar a los usuarios a decidir (toma de decisiones). Muchos son los sectores que para lograr ser más competitivos usan los SR o tienen planes para hacerlo. La industria televisiva, la radial, localización de expertos, búsqueda de personas afines, servicios de noticias y (o) mensajes, son claros ejemplos.

En todas las aplicaciones mencionadas anteriormente, el empleo de técnicas y métodos de recomendación está siendo muy exitoso. Sin embargo, podemos encontrar un número de problemas típicos en los SR actuales [1, 123], entre los que destacan:

- La dificultad al identificar las preferencias del usuario y clasificarle adecuadamente para ofrecer recomendaciones personalizadas.
- La incorporación constantemente de ítems, usuarios y valoraciones. El volumen de datos crece constante, por lo que se llegan a tener problemas de escalabilidad.

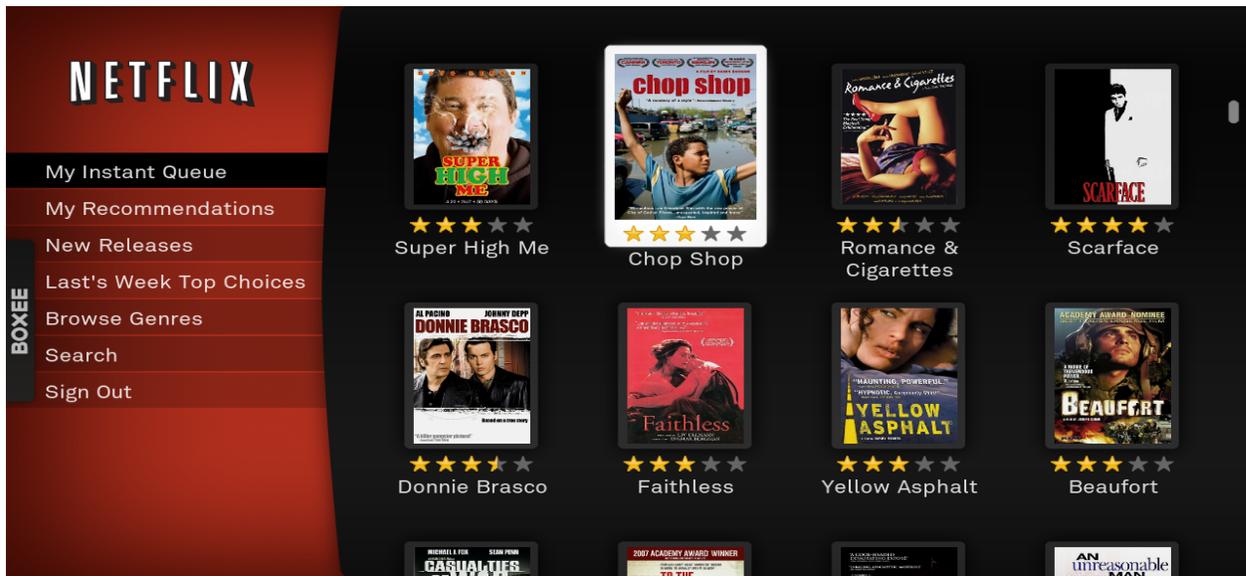


Figura 2.1: SR de películas Netflix.

- La dispersión de los datos, debido a que para la mayoría de los usuarios se tienen disponibles valoraciones sobre algunos de los ítems del sistema.
- La sobre-especialización: el sistema muestra al usuario elementos similares a los que ya ha visto anteriormente.
- Comienzo frío: para el caso de nuevos usuarios y debido a que las sugerencias le son realizadas teniendo en cuenta la comparación entre otros usuarios, de un usuario del que se disponen pocos o ningún dato resulta difícil clasificarlo, y por tanto, recomendarle. Similar situación ocurre para los nuevos productos, de los que en un periodo inicial no se suele disponer de suficientes accesos y valoraciones.

La generación actual de SR aún requiere nuevas mejoras que hagan a los mencionados sistemas más efectivos y aplicables a un rango aún mayor de aplicaciones reales. Estas mejoras incluyen, entre otras, la adición de información contextual durante los procesos de recomendación, la aplicación de estrategias que aborden situaciones con pocas valoraciones de usuarios sobre determinados ítems, la utilización de evaluaciones multicriterio y recomendaciones con mayor flexibilidad e interpretabilidad por parte de los usuarios, así como el estudio de algoritmos más escalables que permitan hacer recomendaciones no sólo para un único usuario, sino también para grupos de usuarios con intereses comunes, siempre asegurando la privacidad y la fiabilidad entre usuarios.

Las maneras de llevarlas a cabo son líneas abiertas de investigación. Se menciona en [26] la posibilidad de añadir meta-información a la descripción de los objetos a recomendar y a las preferencias de los usuarios, junto con la capacidad de inferir conocimiento a partir

de las relaciones existentes en las ontologías de dominio utilizadas. Por ejemplo, un filtrado colaborativo o SR para un campo en concreto puede hacer predicciones acerca de qué debería desear un usuario, dada una lista parcial de gustos de otros usuarios.

2.1.1. Principales modelos de recomendación

Tradicionalmente en los SR existen un número de ítems o productos $I = \{I_1, I_2, \dots, I_m\}$ y además un gran número de usuarios $U = \{U_1, U_2, \dots, U_n\}$. Además por cada usuario, un conjunto de votos, según la calidad de los ítems observados y el criterio que sobre estos el usuario tiene.

Del conjunto de usuarios debemos destacar el usuario activo U_a o A , el cual es el que interactúa con el sistema y al que haremos las recomendaciones en las diferentes aproximaciones en esta memoria. Para identificar un usuario general se suelen emplear las letras U o V .

Similar al usuario, debemos destacar el ítem objetivo, I_t , que es el ítem de supuesto interés del usuario y sobre el que realizamos la predicción. Un ítem cualquier suele ser identificado al emplear I .

En el momento de referirse a los votos dados por el usuario y almacenados en el conjunto de datos, se suele emplear R , siendo $r_{u,i}$ el voto dado por el usuario U al ítem I . Los votos computados por el sistema suelen ser denotados como $\hat{r}_{a,t}$. Finalmente se tiene S como el conjunto de posibles votos o valoraciones, (por ejemplo, $S = \{1\star, \dots, 5\star\}$ o $S = \{agrada, desagrada\}$).

Los SR inicialmente solicitaban explícitamente mediante un formulario los intereses que el usuario consideraba útiles, dicha información era almacenada y posteriormente utilizada para sugerirle determinada información al usuario adecuada a sus intereses. En la actualidad, aunque se utiliza esta técnica en muchas aplicaciones, ya no es necesario que el usuario detalle o especifique explícitamente su interés, pues de forma implícita el sistema registra dicha interacción con el sistema y aprende las preferencias.

De forma general, un SR (Figura 2.2) está formado por tres componentes:

- Base de datos de ítems y usuarios: representa toda información de la que dispone el sistema antes de comenzar el proceso de recomendación.
- Datos de entrada: la información que el usuario suministra explícita o implícitamente, para que comience el proceso de recomendación.
- Algoritmo de recomendación: proceso que combina los datos previos y los de entrada para generar las recomendaciones.

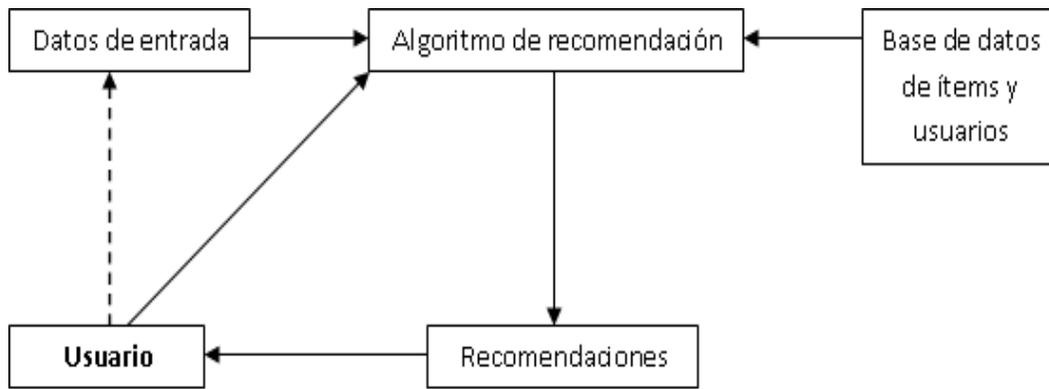


Figura 2.2: Esquema general del proceso de recomendación.

Por ejemplo, de un usuario que interactuó con el sistema previamente, el sistema dispondrá de sus valoraciones previas, que junto con la información del resto de usuarios e ítems, más la información que suministre el usuario para que de inicio el proceso de recomendación, permitirán que el modelo de recomendación suministre las sugerencias oportunas al usuario.

Dentro de las tareas normalmente asociadas a los SR se encuentra la *predicción de voto*, donde para un determinado ítem no visto por el usuario, el sistema es capaz de estimar cómo el usuario lo evaluaría. La segunda tarea es el *encontrar buenos ítems*, en la cual se trata de encontrar los mejores ítems (en algunos casos además los votos) que han de ser recomendados a un determinado usuario. En este caso, los resultados son ordenados considerando la valoración computada.

Estos sistemas funcionan mediante diversos métodos (técnicas), aplicables a uno u otro dominio según la conveniencia de su utilización y las ventajas/desventajas de las mismas; dado que ante una problemática determinada, aplicar una técnica resulta más ventajoso que otra. En situaciones donde se suele tener información muy dinámica, tal vez una adecuada elección sea una que explote las características de los ítems y no las valoraciones de los usuarios. Por ejemplo, en una biblioteca, el potencial que representan los términos (palabras) que componen los libros, da la idea inicial de la utilización de la composición de los ítems. En una subasta, donde los ítems son adicionados y eliminados relativamente rápido, considerar las características en caso de disponerse de ellas, pudiera ser la opción apropiada.

Los SR se suelen clasificar en dos grandes grupos: basados en contenido (filtrado cognitivo) y de filtrado colaborativo (filtrado social) [22, 78, 97]. Entre otras formas de clasificación encontramos:

- basados en grados según la clasificación anterior;
- basado en inspección, son aquellos que solicitan explícitamente información antes de proporcionar cualquier recomendación, siendo el recomendador demográfico el más co-

nocido de este tipo (atributos como el género, sexo, región, carrera y especialidad son utilizados para identificar usuarios en diversas problemáticas);

- basados en actividades, aquellos que se basan fuertemente en las entradas que ofrecen los usuarios, ya sea a través de la navegación o las actividades que realizan en el sistema [59].

De forma resumida, se abordarán las características fundamentales de algunos de los tipos de SR anteriormente citados.

Sistemas de filtrado colaborativo

Los sistemas de filtrado colaborativo, como ya se mencionó, también llamado filtrado social, dependen de un gran número de valoraciones por parte de los usuarios. En la Figura 2.3 mostramos un esquema con las principales tareas en este tipo de sistemas. Recomiendan objetos a un usuario objetivo (activo), basándose en las opiniones de otros usuarios (En la Figura 2.3 tarea de selección de vecinos) con gustos similares [107], lo que significa que la tarea principal de esta clase de SR es predecir la utilidad de los objetos para un usuario particular basándose en una base de datos de valoraciones de usuarios.

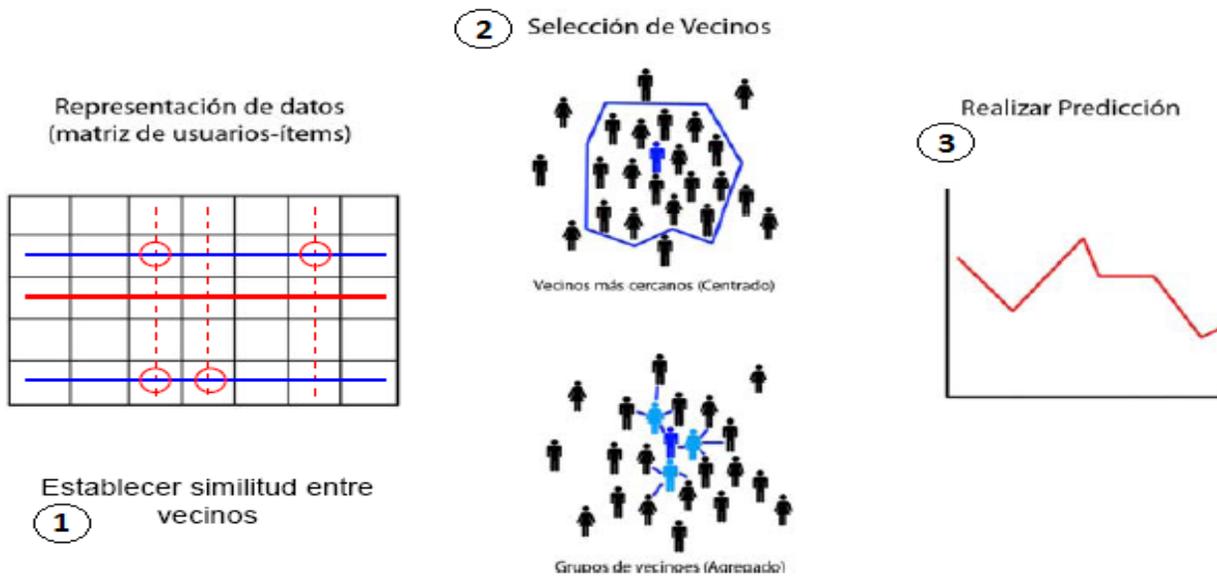


Figura 2.3: Tareas en los sistemas de filtrado colaborativo.

La idea principal es la automatización del proceso del “boca a boca” por el que las personas se recomiendan productos o servicios unos a otros [52, 59, 110]. Si el usuario necesita elegir entre varias opciones en las que no tiene ninguna experiencia, éste probablemente confiará en las opiniones de aquellos que sí tienen esa experiencia. Por ejemplo, al tener que seleccionar

entre varias películas o varios libros, se suele preguntar a un amigo o familiar que haya visto las películas o leído los libros, en busca de una opinión que ayude a tomar una decisión.

Sistemas basados en contenido

Este tipo de sistema es la respuesta más antigua del filtrado. Proporciona recomendaciones basadas en una correlación de “artículo a artículo” [85, 103]; siendo muy “predictivo”, pues se basa en la información del sistema y no del usuario “real”.

De esta forma se recomendarán a los usuarios ítems que guarden una alta relación con los accedidos en el pasado en cuanto a sus características, siendo ésta una de las principales desventajas de esta aproximación. Por ejemplo, en un SR de libros que aplique este tipo de sistema, los usuarios recibirán recomendaciones de aquellos libros que tengan relación de contenido, con los libros accedidos previamente. Mientras que recomendaciones de libros sobre temáticas diferentes a las accedidas, no le serán recomendadas a los usuarios con este enfoque. Otra limitante se encuentra cuando en determinados dominios no es posible a los ítems descomponerlos en sus características o analizarlos por su contenido.

Un caso particular de estos sistemas son los basados en caso [65, 111]. Su utilización se presenta mayoritariamente en el sector del comercio electrónico, donde en función de los productos adquiridos por los usuarios se pueden encontrar otros similares que suelen ser recomendados. Entrée es un ejemplo de este tipo de SR.

Sistemas de filtrado económico

Estos sistemas filtran la información teniendo en cuenta factores de costo, como pueden ser el ancho de banda de los usuarios, el costo de productos adquiridos, el tamaño de ítems visitados u otros rasgos evidenciados en las visitas realizadas previamente por los usuarios [59]. En determinados sistemas se pueden encontrar usuarios que no acceden a determinados recursos pues no tienen tiempo para ello, o porque el ancho de banda que poseen se lo impide. Un Sistema de filtrado económico puede resultar de mucha utilidad en bibliotecas digitales, dado que el tiempo que se dispone es limitado, o bien los tipos de recursos buscados sean específicos. En ocasiones se tienen restricciones según el tipo de ordenador que se dispone y es preferible un tipo de recurso y no otro.

La existencia de un recurso limitado es uno de los requisitos de funcionamiento de este tipo de sistema, mientras que el centrarse en tales limitaciones y no en la necesidad del usuario es una de sus problemáticas en muchas ocasiones.

Sistemas basados en la demografía

En este tipo de sistema se utiliza el conocimiento demográfico que se tiene de los usuarios y sus opiniones sobre los ítems recomendados como base para las recomendaciones [54, 100]. Datos como la profesión, género, sexo, ocupación y otros suelen ser datos demográficos utilizados para caracterizar grupos de usuarios, de los que se asumen que todos sus miembros tendrán preferencias, gustos o puntos de vista similares.

Esta alternativa suele ser utilizada en sectores específicos, donde las recomendaciones por características demográficas cobran sentido, por ejemplo, en la moda, la educación o servicios sociales. Resalta como la principal problemática lo difícil que resulta obtener datos demográficos de los usuarios, ya que estos son contrarios a introducir información personal en los sistemas computacionales.

Sistemas de filtrado basado en reglas

Este enfoque filtra datos e información según un conjunto de reglas que expresa la política de filtrado de la información [59]. Estas reglas pueden ser parte del contenido del perfil del usuario. De forma general las reglas se construyen luego de que el usuario conteste una lista de preguntas que se le muestran, de las cuales se intentan definir los intereses y preferencias del usuario.

Este tipo de sistema presenta dos métodos alternativos de filtrado basados en las reglas: reglas basadas en estereotipo y reglas personales. En el primero de estos métodos se asigna un usuario a un grupo de usuarios similares y éste hereda el perfil de filtrado del grupo; mientras en el segundo de los métodos cada usuario tiene sus propias reglas de filtrado.

Ejemplo de este tipo de sistema se encuentran aplicados a los sistemas de correo electrónico para detectar y eliminar los correos basura (*spam*).

Sistemas basados en el conocimiento

Estos sistemas utilizan el conocimiento anterior acerca de cómo los artículos recomendados pueden satisfacer las necesidades de los usuarios. Como las recomendaciones se basan en el conocimiento anterior, no hay tiempo de aprendizaje antes de hacer buenas recomendaciones. El usuario tiene que decir lo que él desea, entonces el sistema determina, basándose en su base de datos de conocimiento, los elementos disponibles que satisfacen sus requerimientos [20, 22].

Sistemas basados en utilidad

En los sistemas basados en utilidad se realizan sugerencias basándose en el cálculo de la utilidad de cada objeto para el usuario. El principal problema es cómo crear una función de utilidad para cada usuario y que después pueda ser empleada de manera adecuada en

la recomendación. En determinados sectores, dado que se manejan determinados criterios subjetivos, resulta muy complejo utilizar estos sistemas, por ejemplo, los sectores musicales y televisivos.

Sistemas de Recomendación a grupos de usuarios

Tradicionalmente, los SR recomiendan ítems a usuarios individuales y las clasificaciones analizadas anteriormente funcionan de manera general en esa dirección, pero el trabajo colaborativo ha sido una tendencia en los últimos años y con ello el estudio de cómo hacer recomendaciones a grupos de usuarios que laboran de manera síncrona o asíncrona [24, 61, 97].

En [61] se muestran un grupo de aplicaciones de estos tipos de SR en variados contextos. Numerosas son las ocasiones en que múltiples usuarios coinciden o no en el tiempo, realizando determinada búsqueda de información y estos sistemas han probado ser útiles; coincidir en un restaurante, ir de vacaciones o ver una película juntos, son algunos de los ejemplos que se pueden encontrar.

En lo adelante profundizamos un poco más en los enfoques que utilizamos a lo largo de la memoria, filtrado colaborativo y filtrado cognitivo. Se destacan las técnicas más relevantes para la investigación en cada enfoque.

Sistemas de Recomendación híbridos

Los sistemas híbridos combinan dos o más técnicas de recomendación para incrementar el rendimiento general del sistema resultante, atenuando las deficiencias de cada una de las técnicas utilizadas por separado [21, 22]. De forma general, se combina el filtrado colaborativo con otra técnica para evitar el problema de los nuevos ítems o usuarios. Estos sistemas generan un modelo de usuario según el comportamiento de los usuarios.

Algunos métodos híbridos combinan los contenidos y estrategias de colaboración [28, 77, 99]. Se emplean los perfiles de los usuarios basados en el análisis del contenido, para compararlos y determinar los usuarios más similares para la estrategia de colaboración, refinando en cierta forma la calidad de la recomendación. El orden y la forma varían según las estrategias y los objetivos que se persiguen. GroupLens es un ejemplo que implementa un sistema híbrido de Usenet [106] y NewsWeeder es un sistema de noticias, que utiliza igualmente una estrategia de filtrado híbrida [69]. Un problema relacionado a las aplicaciones anteriores es la recomendación de páginas Web [2, 6].

Muchos investigadores proponen la fusión entre otras alternativas para maximizar ventajas y atenuar limitantes, siendo la hibridación entre los sistemas basados en contenido y los de filtrado colaborativo la más estudiada [7].

2.2. Sistemas de recomendación de filtrado colaborativo

El apoyarse diariamente ante disímiles situaciones en opiniones o criterios de personas conocidas, familiares o expertos, fue la base para el surgimiento de los SR de filtrado colaborativo. En general simulan una ayuda grupal ante la incertidumbre que generan las múltiples opciones que nos encontramos en el día a día. Herlocker et al. [52] presentan un análisis de algoritmos existentes de filtrado colaborativo, donde se muestran algunas de sus principales características.

Estos sistemas utilizan diversas técnicas para actualizar las valoraciones realizadas por los usuarios al paso del tiempo, y así explicar el movimiento de los intereses de un usuario, en muchos casos son re-evaluadas, puesto que los gustos y preferencias de las personas no se mantienen inalterables y están influenciadas por muchos factores.

De acuerdo a [19], se pueden dividir en dos categorías principales, dependiendo del algoritmo de cálculo: basado en memoria (o heurística) y basado en modelo/artículo. Los primeros operan sobre toda la base de datos de los usuarios para hacer predicciones; los algoritmos basados en modelo, por el contrario, utilizan la base de datos de los usuarios para estimar o aprender un modelo, el cual entonces utiliza para dar las predicciones. Los algoritmos basados en memoria aprenden rápido pero hacen predicciones lentas, mientras los basados en modelo hacen predicciones rápidas pero aprenden lentamente.

- **Sistemas basados en memoria:** Es la técnica más popular de predicción en las aplicaciones de recomendación colaborativa, también es conocida como el método de los k vecinos más próximos o aproximaciones basadas en vecindad. Se caracteriza por utilizar toda la base de datos de elementos y usuarios para generar predicciones. Primeramente, emplean técnicas estadísticas para encontrar vecinos (usuarios con un historial de valoraciones sobre los elementos similar al usuario actual). Una vez que se ha construido una lista de vecinos se combinan sus preferencias para generar una lista con los N elementos más recomendables para el usuario actual [19, 29, 58, 86].

Entre sus inconvenientes se encuentra la necesidad de disponer de un número mínimo de usuarios con un número mínimo de predicciones cada uno, incluido el usuario para el que se pretende realizar la recomendación.

- **Sistemas basados en modelo:** Utiliza las preferencias de los usuarios para aprender un modelo, el cual entonces utiliza para dar las predicciones. Desarrolla primero un modelo de las valoraciones de los usuarios. Lo asumen como un problema de predicción estadística y calculan el valor esperado para cada ítem en función de las valoraciones anteriores.

El modelo puede construirse *offline* durante varias horas o días según la fuente de datos, resultando muy pequeño, rápido y esencialmente tan exacto como los métodos

basados en memoria, incluso llegan a ser más rápidos que estos últimos, sólo que necesitan para ello de un proceso de aprendizaje intensivo. Suelen ser prácticos en entornos en los que las preferencias de usuario cambian lentamente con respecto al tiempo necesario para construir el modelo, no siendo conveniente su uso para ambientes en los cuales los modelos de preferencias del usuario se actualizan rápidamente o con frecuencia.

Son varias las técnicas que se aplican para concebir este tipo de algoritmo. Por ejemplo, Redes Bayesianas [35], clustering [31, 104, 120] y métodos basados en reglas [73], son los más extendidos y trabajan con éxito en la práctica.

Dentro de las principales limitantes de este tipo de sistema [1, 123] podemos resaltar:

- La recomendación de nuevos artículos o a nuevos usuarios, conocido como “comienzo frío”, situación que ocurre cuando se intenta recomendar un ítem que ha sido valorado por muy pocos usuarios o no ha sido valorado. También se evidencia ante la llegada de un usuario que no ha emitido aún ningún criterio y por ende no tiene colaboradores en el sistema, o en usuarios que no han valorado ítems suficientes.
- La dispersidad de los datos, dada por la diferencia y diversidad en cuanto al número de valuaciones o accesos realizados por los usuarios.
- Continuo incremento del número de usuarios y accesos realizados por estos, lo que tiende a reducir la eficiencia de muchas de las técnicas aplicables.

2.2.1. Etapas del filtrado colaborativo

En el filtrado colaborativo, de una u otra forma, el aporte colectivo es esencial en los pasos de dichos modelos. Diversos son los enfoques seguidos pero todos se centran en seleccionar los usuarios/ítems más similares que puedan predecir mucho mejor el interés del usuario activo.

En los métodos basados en vecindad, lo primero es medir cuán parecidos son todos los usuarios colaboradores con el usuario activo [14, 15, 50]. Posteriormente se seleccionan los usuarios que formarán parte del vecindario y computarán el voto hacía el ítem que se analiza, I_t .

En cuanto a esta selección de vecinos, que es el segundo paso en los métodos basados en vecindad, dado que los sistemas actuales poseen cientos de usuarios, se toman un subconjunto para mejorar la eficiencia y efectividad. Para ello se puede, o bien tomar una cota y seleccionar los que la superen o seleccionar un número fijo siempre de vecinos.

Para seleccionar la vecindad, solo los usuarios que han evaluado el ítem activo generalmente forman parte del vecindario, siendo finalmente seleccionados los que mejor relación guarden con el usuario activo según una medida de similitud, $N_t(a)$. Herlocker et al. en [50]

demonstraron que es mucho mejor establecer un número fijo de vecinos (potencialmente en el rango desde 20 hasta 60 vecinos) que usar un umbral en los pesos de similitud.

Posterior a este paso se normalizan las puntuaciones de los distintos usuarios y se calcula la predicción. Con el objetivo de computar las predicciones, los algoritmos basados en memoria pueden ser clasificados en métodos basados en usuarios [52, 110] y basados en ítems [37, 103]. Los primeros emplean una medida de agregación que considera los votos dados por usuarios similares al mismo ítem.

Los métodos basados en ítems tienen en cuenta la similitud entre ítems (dos ítems son similares si han sido evaluados de forma similar), siendo una buena alternativa a los basados en usuario. El procedimiento consiste en seleccionar los elementos que un usuario determinado votó y después comprobar cuán similar es cada uno al resto de los elementos del sistema, para recomendar los más parecidos. Existen distintas formas de evaluar la similitud entre ítems, pero el procedimiento genérico consiste en tomar dos ítems y calcular su similitud a partir de todos los usuarios que votaron ambos ítems. En teoría es la misma aproximación que la que se tiene con algoritmos basados en usuarios. La ventaja es que en el caso de los elementos la similitud entre ellos es menos variable que la similitud entre usuarios, lo que permite preprocesar estas similitudes y hace el proceso mucho más rápido. Destacan varias aproximaciones, entre las que podemos encontrar: Suma con pesos (*Weighted Sum*), Regresión, Predictor “Slope-one”, Predictor “Slope-one” con pesos, Predictor “Slope-one” Bi-Polar [4, 72, 127].

Dentro de las diversas medidas para determinar la similitud en una u otra aproximación podemos encontrar:

- **Coefficiente de correlación de Pearson (PC):** Esta métrica mide el grado de asociación entre los patrones de votos, obteniéndose valores entre -1 y +1. Un valor positivo es una evidencia de una tendencia general a que los valores de ambos usuarios están correlacionados. De forma general, al emplear esta medida, se seleccionan para formar parte del vecindario aquellos usuarios que se les computen los mayores valores [50]. Se deriva de las fórmulas de regresión lineal, y asume que la relación entre ítems es lineal, los errores independientes y la distribución tiene varianza constante y media cero. Se considera una de las mejores medidas de similitud. El PC se calcula de la siguiente forma:

$$PC(U, V) = \frac{\sum_j (r_{u,j} - \bar{r}_u) \cdot (r_{v,j} - \bar{r}_v)}{\sqrt{\sum_j (r_{u,j} - \bar{r}_u)^2 \cdot \sum_j (r_{v,j} - \bar{r}_v)^2}}, \quad (2.1)$$

donde j representa el número de los ítems en común que ambos usuarios U y V , votaron, es decir, $j \in R_U \cap R_V$. Si los usuarios no tienen ítems votados en común,

entonces la $PC(U, V) = 0$. Además, \bar{r}_u es el voto medio para el usuario U , es decir, $\bar{r}_u = \frac{1}{|R_U|} \cdot \sum_{I_k \in R_U} r_{u,k}$.

- **Medida coseno:** Se considera cada ítem como un vector dentro de un espacio vectorial de m dimensiones y se calcula la similitud como el coseno del ángulo que forman. Esta métrica define la similaridad entre dos usuarios como el coseno del ángulo entre los vectores de votos, con valores entre 0 y 1. Un valor cercano a 1 indica que la similaridad de las valoraciones incrementa. Esta métrica ignora las diferencias en la escala de los votos entre usuarios. De forma similar a la medida **PC**, en el momento de seleccionar los vecinos, se seleccionan los usuarios a los que se les computa un mayor valor en la medida, según la siguiente ecuación:

$$COS(U, V) = \frac{\sum_{i \in R_U \cap R_V} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in R_U} r_{u,i}^2 \sum_{i \in R_V} r_{v,i}^2}}. \quad (2.2)$$

- **Diferencia media cuadrática:** Esta es una medida de distancia (pero el inverso puede ser considerado una métrica de similaridad), la cual evalúa la distancia entre dos usuarios como la diferencia media cuadrática entre los votos dados por el usuario a los mismos ítems.

$$MSD(U, V) = \frac{\sum_{i \in R_U \cap R_V} (r_{u,i} - r_{v,i})^2}{|R_U \cap R_V|}. \quad (2.3)$$

- Otras formas son las medidas de correlación basadas en entropía, correlación de Ringo o correlación de Spearman.

Diversos análisis empíricos han demostrado que al emplear la correlación de Pearson se obtiene los mejores vecindarios [19, 50, 52, 57]. Además, se logra una significativa mejora en la precisión de las predicciones, al considerar un peso asociado al número de ítems que han sido evaluados por cada vecino en el momento de obtener la similitud.

$$sim(U, V) = PC(U, V) \cdot CF, \quad (2.4)$$

con

$$CF = \begin{cases} 1 & \text{Si } k > 50 \\ \frac{k}{50} & \text{sino} \end{cases}$$

donde k es el número de ítems en común.

La idea es penalizar aquellos vecinos que valoran muy pocos ítems y(o) tengan muy pocos ítems en común con el usuario activo. Esta idea viene dada por el razonamiento normal que

tiene una persona en la vida real, donde tiende a confiar más en aquellas personas con las que tienen un mayor número de rasgos o gustos en común.

Huete et al. en [58] presentan una propuesta de filtrado colaborativo basado en memoria, en donde en el paso de selección de los vecinos no emplean las medidas clásicas de correlación lineal, sino que estiman la capacidad de predictiva probabilística de cada usuario respecto a los ítems valorados previamente por el usuario activo. Los autores muestran ejemplos de las deficiencias del empleo de las medidas clásicas y las ventajas de la idea que proponen.

En general, en el momento de computar la predicción, $\hat{r}_{a,t}$, solo se emplean los votos dados a I_t por los mejores vecinos. Una predicción de voto basada en usuarios, puede ser formalizada como una agregación de los votos que los vecinos sugirieron hacía el ítem que se intenta recomendar, $f_A(V, t)$. Estas sugerencias se combinan mediante la ponderación de la contribución de cada vecino por su similitud con respecto al usuario activo:

$$\hat{r}_{a,t} = \frac{\sum_{V \in N_t(a)} \text{sim}(A, V) f_A(V, t)}{\sum_{V \in N_t(a)} \text{sim}(A, V)}. \quad (2.5)$$

En la literatura se pueden encontrar diferentes aproximaciones, las cuales principalmente difieren en la forma en cómo es computada la sugerencia de los vecinos [50, 57]. La selección de una u otra forma depende fundamentalmente de las características de los votos, R . Dentro de las diferentes propuestas, se destacan:

- **Voto bruto.** Esta propuesta considera las sugerencias de los usuarios de forma simple, siendo $f_A(V, t) = r_{v,t}$:

$$\hat{r}_{a,t} = \frac{\sum_{V \in N_t(a)} \text{sim}(A, V) r_{v,t}}{\sum_{V \in N_t(a)} \text{sim}(A, V)}. \quad (2.6)$$

- **Sugerencia centrada en la media.** Se considera que los usuarios pueden usar una escala diferente de votos para cuantificar el mismo nivel de preferencias para un ítem. Por lo tanto, la sugerencia es obtenida al considerar la diferencia entre el voto y el voto medio, siendo $f_A(V, t) = \bar{r}_a + (r_{v,t} - \bar{r}_v)$. Pudiendo obtenerse la predicción [50] de la siguiente forma:

$$\hat{r}_{a,t} = \bar{r}_a + \frac{\sum_{V \in N_t(a)} (r_{v,t} - \bar{r}_v) \cdot \text{sim}(A, V)}{\sum_{V \in N_t(a)} \text{sim}(A, V)}. \quad (2.7)$$

- **Sugerencia basada en la puntuación individual (Z-Score).** Además de la diferencia en la escala de votos analizada previamente, también se considera la variación en las escalas de calificación individual. Aquí, la sugerencia de la vecindad es computada al dividir el voto centrado en la media por la desviación estándar σ , siendo

$f_A(V, t) = \bar{r}_a + \sigma_a(r_{v,t} - \bar{r}_v)/\sigma_v$, obteniéndose finalmente para la predicción la siguiente ecuación:

$$\hat{r}_{a,t} = \bar{r}_a + \frac{\sum_{V \in N_t(a)} (r_{v,t} - \bar{r}_v)/\sigma_v \cdot \text{sim}(A, V)}{\sum_{V \in N_t(a)} \text{sim}(A, V)}. \quad (2.8)$$

Podemos encontrar otras técnicas para realizar las predicciones, como son el encasillamiento, horting, redes bayesianas de creencia y las redes neuronales [108]. En la próxima sección describimos los métodos que resultan más relevantes en nuestra investigación.

2.2.2. Métodos de filtrado colaborativo

Debemos destacar que existen un gran número de algoritmos de filtrado colaborativo que soportan diferentes enfoques para realizar las recomendaciones [12, 16, 25, 74, 77, 82, 101]. En las experimentaciones utilizamos una propuesta muy empleada [50] en las investigaciones de filtrado colaborativo, donde se emplea la PC (Ecuación 2.6) para seleccionar los mejores vecinos y la sugerencia centrada en la media para estimar el voto (Ecuación 2.7).

Otros métodos como ***Slope One*** y **SVD** que describimos en las secciones siguientes, también son empleados en esta investigación.

2.2.2.1. Algoritmo *Slope One*

Slope One es una de las formas más simples de filtrado colaborativo basado en ítems. La idea es estimar las preferencias de nuevos ítems basándose en la diferencia promedio de las votaciones entre un nuevo ítem y los restantes ítems que el usuario voto [71].

Su nombre viene del hecho de que se comienza con la suposición de que hay algún tipo de relación lineal entre las votaciones para un ítem y para otro, relación que es válida para estimar las preferencias de algún ítem Y basado en las preferencias del ítem X , a través de una función lineal como $Y = mX + b$. Entonces, el *Slope One* hace el supuesto adicional de que la pendiente es igual a la unidad ($m = 1$), logrando una simplificación en la función lineal anterior. Donde solo resta encontrar, $b = Y - X$, que representa la diferencia promedio entre las votaciones para cada par de ítems. Esto ha demostrado ser mucho más exacto que la regresión lineal en algunas instancias, y toma la mitad o menos del espacio de almacenamiento.

Este algoritmo es atractivo porque el procesamiento *online* es rápido, al no depender del número de usuarios en el modelo de datos. Depende sólo de la diferencia media de votaciones entre cada par de ítems, que pueden ser precalculadas. Además, la estructura de datos subyacente puede ser eficiente actualizada: cuando cambia la preferencia de un usuario hacia un ítem, se actualizan los valores correspondientes con el resto de los ítems. En contextos en los que las preferencias pueden cambiar rápidamente, esto es una ventaja [4].

2.2.2.2. Técnica de Factorización de Matrices

Las técnicas de factorización de matrices, nos permiten descomponer una matriz en el producto de dos o tres matrices con una estructura especial. Recientemente, estas técnicas han ganado popularidad, gracias a su atractiva escalabilidad y precisión [97].

Entre ellos destaca el *Singular Value Decomposition* (SVD), algoritmo para factorizar una matriz real o compleja. Es una consecuencia directa de un teorema en álgebra lineal: supongamos que tenemos una matriz donde cada columna representa a un usuario, y cada fila representa un producto (ítem). Por lo tanto, con M usuarios y N productos, tendremos una matriz $M \times N$. El teorema define que podemos descomponer una matriz en tres componentes: el primer componente identificado por U representa una matriz ($M \times M$), el segundo componente S , una matriz ($M \times N$), y el tercer componente, V , una matriz con dimensiones ($N \times N$). De forma tal que podemos usar esta descomposición como una aproximación de la matriz original, $M \times N$. Al tomar los primeros k valores propios de la matriz S , podemos obtener una representación comprimida de los datos.

Esta técnica de factorización de matrices, es comúnmente utilizada cuando se dispone de muchos datos y se quiere obtener una aproximación de menor escala. Dada una matriz A de $M \times N$ con rango r , la descomposición $SVD(A)$ es definida como $SVD(A) = U \times S \times V^T$. Donde la matriz S es una matriz diagonal, llamada singular, que tiene solamente r entradas distintas de cero, claves para la lograr la descomposición de las tres matrices. U y V son dos matrices ortogonales.

SVD produce un conjunto de vectores propios no correlacionados. Cada usuario y el ítem están representados por un vector propio correspondiente. El proceso de reducción de dimensionalidad permite que los usuarios que calificaron ítems similares sean asignados dentro del espacio generado por los mismos vectores propios [102].

En los sistemas de comercio electrónico actuales, donde es necesario realizar búsqueda en millones de potenciales vecinos, y además donde los usuarios necesitan recibir recomendaciones oportunas, los investigadores recomiendan en muchos casos la utilización del SVD [18, 64, 105]. Uno de los ejemplos de la eficiencia de este método, lo encontramos en la competencia que duró cerca de 3 años organizada por Netflix Inc., en la cual uno de los equipos ganadores utilizó esta estrategia [9].

2.3. Sistemas de recomendación basados en contenido

Por otro lado, un SR basado en contenido se emplea para sugerir/recomendar a los usuarios nuevos ítems en función de la similitud del contenido de este respecto a los ítems valorados previamente. De forma general, se suelen utilizar etiquetas para describir un ítem determinado. Imaginemos el sector noticioso donde cada noticia está compuesta por un conjunto

de términos, o el sector automovilístico, donde cada modelo de vehículo se describe por una serie de características. Un SR basado en contenido en estos entornos, debería de ser capaz de ofrecer sugerencias a los clientes/usuarios en función de los accesos a noticias previas o bien según las características de los vehículos comprados anteriormente.

Los SR basados en contenido o de filtrado cognitivo, son los sistemas más antiguos en el campo de la recomendación. El surgimiento de internet unido a la gran cantidad de material y texto generado, facilitó su amplio uso sobre mediados de la década de los años 90 del siglo XX.

Estos sistemas suelen construir un modelo o perfil con los intereses de los usuarios basados en las características de aquellos ítems valorados. Por lo tanto, las recomendaciones se realizan enfocándose en los perfiles, los cuales contienen las preferencias del usuario.

En la Figura 2.4 se ilustra un ejemplo de la información básica que se extrae de las relaciones entre ítems y usuarios en un SR, información que luego forma parte de los perfiles de los usuarios [78]. Estos perfiles se comparan con el contenido del resto de los artículos candidatos a ser recomendados o con el resto de los perfiles para encontrar usuarios o ítems similares.

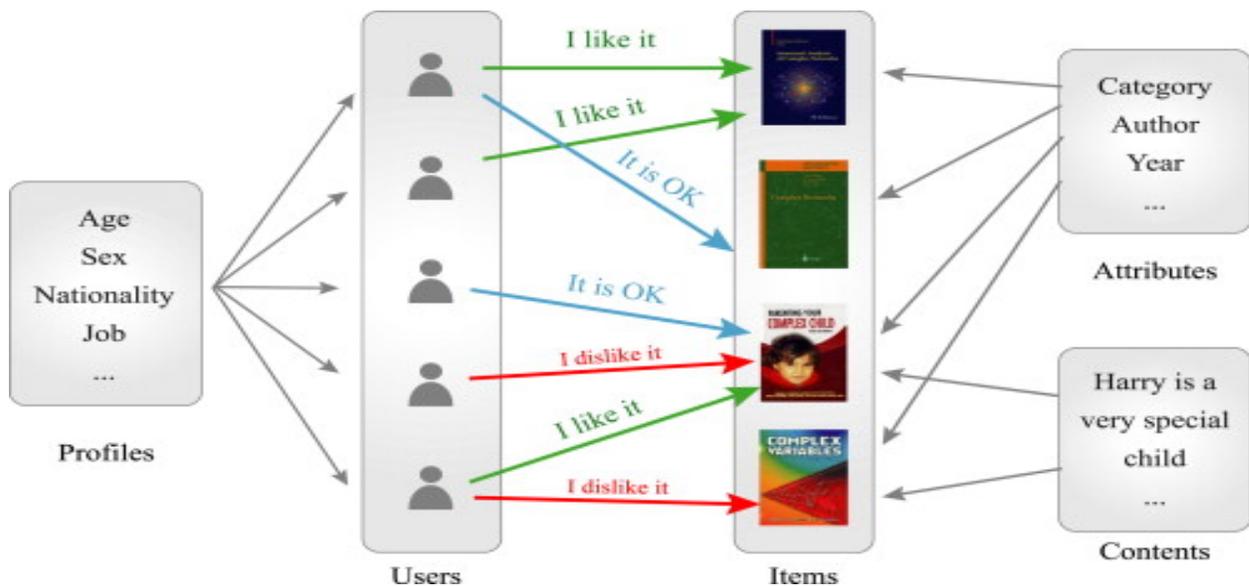


Figura 2.4: Representación de la información contenida en los perfiles de usuarios.

Si un usuario suele escuchar canciones de una determinada cantante, en su perfil se registrará dicho gusto musical, por lo que en el momento de recomendarle al usuario nuevas letras de canciones, la cantante de su preferencia ha de ser una de las que de existir canciones en el listado a recomendar, será escuchada por el usuario. Lo mismo ocurriría en otro entorno donde se maneje información descriptiva de servicios y(o) productos.

Estos sistemas enfocan las recomendaciones en una correlación de “artículo a artículo”, facilitando la recomendación de nuevos productos o servicios al establecer fácilmente una similitud con lo anterior. Es factible realizarles recomendaciones a nuevos usuarios al enfocarse en un perfil definido explícitamente. Sin embargo dentro de las limitantes de estos sistemas se pueden destacar:

- En entornos donde no es posible descomponer los ítems, es imposible su utilización.
- Presenta serios problemas cuando el dominio de los ítems no es textual, por ejemplo, música, películas, etc. De estos dominios y sus ítems resulta muy complejo extraerles su contenido o características completamente, solo es posible manejar parte de su información.
- Tiende a sobre-especializarse: son recomendados los ítems muy valorados con mayor frecuencia que el resto de ítems.
- Nuevos usuarios que acceden al sistema o han valorado muy pocos ítems. Para estos usuarios es difícil realizar oportunas recomendaciones al no disponer un perfil con información suficiente.

En los SR se emplean comúnmente diversas técnicas de minería de datos, entre las que destacan, técnicas de clasificación, agrupamiento y descubrimiento de reglas de asociación. La selección de la técnica apropiada para enfrentar una problemática, depende en gran medida de la información que se disponga, pero otros elementos pueden incidir en dicha elección. Dentro de los SR basados en contenido, podemos encontrar el empleo de algoritmos de agrupamiento.

De forma general, los algoritmos de agrupamiento se enmarcan dentro de los sistemas basados en aprendizaje o clasificación no supervisada, es decir, que no se conoce la clase a la que pertenecen los patrones de entrenamiento; contrariamente al aprendizaje supervisado donde sí que se conoce la relación entre patrones y clases. De hecho, el aprendizaje supervisado se basa en minimizar los errores en la asignación de los patrones a las clases conocidas.

Cuando no se dispone de esta información, una adecuada solución es recurrir al aprendizaje no supervisado, que se basa en determinar la organización de los ítems en grupos o clusters, descubriendo similitudes y diferencias entre ellos, así como extraer conclusiones sobre el problema [114]. Se concibe que los ítems que pertenecen a un mismo cluster serán muy homogéneos entre sí y por otra parte que la heterogeneidad entre los distintos grupos sea apreciable.

En [114] los autores abordan tres grandes categorías donde aplicar las técnicas de agrupamiento. La primera en extracción de datos informativos, donde en múltiples ocasiones, el

número de datos disponibles (N) es muy grande y, como consecuencia, es necesario llevar a cabo un procesamiento para disponer de una cantidad de datos menor pero a la vez, suficientemente informativa.

Las técnicas de agrupamiento también pueden utilizarse para extraer conocimiento, por ejemplo, en forma de reglas a partir de los datos que se disponen. La otra categoría abordada se refiere a la predicción basada en grupos, en la que sobre el conjunto de datos se aplican las técnicas de clustering para determinar características comunes de los patrones que forman cada grupo, y que a su vez los diferencian de los patrones que pertenecen a otros grupos. En diversas aplicaciones prácticas ha quedado demostrada la eficiencia de estas técnicas [17, 49, 67, 104, 120], presentándose como una aceptable respuesta en problemas de alta dimensionalidad.

En la próxima sección se describen los modelos que resultan más relevantes para nuestra investigación.

2.3.1. Algoritmos de *Clustering* o Agrupamiento

Se conoce como *clustering* al proceso por el cual un algoritmo realiza agrupaciones de los datos. Estas agrupaciones son conocidas como *clusters* [63, 113]. El *clustering* tiene un amplio rango de aplicaciones prácticas, desde el procesamiento de imágenes, a las aplicaciones en bolsa, marketing o reconocimiento de patrones.

Contrariamente a la clasificación, donde la etiqueta de cada clase es conocida para cada objeto, en el *clustering* no existen estas etiquetas predefinidas, por lo que el tipo de aprendizaje que se usa es no supervisado. El objetivo fundamental del *clustering* es el de descubrir las relaciones de similitud y disimilitud que existen en un determinado conjunto de datos. Para ello, básicamente, lo que hace es identificar regiones densas y dispersas en el espacio definido por el conjunto de datos.

La técnica de *clustering* particiona los N elementos (x) que se tienen en m conjuntos o *clusters*, de forma que si se tiene X como el conjunto de datos original:

$$X = \{x_1, x_2, \dots, x_N\}. \quad (2.9)$$

C_1, C_2, \dots, C_m serán los clusters obtenidos, quienes deben cumplir, además de que todos los patrones que pertenecen a un cluster (C_i) sean parecidos entre ellos y diferentes de los patrones que pertenecen a otros clusters, las siguientes condiciones:

- $C_i \neq \emptyset, i = 1, \dots, m.$
- $\bigcup_{i=1}^m C_i = X.$
- $C_i \cap C_j = \emptyset, i \neq j; i, j = 1, \dots, m.$

La última de las condiciones da la idea de la pertenencia única de cada elemento a un grupo, esto es llamado *clustering* “duro” (*hard clustering*). Además de esta definición existe el *clustering* “blando”, “difuso” o “suave” (*soft clustering*), donde los elementos pueden pertenecer simultáneamente a más de un grupo con diferentes medidas de pertenencia, entre 0 y 1, pudiendo pertenecer a un *cluster* más que a otro. Como puede comprobarse, el *clustering* duro es un caso particular del difuso, dado que ocurre cuando los elementos tienen una pertenencia total a uno de los *clusters* formados y pertenencia nula al resto.

Dentro del *clustering*, destacan las siguientes técnicas o métodos:

- ***Clustering por partición.*** Es el que de forma general fue descrito anteriormente y donde se necesita conocer a priori el número de grupos a formar. El método de partición separa n objetos en k grupos llamados particiones, donde cada partición representa un *cluster*. Este método asume que (1) $k \leq n$, (2) cada partición debe contener al menos un objeto, y (3) cada objeto debe ser miembro de una o más particiones (en principio, está permitido que un objeto pertenezca a más de una partición con diferentes grados de pertenencia, lo que suele hacerse utilizando técnicas basadas en Lógica Difusa) [63, 113]. Estos métodos suelen crear una partición inicial, que es posteriormente mejorada usando técnicas iterativas. El proceso iterativo mueve los objetos de una partición a otra según se optimice alguna función objetivo. Un ejemplo de técnica de este tipo es el algoritmo de las k -medias [113], o el método EM (*Expectation-Maximization*) [70].
- ***Clustering jerárquico.*** A diferencia de los métodos anteriores, en estos métodos no se requiere conocer a priori el número *clusters*. Los métodos jerárquicos de *clustering* realizan una descomposición jerárquica del conjunto de datos, usando técnicas aglomerativas o divisivas [63, 113]. En estos métodos se va particionando el conjunto de datos por niveles, de modo tal que en cada nivel se unen o se dividen dos grupos del nivel anterior, según el tipo de algoritmo empleado.
- **Métodos basados en “densidad”.** Los métodos basados en densidad emplean la noción física de densidad [40]. El funcionamiento se basa en encontrar *clusters* haciendo crecer un determinado *cluster* inicial siempre que la densidad en los patrones cercanos o vecinos sea mayor que un determinado umbral. Dado un *cluster* c , el umbral de densidad requiere que cada patrón en c contenga un mínimo número de otros patrones en un radio predeterminado. Dentro de estos métodos basados en densidad se podrían situar también los Mapas Topográficos Generativos (*Generative Topographic Mapping*, GTM) [13], que realmente son una reformulación probabilística de los Mapas Autoorganizativos [66].

- **Otros métodos.** Otros métodos son por ejemplo los basados en cuantización, que discretizan el espacio donde se quiere hacer el *clustering* en un número finito de categorías, formando una estructura de cuadrícula y realizando el *clustering* sobre esa estructura “cuadriculada”[46]. También son destacables los métodos basados en modelos, que realizan un modelado de cada *cluster* y encuentran el mejor ajuste del modelo. Se pueden encontrar además los basados en redes neuronales, básicamente Mapas Autoorganizativos [66] y redes basadas en la Teoría de la Resonancia.

Según [113], una labor de agrupamiento involucra la selección de características, la elección de la medida de proximidad, la elección del criterio de agrupamiento, la elección del algoritmo de agrupamiento y la validación e interpretación de resultados.

Existen muchas medidas de similitud y disimilitud que pueden ser usadas como medidas de proximidad entre dos puntos, entre dos distribuciones de puntos, o bien entre un punto y una distribución. Estas medidas están ampliamente referenciadas en la literatura [113], siendo las más usadas las distancias de Manhattan, Euclídea, Mahalanobis y Bhattacharyya como medidas de disimilitud, y el producto escalar y la medida de Tanimoto como medidas de similitud.

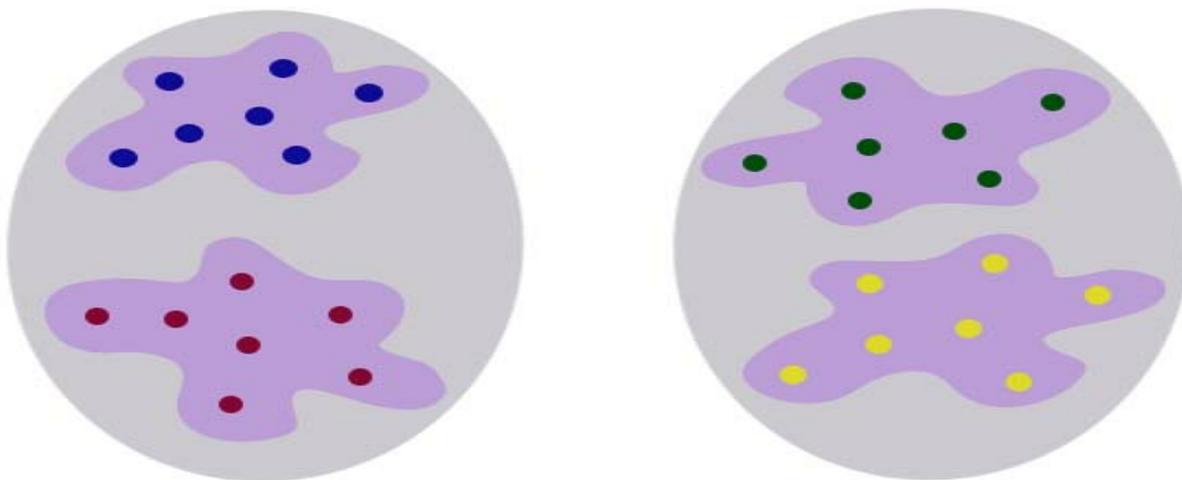


Figura 2.5: Ejemplo donde al aplicar un algoritmo de agrupamiento se obtienen dos o cuatro *clusters*.

Para un mismo conjunto de datos, una elección de diferentes medidas y (o) criterios puede conllevar a obtener diferentes resultados. La Figura 2.5 muestra dos agrupaciones en un mismo conjunto de datos, una puede distinguirse para cuatro grupos y otra sólo para dos. Un agrupamiento se puede observar en los ítems que son contenidos por las dos figuras de

forma regular, en el primer grupo los ítems de color azul y rojo, y en el segundo grupo los de color verde y amarillo. Mientras las figuras de forma irregular representan otro agrupamiento, uno para cada grupo de ítems de color similar. En casos como este, los resultados suelen ser analizados por expertos antes de tomar la decisión de cual agrupamiento es el correcto para la problemática que se aborda, ya que desde diferentes aristas ambos agrupamientos pueden ser válidos.

2.3.1.1. Algoritmo K-Medias

El algoritmo k-medias data del año 1967 y fue propuesto por Mcqueen [80], es muy utilizado por su simplicidad, rapidez y eficiencia, siendo el más conocido dentro de los algoritmos de *clustering* “duro” [60] y posiblemente el más utilizado en general dentro de todos los algoritmos de agrupamiento.

En este algoritmo los ítems que se agrupan pertenecen exclusivamente a uno de los *clusters* o grupos determinados. La asignación de los ítems a los *clusters* se lleva a cabo de acuerdo con el criterio que se haya adoptado.

El algoritmo consta de los siguientes pasos:

- Paso 1: Considerar los k primeros ítems del conjunto de datos como k conglomerados con un único ítem.
- Paso 2: Asignar en el orden del conjunto de datos cada uno de los ítems al centroide más próximo. Después de cada asignación se recalcula el nuevo centroide.
- Paso 3: Después de que todos los ítems hayan sido asignados en el paso anterior, calcular los centroides de los conglomerados obtenidos, y reasignar cada ítem al centroide más cercano.
- Paso 4: Repetir los pasos 2 y 3 hasta que se alcance un determinado criterio de parada.

La idea del primer paso es considerar los k primeros ítems del conjunto de datos como los k centroides iniciales, luego cada uno de los siguientes ítems del conjunto de datos se asigna al centroide más próximo, con la característica de que al efectuar cada asignación se recalculan las coordenadas del nuevo centroide. Una vez asignados todos los ítems, se calculan los centroides para cada uno de los grupos y se reasigna cada ítem al centroide más cercano sin que en este paso se lleve a cabo un nuevo cálculo del centroide para cada asignación. Los dos pasos anteriores se repiten hasta que se verifique un determinado criterio de parada.

El algoritmo es dependiente de los k centroides iniciales seleccionados y del orden en el que se analizan los ítems. Es un algoritmo que converge a las pocas iteraciones, y se suele utilizar como condición de parada la de que pocos puntos cambien de *cluster*.

En la Figura 2.6 se observan agrupaciones que se obtienen al aplicar el algoritmo k-medias para un grupo de usuarios que se tienen. Como se muestra, se pueden obtener seis grupos de usuarios para el valor de k igual seis.

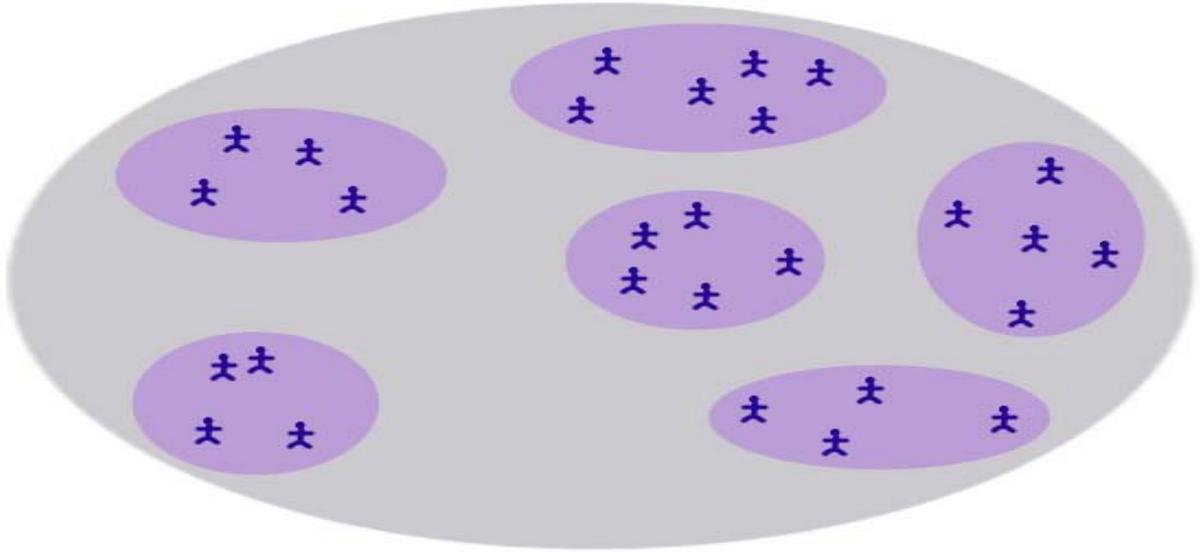


Figura 2.6: Ejemplo al aplicar el algoritmo k-medias para k igual seis.

2.3.1.2. Algoritmo Aspect Model

El modelo que se analiza se puede clasificar dentro del grupo de los algoritmos de *clustering* “difuso”, en el que cada ítem pertenece a un *cluster* en un grado determinado, pudiendo pertenecer a más de un grupo con diferentes grados de pertenencia.

En [55], Hofmann propone una técnica estadística aplicable en sistemas de recuperación y filtrado de información, procesamiento de lenguaje natural y áreas relacionadas. En dicho artículo se enfoca la técnica en el contexto documental (*information retrieval*). El punto de partida de esta técnica es un modelo estadístico que ha sido llamado Aspect Model (*AM*). El mismo es un modelo de una variable oculta para datos concurrentes, los cuales asocian a cada observación una variable oculta z que pertenece al conjunto $Z = \{z_1, z_2, \dots, z_k\}$. En el contexto donde se aplica el modelo se tiene un conjunto de documentos $D = \{d_1, d_2, \dots, d_m\}$ y palabras o términos que los conforman $P = \{p_1, p_2, \dots, p_n\}$, obteniéndose las observaciones $\{d, p\}$, las cuales son asociadas con una de las k variables ocultas definidas.

Mientras otros modelos de agrupamiento ubican cada documento en uno de los grupos formados, este modelo asume que cada documento puede tener múltiples aspectos, y por ende, ha de tener cierto grado de pertenencia a cada grupo en específico de los k que se

decida formar. Al mismo tiempo, las n palabras que están presentes en los documentos tendrán un determinado grado de pertenencia hacia los grupos formados.

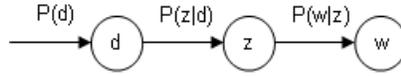


Figura 2.7: Representación gráfica asimétrica de Aspect Model.

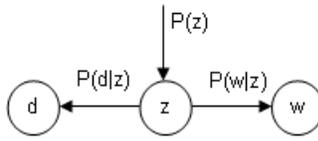


Figura 2.8: Representación gráfica simétrica de Aspect Model.

El entrenamiento de este modelo implica disponer de un vector de términos (p) con dimensión igual al tamaño del vocabulario disponible para cada documento d_i . Este vector representa la importancia de cada término para el documento. La unión de todos los vectores de documentos (d), permite obtener una matriz de entrada para el algoritmo (documentos x términos).

Esta matriz permite estimar (a través del algoritmo EM (*Expectation - Maximization*) [36], el cual es un método iterativo que alterna cada iteración entre dos pasos, E (Ecuación 2.10) y M (Ecuación 2.11)), las tres probabilidades condicionales ($P(U|Z)$, $P(Z)$ y $P(T|Z)$) siguiendo similar protocolo al empleado por Hofmann [55], donde $P(Z)$ sería la probabilidad a priori que se tiene de cada una de las grupos que se decidan formar, mientras, $P(D|Z)$ y $P(P|Z)$ serían las probabilidades de que un documento o palabra pertenezca a uno de los grupos. Las Figuras 2.7 y 2.8 muestran la representación asimétrica y simétrica del modelo.

Paso E:

$$P(z|d, p) = \frac{P(z)P(d|z)P(p|z)}{\sum_{z'} P(z')P(d|z')P(p|z')}. \quad (2.10)$$

Paso M:

$$\begin{aligned}
P(d|z) &= \frac{\sum_p n(d,p)P(z|p,d)}{\sum_{p,d'} n(d',p)P(z|p,d')}, \\
P(p|z) &= \frac{\sum_u n(d,p)P(z|p,d)}{\sum_{p',d} n(d,p')P(z|d,p')}, \\
P(z) &= \frac{1}{R} \sum_{d,p} n(d,p)P(z|d,p), \quad R \equiv \sum_{d,p} n(d,p).
\end{aligned}$$

Con estos valores es posible para cada documento D obtener la probabilidad de relevancia de cada una de las palabras (P), $P(D, P)$.

$$P(D, P) = \sum P(z)P(D|z)P(P|z). \quad (2.11)$$

Se puede tener, por ejemplo, un grupo de documentos que hablen de política, otros de cultura y otros de deporte, por citar algunas clasificaciones. Cada uno de estos documentos según la composición interna que tengan podrán pertenecer en mayor o menor medida a los k grupos que se decidan formar, lo que podría dar como resultado que dos o más documentos con clasificaciones totalmente distintas pudieran tener hacia uno o varios de los grupos anteriores un grado de pertenencia muy similar.

La técnica de minería de datos basada en AM es conocida como PLSA (*Probabilistic Latent Semantic Analysis*) y como ya hemos mencionado, ha sido aplicada en la recuperación de información y aplicaciones de filtrado colaborativo. En este último campo se utiliza para identificar grupos de usuarios con intereses comunes a los que recomendarles ítems a partir de la información que dichos usuarios manifestaron sobre otros ítems accedidos previamente.

Este modelo y otras variantes son detalladas por Hofmann en [56], donde se muestra el buen desempeño que se alcanza con respecto a un conjunto de variantes tradicionales. En la recomendación de películas y en función de los votos que los usuarios dan, se pueden encontrar algunas propuestas desarrolladas que siguen este enfoque. Shein et. al. en [109], muestran el desarrollo de un modelo híbrido para la recomendación de ítems, atacando con ello uno de los problemas de estos sistemas, el comienzo frío. Ellos hacen uso del AM en el contexto de personas-actores-películas, donde se obtienen mejores resultados que con otras aproximaciones sobre la misma problemática.

2.3.1.3. Extensión de Aspect Model

El modelo que fue descrito previamente, AM , en el campo de la recomendación se considera un modelo puro de filtrado colaborativo, al tenerse en cuenta solamente usuarios y ítems. Popescul et. al. extienden [87] el modelo propuesto por Hofmann (1999) en [55], donde

muestran una propuesta híbrida de un SR, incorporando en su propuesta además de los datos referentes a usuarios y ítems, el contenido de estos últimos.

En la propuesta, los autores consideran la situación donde un usuario, U , puede seguir un tema o tópico de interés, el cual a su vez genera tanto los documentos, D como el contenido de estos, P . Los usuarios, documentos y contenido son considerados independientes dados los tópicos de interés.

Sea $n(U, D, P)$ el número de veces que el usuario U lee el término P en los documentos D . Teniendo los datos del conjunto de entrenamiento de la forma anterior, el logaritmo (\log) de la probabilidad, L , es computado según la ecuación:

$$L = \sum_{u,d,p} n(u, d, p) \log P(u, d, p). \quad (2.12)$$

En el modelo las probabilidades de distribución son estimadas empleando el algoritmo EM (Ecuaciones 2.13 y 2.14). Este algoritmo se ejecuta hasta que se alcanza el máximo local del logaritmo de la probabilidad.

Paso E:

$$P(z|u, d, p) = \frac{P(z)P(u|z)P(d|z)P(p|z)}{\sum_{z'} P(z')P(u|z')P(d|z')P(p|z')}. \quad (2.13)$$

Paso M:

$$\begin{aligned} P(u|z) &\propto \sum_{n,c} n(u, d, p)P(z|u, d, p), \\ P(d|z) &\propto \sum_{u,c} n(u, d, p)P(z|u, d, p), \\ P(p|z) &\propto \sum_{u,n} n(u, d, p)P(z|u, d, p), \\ P(z) &\propto \sum_{u,n,c} n(u, d, p)P(z|u, d, p). \end{aligned} \quad (2.14)$$

2.3.2. Otros algoritmos

En este enfoque de recomendación por contenido, suelen emplearse medidas para establecer la semejanza entre documentos, por ejemplo, la medida coseno y la distancia Euclidiana. Estas propuestas miden la relevancia de cada documento con un tema en particular y realizan las recomendaciones según el valor de dicha relevancia.

Un documento dispone de una cantidad finita de palabras, t . Siendo posible representarlo como un vector de t dimensiones. Cada dimensión representa a la palabra o término, y tiene asociado un peso, normalmente la frecuencia relativa en el documento:

$$D_i = (d_{i1}, d_{i2}, \dots, d_{it}),$$

donde d_{ij} representa el peso del j - ésimo término en el documento D_i .

Considerando esta representación por los documentos de una colección, es posible estimar la relevancia de un documento para una determinada temática o medir la semejanza entre documentos. En este último caso se mide la semejanza entre documentos dado su contenido. Sistemas que aplican este enfoque, consideran los documentos valorados previamente por el usuario para determinar aquellos que el usuario no ha valorado y son más semejantes.

En el caso de dos vectores que representan documentos, Da y Db , la distancia euclidiana se calcula de la siguiente forma:

$$f(Da, Db) = \sqrt{\sum_{j=1}^t (Da_j - Db_j)^2}. \quad (2.15)$$

Esta función se utiliza para medir la distancia entre dos puntos en el espacio multidimensional. La distancia euclidiana mínima entre dos puntos es cero, y cuanto más se aleje de cero, más diferentes serán los documentos que los vectores representan.

La medida coseno, por su parte, es una función trigonométrica utilizada para medir el ángulo entre dos vectores en el espacio vectorial. El ángulo que se forma entre dichos vectores indica que tan cercano está uno del otro, indicando la similitud entre los documentos. Esta medida varía entre cero y uno, siendo cero la mayor diferencia entre los vectores y uno que no hay diferencia entre ellos en el espacio dimensional.

$$f(Da, Db) = \frac{\sum_{j=1}^t Da_j * Db_j}{\sqrt{\sum_{j=1}^t Da_j^2 * \sum_{j=1}^t Db_j^2}}. \quad (2.16)$$

Los modelos analizados en esta sección sirven de base para los modelos utilizados en el capítulo 6. En dicho capítulo se puntualizan los elementos que permiten comprender su empleo en el contexto seleccionado.

2.4. Explicación de las recomendaciones

A pesar de la popularidad y el constante crecimiento de los SR, motivos de seguridad, privacidad o simplemente desconfianza, hacen que los usuarios no se sientan seguros de usarlos y(o) confiar en las recomendaciones recibidas. Estos sistemas suelen ser vistos como cajas negras en la que no hay otra opción que confiar en sus recomendaciones al usarlos [51]. Su

aceptación en diversos dominios no ha sido muy amplia, dado que los usuarios al desconocer su funcionamiento, limitan su utilización.

Los años recientes reportan un gran número de investigaciones destinadas a paliar dichas limitaciones. Las respuestas académicas se enfocan en mostrar razones por las que se obtienen las recomendaciones o bien fundamentan las mismas. Una forma de incrementar la aceptación de dichos sistemas, ha sido dotándolos de mecanismos de explicación de las sugerencias ofrecidas [39, 42, 81, 84, 90, 112, 125, 126].

La explicación en los SR cuenta con múltiples objetivos y va desde lograr inspirar confianza al usuario hasta ayudarlo a tomar una adecuada decisión. Todo depende de lo que se persiga. En [115, 116, 117] encontramos interesantes estudios acerca de los objetivos potenciales de la explicación en aras de facilitar el uso de los SR:

- **Transparencia:** explicar cómo funciona el SR.
- **Escrutabilidad:** permitir al usuario decir si el SR funciona bien o mal.
- **Confianza:** incrementar la confianza de los usuarios en el SR.
- **Eficacia:** ayudar a los usuarios a tomar buenas decisiones.
- **Persuasión:** convencer a los usuarios para que prueben o compren los ítems.
- **Eficiencia:** ayudar a los usuarios a tomar decisiones rápidas.
- **Satisfacción:** incrementar la facilidad de uso o disfrute del SR.

De forma general se atribuye a la explicación el concepto de que los sistemas sean transparentes. El ser transparente en los mecanismos empleados y mostrar las razones de las recomendaciones, encuentra precisamente su origen en los sistemas expertos [68], enfocado más que todo en la explicación de las conclusiones arribadas.

Cuando se da una explicación, se intenta ser transparente, exponiendo las razones y los datos considerados en la recomendación. En muchas propuestas resulta complejo lograr mediante la explicación una adecuada transparencia. Lo anterior resulta más crítico cuando nos enfocamos en el campo de aplicación de los SR. En entornos de entretenimiento, el aceptar una recomendación que no nos agrada finalmente, no es tan arriesgado como en dominios que involucren dinero y tiempo. Por ejemplo, un usuario se lo pensaría en aceptar una recomendación para adquirir un producto costoso, sin embargo, si se le muestran razones que lo persuadan, llegaría a confiar en el sistema y muy posible vuelva a emplearlo.

Podemos encontrar una variedad de formas para mostrar las explicaciones de las recomendaciones, las que de forma general se apoyan en los algoritmos que sustentan los SR. Estas formas persiguen mostrar cómo de relevante es el ítem recomendado. En [116, 117] destacan

la presentación y justificación: del mejor ítem, los n mejores ítems en un listado y los ítems más similares.

Entre las soluciones propuestas podemos encontrar interfaces que describen el comportamiento del vecindario en sistemas colaborativos, es decir, argumentan la recomendación apoyándose en el criterio que tuvieron los usuarios más similares respecto al ítem recomendado. Otras propuestas justifican su sugerencia en los ítems accedidos previamente y su relación semántica con el que se recomienda. Esta opción es común en sistemas basados en contenido.

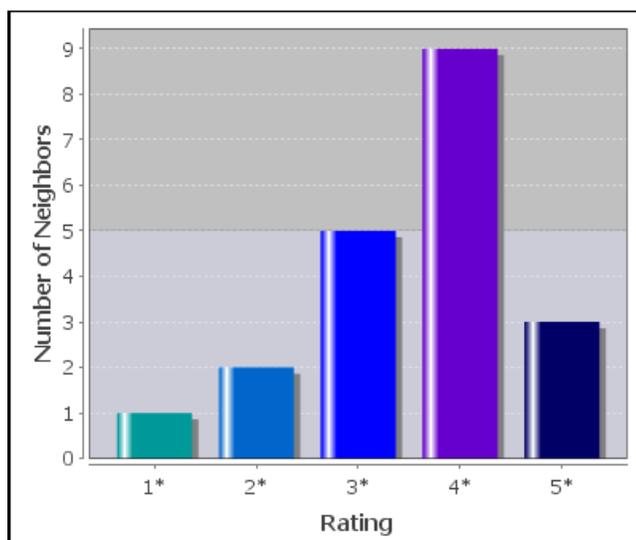


Figura 2.9: Interfaz de explicación. Histograma con los votos del vecindario.

La Figura 2.9 muestra una interfaz clásica (histograma en gráfico de barras) para explicar la predicción de un voto para un ítem específico [53]. Como hemos mencionado, en el proceso de predicción cada candidato sugiere un voto $f_A(V, t)$, el cual se combina para obtener la predicción final. Estas sugerencias son mostradas en el gráfico anterior en las posibles categorías.

De forma general las explicaciones intentan apoyar a los usuarios en momentos de duda: es cómo ir a una tienda o a una biblioteca y aunque tengamos o no alguna idea de lo que quisiéramos, nos auxiliáramos del encargado para tomar una decisión definitiva. Esta persona, por su experiencia y la retroalimentación que tiene del entorno, nos informará oportunamente para tomar la decisión final.

En la vida diaria son muchas las decisiones que se tienen que tomar y, sin duda, se necesitan argumentos para elegir la adecuada. Si unos amigos con los que compartimos gustos nos sugieren una película de dos horas de duración, al verla independientemente de si al final nos gusta o no, solo invertiríamos ese tiempo. Sin embargo, al recibir recomendaciones que impliquen un tiempo mayor o gastos económicos (por ejemplo, un destino turístico, un boleto

aéreo o un equipo electrónico), los recursos involucrados son mayores y si finalmente no nos agrada, notaremos la pérdida en un mayor grado. Estos ejemplos nos confirman la idea de lo importante que pueden resultar las explicaciones para los usuarios.

Los SR que emplean mecanismos de explicación en sus interfaces logran que los usuarios confíen más en las recomendaciones recibidas [90]. Los clientes, al estar satisfechos, se encuentran más persuadidos en el momento de comprar y normalmente vuelven a utilizar la aplicación [116].

En [42] y [84] encontramos taxonomías para clasificar las investigaciones en esta área. En la primera de ellas, Friedrich y Zanker se basan en tres dimensiones: el tipo de razonamiento, el paradigma y las categorías que utiliza cada propuesta. Esta propuesta se ejemplifica con varios de los principales trabajos que se pueden encontrar en el área. Por su parte, Papadimitriou et al. en su propuesta [84] se basan en tres estilos posibles del algoritmo empleado y la combinación de estos: basado en usuarios similares, ítems valorados similarmente e ítems con características similares.

Dentro de los trabajos mencionados en el momento de ejemplificar las taxonomías anteriores, ya sea en aplicaciones comerciales o académicas, encontramos diversas formas de explicación. En [117] se muestran diferentes estilos agrupados dentro de algoritmos de recomendaciones; basado en casos, colaborativos, basado en contenido, demográficos, conversacionales y basados en conocimiento/utilidad. Cada uno de los mencionados estilos se argumenta y explica con un gran número de ejemplos:

- En los basados en casos encontramos, por ejemplo, LibraryThing.com y Netflix.com, propuestas que emplean mecanismos para mostrar ítems muy similares a los consultados previamente.

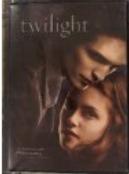
En [79] presentan una propuesta de explicación basada en la recomendación de casos. La idea de los autores es explicar las recomendaciones ofrecidas de forma que se digan las oportunidades que tiene el cliente aunque no se satisfagan todas sus restricciones, en función de las características de los elementos, en lugar de informarle la no existencia de productos con las características deseadas. La propuesta está soportada sobre un SR convencional de búsqueda de productos.

Otros estudios de explicaciones con este enfoque los podemos encontrar en [39] y [81]. En [39] se muestra una revisión de la explicación en sentido general y en especial en el razonamiento basado en casos. El estudio incluye sistemas expertos que contemplan la explicación dentro de sus mecanismos. Mientras en [81] encontramos una breve revisión de propuestas que abordan la explicación dentro de los SR. En el trabajo se presenta una solución basada en la recomendación de los mejores casos (*Top Case*) y la posterior explicación al usuario según las necesidades de justificación que este requiera.

- En los colaborativos contamos con Amazon y LIBRA, en los que ya sea a través de histogramas, palabras claves u otra forma, se describe el comportamiento de los usuarios que forman la vecindad, dándole al usuario la idea de cómo sus vecinos valoraron el ítem recomendado.
- Los SR Movieexplain, OkCupid.Com y Pandora.com, muestran un estilo basado en contenido, acá podemos incluir también, Amazon.com y MovieLens, sistemas con enfoque híbrido que aprovechan todas las ventajas de sus algoritmos. Amazon es un ejemplo de sistema en donde se aplica más de un estilo: nos podemos encontrar con explicaciones soportadas en similitud por contenido de ítems accedidos previamente (Figura 2.10), donde suelen mostrarse un listado de etiquetas para argumentar la recomendación.

Recommendations Based on Your New Likes Return to the Betterizer

Page 1 of 15

 	 	 	 
<p>Twilight > Kristen Stewart, Robert Pattinson, ... ★★★★★ (1,634) \$7.99</p>	<p>Harry Potter and the Goblet of Fire > Daniel Radcliffe, Emma Watson, ... ★★★★★ (782) \$6.99</p>	<p>Twilight > Robert Pattinson, Kristen Stewart, ... ★★★★★ (16) \$10.99</p>	<p>Twilight > Kristen Stewart, Robert Patinson ★★★★★ (53)</p>
<p>Recommended because you said you liked:</p> <p>The Twilight Saga > Kristen Stewart, Robert Pattinson, ...</p> <p>The Twilight Saga > Kristen Stewart, Robert Pattinson</p>	<p>Recommended because you said you liked:</p> <p>Harry Potter and the Order of the Pho... > Daniel Radcliffe, Emma Watson, ...</p> <p>Harry Potter and the Half-Blood Prince > Daniel Radcliffe, Rupert Grint, Emma...</p>	<p>Recommended because you said you liked:</p> <p>The Twilight Saga > Kristen Stewart, Robert Pattinson, ...</p> <p>Twilight in Forks > Various, Jason Brown</p>	<p>Recommended because you said you liked:</p> <p>The Twilight Saga > Kristen Stewart, Robert Pattinson, ...</p> <p>The Twilight Saga > Kristen Stewart, Robert Pattinson</p>

> [See all of your recommendations](#)

Figura 2.10: Interfaz de explicación en Amazon, basada en contenido de ítems valorados previamente.

Billsus y Pazzani en [11] recomiendan noticias a los usuarios y emplean cómo estilo de explicación el basado en palabras claves que forman parte de las noticias actuales y las visitadas previamente. Por ejemplo, “La historia es recomendada, porque contiene las palabras p1, p2 y p3”. Emplean además otros dos estilos, apoyados en la relevancia que reciben las noticias al clasificarlas respecto al interés reciente mostrado por cada usuario.

En [23], Byrne presenta un framework para identificar y explicar nuevas noticias. El prototipo ofrece explicaciones completamente textuales soportadas en el modelo empleado. Se tienen en cuenta los eventos que han ocurrido previamente, relacionados con la noticia y además el momento en que estos ocurrieron en el tiempo.

- Los basados en conocimiento/utilidad resultan muy útiles en aplicaciones muy comerciales dedicadas a la oferta/venta de productos y catálogos de los mismos. Los SR Qwikshop, SAY, myCameraAdvisor, Top Case y Sitios Proveedores de Internet, son ejemplos de este estilo, en donde dichos sistemas, ya sea por mediación de la información recolectada en el perfil, o mediante filtros de opciones disponibles, muestra y argumenta al usuario información filtrada en función de sus requerimientos.

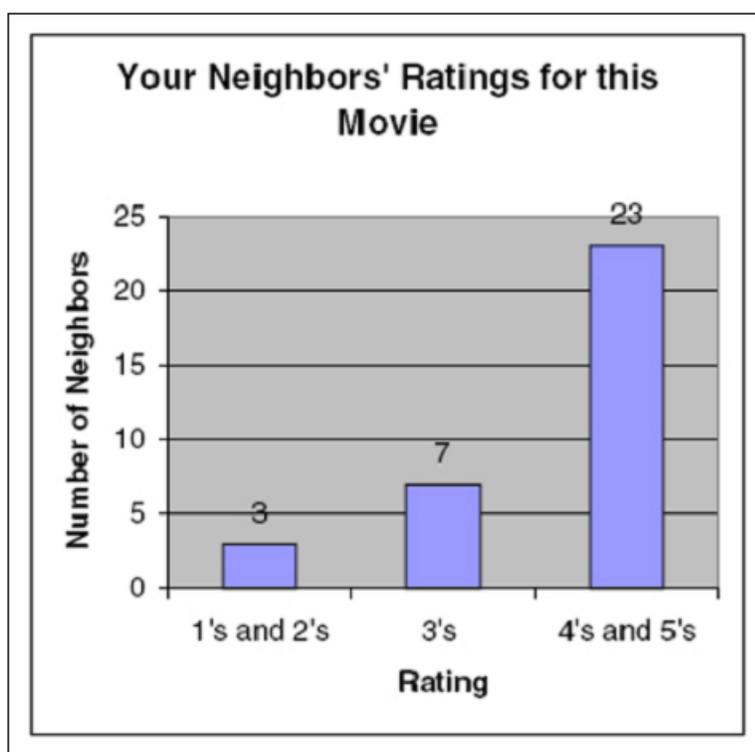


Figura 2.11: Histograma que agrupa los votos de los vecinos.

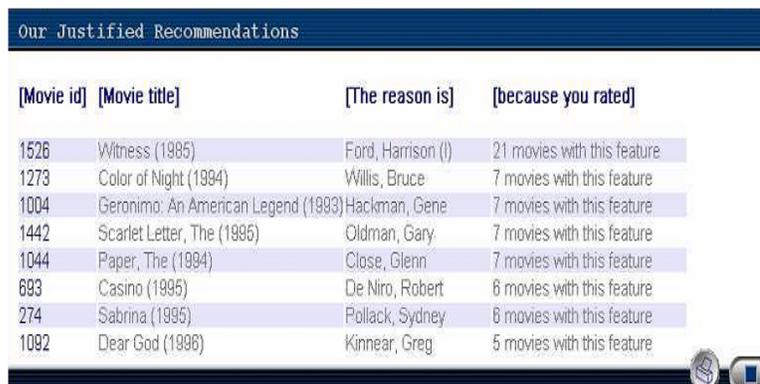
Hetlocker et al. en [53] muestran un estudio con MovieLens, donde describen cómo la explicación de las recomendaciones puede persuadir a los usuarios a tener mayor confianza en el sistema. Los autores explican como pueden diseñarse explicaciones para cada posible etapa. Respecto al criterio de los usuarios, el estudio desarrollado reveló que de las 21 interfaces utilizadas, la interfaz donde se muestra un histograma con los votos agrupados de los vecinos resultó la más efectiva (véase Figura 2.11). En los siguientes ejemplos mostramos algunas explicaciones utilizadas que describen el comportamiento del vecindario:

- Los 20 usuarios más similares, evaluaron la película “X” de la siguiente forma: 1 con una 1★, 3 con 2★, 3 con 3★, 9 con 4★ y 4 con 5★. Esta propuesta está basada en las valoraciones de los vecinos más similares (véase la Tabla 2.1).
- De tus 20 vecinos, 8 evaluaron la película “X” con el mismo voto que tú le diste. Esta propuesta se basa en las películas evaluadas en el pasado con valoración similar a la dada por los vecinos.

Tabla 2.1: Tus vecinos valoraron la película “X”.

Votos	Número de vecinos
★	1
★★	3
★★★	3
★★★★	9
★★★★★	4

En la Figura 2.12 encontramos otro ejemplo de interfaz de explicación muy empleado [112]. La explicación se fundamenta en los actores que intervienen en las películas vistas previamente.



[Movie id]	[Movie title]	[The reason is]	[because you rated]
1526	Witness (1985)	Ford, Harrison (I)	21 movies with this feature
1273	Color of Night (1994)	Willis, Bruce	7 movies with this feature
1004	Geronimo: An American Legend (1993)	Hackman, Gene	7 movies with this feature
1442	Scarlet Letter, The (1995)	Oldman, Gary	7 movies with this feature
1044	Paper, The (1994)	Close, Glenn	7 movies with this feature
693	Casino (1995)	De Niro, Robert	8 movies with this feature
274	Sabrina (1995)	Pollack, Sydney	8 movies with this feature
1092	Dear God (1996)	Kinnear, Greg	5 movies with this feature

Figura 2.12: MoviExplain, interfaz de explicación basada en autores presentes en películas vistas previamente.

Chen y Pu in [27, 89, 90] abordan la construcción de confianza a través del correcto diseño de interfaces para proveer explicaciones. Los autores se enfocan en que dichas interfaces se auto expliquen ellas mismas.

En Newsmap [8], los autores muestran un sistema en el que se muestran las noticias en un mapa (Figura 2.13); cada noticia tiene una simple forma rectangular, un color, una determina intensidad de brillo y elementos tipográficos específicos. Todos estos elementos se combinan

para mostrar la noticia según su importancia en el tiempo u otros elementos. Cada uno de los elementos anteriores varían, encontrándose noticias con un mayor/menor tamaño o brillo, lo cual denota mayor/menor trascendencia e importancia para que el usuario la tenga en cuenta.



Figura 2.13: Newsmap, interfaz de explicación de un sistema de recomendación noticioso.

2.4.1. Valoración de las explicaciones

Hasta aquí hemos visto un número de investigaciones que muestran diferentes estilos de explicación, de forma general enfocadas en el algoritmo empleado para realizar las recomendaciones. La mayoría de las propuestas emplean un solo enfoque o recurso, por ejemplo, las basadas en el voto del vecindario donde con un enfoque colaborativo solo muestran las valoraciones de estos usuarios similares (véanse Figuras 2.9 y 2.11), o las basadas en contenido como la presentada en la Figura 2.10, en la que se explica, apoyándose del contenido disponible de las valoraciones previas.

Muchas de las propuestas analizadas, suelen ser simples e intuitivas, pero no suficientemente informativas para que los usuarios tomen decisiones. Consideramos que en determinadas ocasiones es útil disponer de otras informaciones, por ejemplo, al tomar decisiones en acciones que implican determinados gastos. Un usuario cuando va a comprar una cámara fotográfica profesional o un nuevo coche, necesita no solo una simple recomendación de sus amigos, sino opiniones diversas de expertos que le permitan seleccionar entre las opciones y tomar la decisión que realmente le es oportuna.

Nuestra opinión es que un usuario creará más en las recomendaciones que reciba, cuanto más sepa sobre quienes o porqué le recomiendan. En situaciones donde es necesario invertir, ya

sea dinero, esfuerzo o tiempo, un usuario siempre solicitará argumentos, de los que dependerá el aceptar o rechazar una recomendación. Mientras en otros dominios, como por ejemplo, los de ocio, el tener simples explicaciones tal vez sea suficiente para los usuarios.

2.5. Evaluación de los SR

Según la RAE evaluar es “señalar el valor de una cosa, o estimar, apreciar, calcular el valor de una cosa”. En la inteligencia artificial, una vez se obtiene un modelo, se procede a su validación de diferentes formas. Para ello se utilizan diferentes métricas y colecciones de datos asociadas a un problema en específico, que permiten evaluar el rendimiento del modelo, la satisfacción de usuarios, etc.

En un SR, la evaluación permite:

- Comprobar que el sistema cumple con el objetivo para que el que fue diseñado.
- Detectar las deficiencias y corregirlas.
- Comparar con otros sistemas o algoritmos.

En las siguientes secciones describimos el conjunto de datos que utilizamos en la primera parte de la memoria, y las principales métricas que se emplean en toda la investigación.

2.5.1. Colecciones de datos

Un conjunto de datos o *dataset*, es una colección de datos normalmente tabulada, donde por cada elemento (individuo o instancia) se indican varias características (atributos). La publicación de los conjuntos de datos utilizados en un experimento son clave para su reproducibilidad, y cada vez son más las leyes públicas y normas de revistas científicas que obligan a hacerlos públicos para evitar sesgos en las publicaciones.

Podemos encontrar conjuntos de datos muy utilizados en las investigaciones de SR. Destacan aquellos que caracterizan el mercado del comercio electrónico y el ocio, por la especial atención que despierta en los internautas, por ejemplo, IMDB, Entree, Book Crossing, Netflix, Yahoo Musics, Jester y MovieLens.

El conjunto de datos MovieLens [83], es uno de los más utilizados en el campo de la recomendación y uno de los que empleamos en las experimentaciones. Contiene información coleccionada por miembros del Proyecto de Investigación GroupLens de la Universidad de Minnesota a través de la página Web <http://movielens.umn.edu>.

En esta investigación, la información que manejamos hace referencia a 100.000 votos anónimos en una escala de 1★ a 5★, siendo 1★ la valoración más negativa y 5★ lo contrario. El total de valoraciones se realizó sobre un total de 1.682 películas por parte de 943 usuarios,

para una media de 106 películas aproximadamente por usuario. Cada uno de los usuarios emitió su criterio sobre al menos 20 películas.

Las valoraciones que se tienen, fueron realizadas por los usuarios durante un periodo de siete meses, desde el 19 septiembre de 1997 hasta el 22 de abril de 1998. De los usuarios se conocen algunos datos demográficos, entre los que destacan la edad, sexo, ocupación y código postal de su ubicación. De cada una de las películas se conoce el título, fecha de estreno y los géneros en que se puede clasificar. Se cuenta con un total de 18 géneros diferentes en los que se clasifican las película (acción, aventura, drama, etc.).

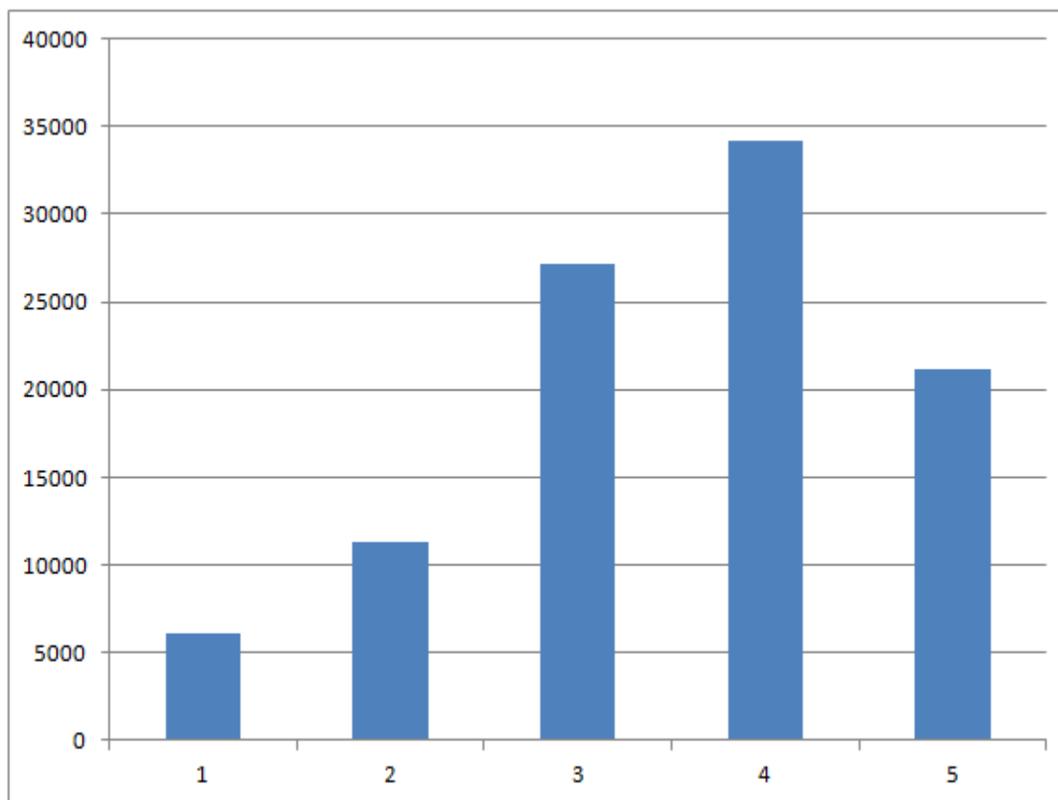


Figura 2.14: Distribución de votos en el conjunto de datos MovieLens.

En el histograma de la Figura 2.14 podemos ver la distribución de votos en el conjunto de datos, mientras en la Figura 2.15 observamos como se comporta la popularidad de los ítems.

2.5.2. Métricas de evaluación

Según Ghani y Gunawardana [47], la decisión sobre qué métrica de evaluación es la apropiada es una situación crítica, ya que cada medida puede favorecer un determinado algoritmo.

Cuando evaluamos un SR, comprobamos generalmente una determinada tarea que el sistema debe realizar, por ejemplo:

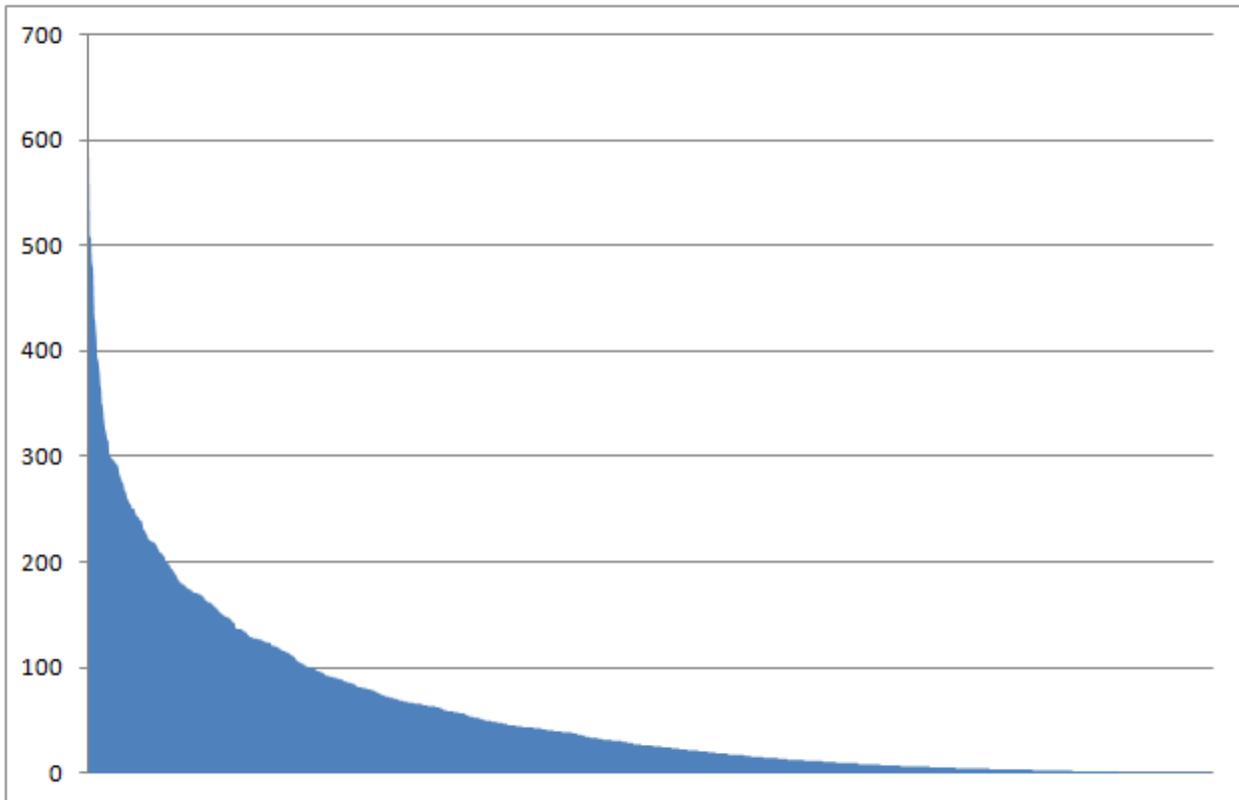


Figura 2.15: Popularidad de los ítems en el conjunto de datos MovieLens.

- Mostrar todos los ítems encontrados.
- Predecir el voto.
- Mostrar una lista ordenado de ítems.

Para cada una de estas tareas podemos encontrar un gran número de medidas. En la investigación comprobamos la precisión de las propuestas que se presentan en los diferentes capítulos.

La precisión evalúa la capacidad del sistema de mostrar el ítem esperado en una lista y(o) la valoración. Pero, ¿qué ítems realmente son relevantes?, pues bien, son relevantes los que:

- i** aparecen en la lista y están en el conjunto de prueba,
- ii** los que el usuario considera útiles y toma la recomendación,
- iii** y los que aparecen dentro de las primeras n posiciones de la lista y están en el conjunto de prueba.

En la tarea asociada a predecir un voto, utilizamos las métricas Hamming, MAE y RMSE, las que básicamente difieren en el costo asociado a cada error. Así, la primera de las medidas

computa el número de predicciones obtenidas correctamente del conjunto de datos de prueba (N) [5, 62], considerando la predicción exacta media respecto al voto original de la siguiente forma:

$$Hamming = \frac{1}{|N|} \sum_{(u,i) \in N} d(\hat{r}_{u,i}, r_{u,i}), \quad (2.17)$$

$$\text{donde } d(\hat{r}_{u,i}, r_{u,i}) = \begin{cases} 0 & \text{Si } \hat{r}_{u,i} = r_{u,i} \\ 1 & \text{en caso contrario} \end{cases}$$

Mientras la MAE considera el error absoluto medio, y la RMSE la raíz del error cuadrático medio [54, 97], de la siguiente forma:

$$MAE = \sqrt{\frac{1}{|N|} \sum_{(u,i) \in N} (|\hat{r}_{u,i} - r_{u,i}|)}, \quad (2.18)$$

$$RMSE = \sqrt{\frac{1}{|N|} \sum_{(u,i) \in N} (\hat{r}_{u,i} - r_{u,i})^2}. \quad (2.19)$$

Está última métrica, RMSE, al elevar al cuadrado los errores, da un peso relativamente alto a los grandes errores que se cometen. Resulta una métrica muy útil cuando los grandes errores son particularmente no deseados. Con el fin de obtener el error en la misma magnitud de los votos, se calcula la raíz de la pérdida esperada.

Para la tarea de mostrar los mejores ítems en un listado ordenado (*ranking*), utilizamos métricas basadas en precisión. Dentro de las métricas estándares usadas para medir la precisión de un sistema encontramos la precisión y la exhaustividad o recall [54, 97], las cuales computan el porcentaje de ítems para los cuales la predicción obtenida es correcta. En el contexto donde la empleamos, se evalúa un conjunto o listado ordenado de predicciones y por tanto, una alternativa es fijar una posición y evaluar esta métrica solo para los k predicciones superiores. Enfocándonos en la precisión, obtenemos la métrica de precisión a k ($P@k$), la cual se computa como:

$$P@k(U) = \frac{1}{k} \sum_{i=1}^k rel(R_i, U), \quad (2.20)$$

donde $rel(R_i, U) = 1$ si el i -ésimo ítem en el *ranking* ha sido accedido (es relevante) por el usuario U y $rel(R_i, U) = 0$ en caso contrario.

Otra métrica que utilizamos para evaluar la capacidad del sistema de mostrar el ítem en las primeras posiciones, es la Ganancia acumulada de descuento normalizada (nDCG) [97]. Esta

métrica tiene en cuenta las posiciones en el *ranking* de los ítems accedidos por el usuario U , $Acc(U)$. En particular, se penaliza un ítem relevante cuando aparece en posiciones inferiores en el *ranking* \mathcal{R} . En esta métrica, el valor de relevancia es logarítmicamente reducido en proporción a la posición que se encuentre en el resultado.

$$DCG = rel(R_1, U) + \sum_{i=2}^d \frac{rel(R_i, U)}{\log_2 i}. \quad (2.21)$$

Dado que el conjunto de ítems relevantes depende del usuario que se analiza, el valor DCG no es comparable entre usuarios. Para facilitar la comparación, la ganancia acumulada puede ser normalizada considerando los resultados obtenidos, empleando un *ranking* perfecto. Esta medida es conocida como $nDCG$. Todos los cálculos de $nDCG$ son entonces valores relativos y por tanto pueden ser comparados entre usuarios.

Parte I

Recomendación basada en filtrado colaborativo

En esta primera parte de la investigación, nos centramos completamente en los SR basados en filtrado colaborativo, que como hemos mencionado, simulan una ayuda grupal ante la incertidumbre que generan las múltiples opciones que nos encontramos día a día.

Primeramente proponemos un método para la selección de vecinos, centrándonos en la capacidad de cada usuario de predecir las votaciones previas del usuario al que se le hace la recomendación. Posteriormente, y considerando la tarea de predicción de voto, evaluamos por un lado, la propuesta anterior respecto a otras aproximaciones clásicas, y por otro lado, la influencia de determinadas características del conjunto de datos en la recomendación.

Aprovechando toda la información disponible de la etapa anterior, diseñamos una propuesta de explicación para mostrarle al usuario las razones por las que se le hace la recomendación. Nuestra hipótesis hasta aquí, es que si un usuario es capaz de predecir los votos previos de un usuario activo, entonces será un buen candidato tanto para formar el vecindario como para explicar las recomendaciones.

La propuesta de interfaz la insertamos en un SR de películas y desarrollamos una experimentación con usuarios reales, a los que mostramos además de nuestra propuesta, otra explicación clásica de gran aceptación por los usuarios (histograma de votos del vecindario). En esta experimentación comprobamos la importancia que cada usuario concede a determinada información de la explicación en el momento de evaluar la recomendación recibida, por lo que finalmente decidimos realizar un estudio para comprobar si este comportamiento lo podíamos detectar automáticamente, con el fin de aplicarlo para mejorar la recomendación del SR.

Capítulo 3

Modelos de recomendación basados en filtrado colaborativo

En este capítulo se tratan dos temas relacionados con los modelos de filtrado colaborativo. El primer tema está relacionado con la selección de los mejores vecinos para realizar la recomendación. En el segundo tema nos centramos en la evaluación del impacto en la recomendación de determinadas características.

En la selección del vecindario, primeramente se describen aproximaciones clásicas que permiten comprender y visualizar las deficiencias que estas presentan. Seguidamente se muestra nuestra alternativa para seleccionar los mejores vecinos, en el que nos centramos más que en una simple correlación lineal entre los vectores de votos, en la capacidad que tienen los vecinos de predecir los votos previos del usuario al que se le hace la recomendación. En esta idea consideramos que los usuarios que formen parte del vecindario son los que mejor son capaces de predecir los votos previos, por lo tanto, serán los que mejor estimarán los votos futuros.

Relacionado con el segundo tema, evaluación del impacto en la recomendación de determinadas características, la idea es estudiar, por ejemplo, si el que un ítem haya sido votado muy pocas veces, o el que un usuario vote con una cierta escala, son situaciones que afectan el proceso de recomendación.

En la metodología de experimentación nos centramos en la predicción de un voto para evaluar cada uno de los temas anteriores y utilizamos el conjunto de datos MovieLens, analizado en detalles en el capítulo anterior. Cada uno de los objetivos es analizado en una sección donde se muestran los principales resultados.

En la primera parte del capítulo se describe nuestra propuesta de selección de vecinos. Seguidamente se describe la experimentación considerando la propuesta anterior y otra alternativa clásica, comentando los principales resultados obtenidos. El estudio experimental incluye el análisis del impacto en la recomendación de determinadas características del conjunto de datos.

3.1. Selección de vecinos con alta calidad predictiva

La elección de los mejores usuarios que pueden ayudar a predecir el voto para un ítem determinado, es muy importante en los algoritmos de recomendación basados en vecindad. Algunas aproximaciones clásicas consideran una correlación lineal para la selección de vecinos, pero como veremos en esta sección, este método no siempre garantiza la elección de los mejores vecinos. Entre dos usuarios en un SR existen relaciones que la correlación lineal no puede detectar.

En esta sección describimos en detalles la forma de seleccionar el vecindario que consideramos como alternativa a las aproximaciones clásicas. Para ilustrar dicho proceso mucho mejor, se considerará el análisis del conjunto de votos en la Tabla 3.1. El objetivo que se persigue en el ejemplo es computar la predicción a ofrecer a Ann para el ítem seleccionado I_t , específicamente para el $I_t = I_7$.

Siguiendo los métodos basados en vecindario [50, 52], el primer paso para encontrar los vecinos es pesar todos los usuarios/ítems con respecto al criterio de similitud seleccionado. Así, al considerar el coeficiente de correlación de Pearson (PC), pueden encontrarse ocho candidatos que tienen el mismo valor de PC, donde $PC(Ann, Sara) = PC(Ann, John) = PC(Ann, Bill) = PC(Ann, Karl) = PC(Ann, Jane) = PC(Ann, Josh) = PC(Ann, Roge) = PC(Ann, Rina) = 1$, por lo tanto, no puede distinguirse de estos candidatos en el momento de realizar una selección de $n < 8$ vecinos.

Tabla 3.1: Votos de usuarios. El objetivo es proveer una predicción a Ann para el ítem I_7 .

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	...	I_{18}
Sara	5	4	5	3	3	3	2	2	2	2	2	2	...	2
John	4	3	4	2	2	1	2	1	1	1	1	1	...	1
Bill	3	2	3	1	1	1	5	1	3	3	3	3	...	3
Juan	2	1	2	1	2	1	3	3	2	2	2	2	...	2
Katy	3	2	3	2	2	2	4	2	1	1	1	1	...	1
Karl	5			2		1	2	2	2	1	2	2	...	2
Jane		3			2		2	3	3	3	1	1	...	1
Josh			4	2			5	4	2	2	2	2	...	2
Roge		3		2			3	4	1	2	3	3	...	3
Rina	4				2		4	4	4	4	2	2	...	2
Ann	4	3	4	2	2		?							

Como se ha visto en el capítulo anterior (Véase Sección 2.2), en el proceso de predicción, cada usuario candidato sugiere un voto $f_A(V, t)$, el cual es combinado para obtener la predicción final, (Ecuación 2.5). Así, independientemente del criterio empleado para computar $f_A(V, t)$, cada usuario seleccionado puede sugerir un voto diferente. Por ejemplo, si

$f_A(V, t) = \bar{r}_a + (r_{v,t} - \bar{r}_v)$, se tiene que $f(\text{Sara}, I_t) = 2,42$, $f(\text{John}, I_t) = 3,21$ y $f(\text{Bill}, I_t) = 5,47$.

Dada la situación anterior, surge una incógnita: ¿cuál de las predicciones anteriores o la de los restantes vecinos con PC igual a 1 es la mejor predicción para Ann?, o en otras palabras, ¿qué usuario puede ayudar a realizar una mejor predicción? En lo adelante se mostrará como atacar dicha situación desde una nueva perspectiva, donde se computa la similaridad entre usuarios a través de la capacidad de predicción sobre los ítems pasados.

De forma similar a [58], la idea será en primer lugar, evaluar la capacidad de cada usuario de predecir los votos pasados de Ann, y finalmente, seleccionar el vecindario que mejor comportamiento predictivo muestre. En el caso que se tiene de ejemplo (Tabla 3.1), al enfocarse en el ítem I_1 , el cual ha sido valorado por Ann con 4★, podríamos preguntarnos cómo sería la sugerencia de Sara para este ítem en particular. Empleando $f_{Ann}(\text{Sara}, I_1) = \bar{r}_{Ann} + (r_{\text{Sara}, I_1} - \bar{r}_{\text{Sara}})$ podemos esperar que la predicción sea $f(\text{Sara}, I_1) = 3 + (5 - 2,58) = 5,42$. Similarmente, para John y Bill se tendrían las sugerencias, $f_{Ann}(\text{John}, I_1) = 5,21$ y $f_{Ann}(\text{Bill}, I_1) = 3,47$ respectivamente. Llegado a este punto es posible percatarse que la sugerencia más cercana respecto al voto real dado por Ann, es la ofrecida por Bill. De estos candidatos a vecinos, Bill es hasta ese momento el que mejor capacidad predictiva muestra, y por tanto, de los tres el que primero formaría parte del vecindario. Lo que hemos descrito hasta aquí es la base de nuestra propuesta, donde se diseña un indicador capaz de medir la calidad predictiva de los usuarios para formar el vecindario.

En la Figura 3.1 tenemos una representación del comportamiento predictivo de un grupo de usuarios. Siguiendo con el usuario objetivo del ejemplo anterior, la primera fila representa cada una de las valoraciones de Ann según la escala de la Figura 3.2. Por cada usuario, en la Figura 3.1 se tiene una fila, donde cada celda según la escala anterior tiene un color, que representa el valor absoluto de la diferencia entre el voto sugerido por el usuario y el valor dado por Ann para dicho ítem (el color blanco significa que el usuario no valoró el ítem). El color verde nos representa un acierto total y el resto de los colores según se avanza en la escala mayor diferencia en la predicción.

En el ejemplo de la Figura 3.1 encontramos ordenados decrecientemente los usuarios por su capacidad predictiva y podemos observar como los 10 primeros usuarios predicen con alto acierto un gran número de los ítems en común con Ann. La capacidad predictiva de los restantes usuarios no es tan buena, incluso disponen de un menor número de ítems en común.

Luego de analizar los ejemplos anteriores, valoramos como hipótesis en lo adelante, que si un usuario U fue bueno para predecir los votos pasados del usuario activo, entonces él/ella será igual de bueno para predecir los votos de los ítems no vistos aún, y por tanto, será un buen candidato a formar parte del vecindario. Nótese que el principal elemento en esta propuesta es que se considerarán las predicciones de los votos en lugar de los votos originales. Por lo tanto, se considerará por una lado, los votos dados por el usuario activo, $R_A = \{r_{a,1}, \dots, r_{a,m}\}$

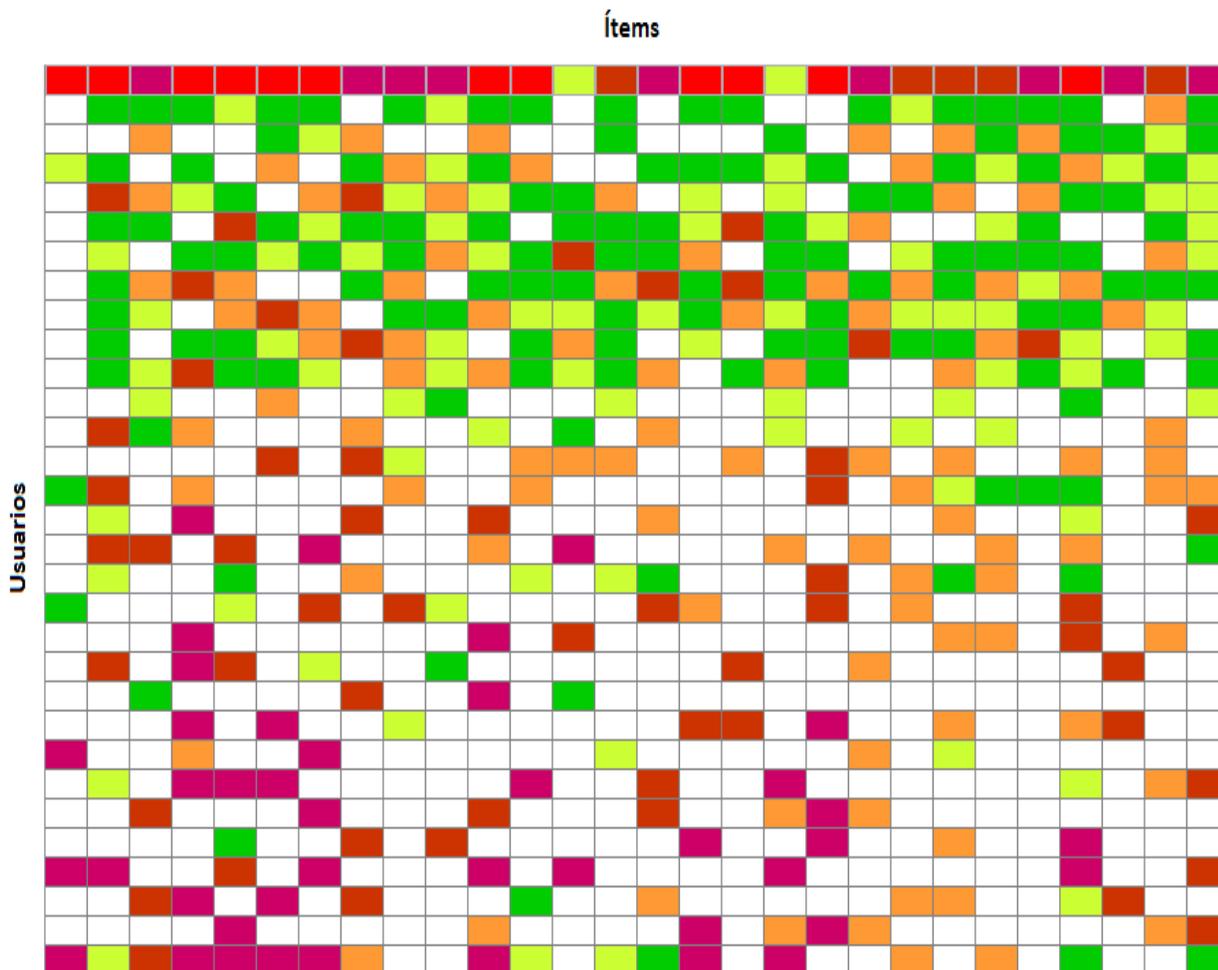


Figura 3.1: Predicciones de los usuarios para las valoraciones de Ann.



Figura 3.2: Escala de colores para representar el voto.

a los m ítems evaluados, y por otro lado, los votos que son sugeridos (recomendados) para cada ítem por parte del usuario U , $\hat{F}_A(U) = \{\hat{r}_{a,1}, \dots, \hat{r}_{a,m}\}$.

Con el objetivo de determinar la calidad de las predicciones, la cual mide en cierta medida la capacidad predictiva de un usuario, se considera que tanto los votos estimados, $\hat{F}_A(U)$, y los originales, R_A , son similares en términos absolutos. Se propone para ello el uso de una función de pérdida para medir la magnitud de la diferencia entre los dos votos, $L(r_{a,i}, \hat{r}_{a,i})$, representando el costo incurrido entre el verdadero voto u original, $r_{a,i}$, y el estimado, $\hat{r}_{a,i}$, por el usuario U . Particularmente, se considera la desviación absoluta entre el voto estimado,

$\hat{r}_{a,i}$, y el verdadero, $r_{a,i}$:

$$L(r_{a,i}, \hat{r}_{a,i}) = \text{abs}(r_{a,i} - \hat{r}_{a,i}). \quad (3.1)$$

Se debe señalar que no es posible utilizar la correlación de Pearson o la medida coseno para este propósito, aunque ambas medidas expresan la proximidad entre las calificaciones. Como es bien sabido, una alta correlación no implica necesariamente que se tengan buenas predicciones. Imaginemos la situación donde la predicción es $r + k$ con una correlación igual a uno, acá se empeora la predicción con el valor de k . Por ejemplo, en la Tabla 3.1, la $PC(Ann, Sara) = PC(Ann, John) = 1$, pero considerar a Sara en este caso, que evaluó con un punto más que Ann todos los ítems, empeoraría la predicción. El caso de considerar a John sería el ideal.

Por lo tanto, la capacidad predictiva de un usuario puede ser medida al considerar la pérdida esperada en las valoraciones anteriores del usuario activo. Nótese que al igual que los enfoques basados en vectores, la pérdida esperada se calcula teniendo en cuenta los votos comunes, es decir, en aquellos votos comunes entre el usuario activo y el usuario del que se mide su capacidad predictiva, $R_A \cap R_U$, siendo R_\bullet el subconjunto de ítems evaluados por el usuario:

$$EL(A, U) = \frac{\sum_{i \in R_A \cap R_U} L(r_{a,i}, \hat{r}_{a,i}(u))}{|R_A \cap R_U|}. \quad (3.2)$$

Dado que se ha considerado un enfoque basado en memoria al recomendar un voto para un ítem específico, I_t , debe aclararse que se computa $EL(A, U)$ para aquellos usuarios, U , que hayan evaluado el ítem activo, I_t , es decir, todos los usuarios, U tales que $r_{u,t} \in R_U$. En el ejemplo que estamos analizando para Ann y el ítem I_7 , se computaría $EL(A, U)$ para todos los usuarios dado que todos han evaluado el ítem. Considerando el voto dado por cada usuario como el voto sugerido, se tendría $EL(Ann, Sara) = (|4-5|+|3-4|+|4-5|+|2-3|+|2-3|)/5 = 1$, $EL(Ann, John) = (|4-4|+|3-3|+|4-4|+|2-2|+|2-2|)/5 = 0$ y $EL(Ann, Katy) = 0,6$.

Estos usuarios serán ordenados en orden creciente de la pérdida esperada ($EL(Ann, John) < EL(Ann, Katy) < EL(Ann, Sara)$) y los N primeros serán seleccionados para formar parte de la vecindad del usuario activo. Entonces, los votos sugeridos serán combinados de tal manera que los mejores predictores tendrán un mayor peso en la combinación que se computará.

De forma general, en la Tabla 3.2 se presenta un modelo de recomendación que incluye las ideas descritas anteriormente para la selección del vecindario. En el resto de la memoria identificaremos este algoritmo como **LM**.

Al considerar la eficiencia de la propuesta, se debe destacar que el enfoque es equivalente a las medidas basadas en vectores, ya que las predicciones se pueden calcular en un tiempo

Tabla 3.2: Modelo de recomendación. Selección de vecinos con buena calidad predictiva.

Entradas: A usuario activo, I_k ítem activo
Salida: r_a voto estimado
1. Selección de vecinos
1.1 Por cada U que haya evaluado el ítem activo I_t
1.1.1 Computar $EL(A, U)$,
$EL(A, U) = \frac{\sum_{i \in R_A \cap R_U} L(r_{a,i}, \hat{r}_{a,i}(u))}{ R_A \cap R_U }$
1.2 Ordenar los usuarios y seleccionar los N mejores vecinos, es decir, aquellos con menor E.L
2. Proceso Predictivo
2.1 Computar un voto $r_{a,t}$ de acuerdo a
$\hat{r}_{a,t} = \frac{\sum_{V \in N_t(a)} EL(A, V) f_A(V, t)}{\sum_{V \in N_t(a)} EL(A, V)}.$

constante, suponiendo claramente que algunos valores como el voto medio del usuario o la desviación estándar se hayan computado previamente.

3.1.1. Impacto de características en la recomendación

Antes de continuar con la próxima sección de experimentación, retomaremos los ejemplos anteriores para ejemplificar el segundo de los objetivos que evaluaremos. En la Tabla 3.1, donde se persigue proveer una predicción a Ann para el ítem I_7 , este ítem fue evaluado por todos los usuarios con votos entre $2\star$ y $5\star$. Sin embargo, ¿qué pasaría si el objetivo de predicción fuera el ítem I_6 ?. Este último ítem solamente fue evaluado por seis usuarios y con bajos votos, entre $1\star$ y $3\star$.

La Figura 3.1 en su primera columna, muestra una situación similar a lo que ocurre con el ítem I_6 en el ejemplo anterior y con otros ejemplos en general, ítems muy poco valorados. La situación anterior nos llevó a preguntarnos si los resultados finales de predicción que se obtienen, dependen del número de usuarios que han valorado cada uno de los ítems y de la forma de evaluación.

En la tarea que nos centramos en la experimentación, predicción de votos, tradicionalmente se emplean métricas estándares como la MAE y RMSE para evaluar el desempeño de los sistemas. Estas métricas consideran que la desviación estándar del error es constante sobre todas las predicciones, es decir, cada predicción tiene igual peso en el computo final de la métrica.

La anterior suposición no siempre es adecuada, por ejemplo, en un SR de libros o películas, puede que la precisión de las predicciones sea diferente y muy dependiente de la antigüedad del ítem. En los ítems más recientes, el tener menos valoraciones pudiera ser un elemento que incidiera en la predicción. En situaciones como está, cuando puede que no sea razonable suponer que todos los errores deben ser tratados por igual, se puede usar un peso asociado

a cada error, tratando de dar a cada error en la predicción la influencia apropiada sobre la medida global.

Diferentes características pueden ser tratadas en dependencia de los ítems, por ejemplo, en la recomendación de películas el género, la fecha de lanzamiento, precio, etc. Pero también, este análisis puede realizarse considerando atributos asociados al usuario, por ejemplo, su ubicación, distribución de votos, etc. Por ejemplo, considerando la fecha de lanzamiento, se puede asignar pesos de tal manera que cuanto mayor es el peso, mayor importancia se le da a las películas más recientes, e inversamente, para los pesos pequeños. En este caso observamos que aún cuando el error promedio es menor que un umbral razonable, el rendimiento de un sistema podría ser inadecuado en el análisis de esta característica en particular.

Por medio de estos pesos se podría relacionar el error en las predicciones con cualquier característica que pueda ser relevante tanto para el usuario como para el vendedor. También se puede investigar sobre la forma en que el SR funciona, mediante la asociación de diferentes pesos a los valores de un atributo directamente involucrado en las predicciones. Así, cuanto mayor sea el peso, mayor es la importancia del atributo que se analiza.

En la segunda parte de la experimentación, nos enfocaremos en una alternativa para medir el impacto de una determinada característica en la recomendación, error medio absoluto ponderado (wMAE), aunque pudiera emplearse el error cuadrático de forma similar (wRMSE). Una ventaja de esta alternativa que empleamos, es que se puede medir fácilmente la importancia de una característica en cada predicción. Esta medida se ha utilizado en áreas como la meteorología o las previsiones económicas [122]. Parafraseando a George Orwell en su novela “Rebelión en la granja”(1945), podemos decir que “Todos los errores son iguales, pero algunos errores son más iguales que otros.

3.2. Experimentación

Para comprobar la utilidad de la propuesta descrita en la sección anterior, se seleccionó el conjunto de datos MovieLens. En esta sección primeramente se detalla la metodología utilizada y posteriormente se describe toda la experimentación realizada, en la cual, se muestran los principales resultados obtenidos.

3.2.1. Metodología de experimentación

Una vez centrados en el entorno donde se desarrollará la experimentación, se definen los objetivos que se persiguen, la forma en que se emplean los datos y las medidas para evaluar el estudio. A continuación se describe cada uno de estos elementos para dar paso al análisis de los resultados.

- **Objetivos de la experimentación:** El primer objetivo de la experimentación es determinar la calidad del vecindario seleccionado al emplear el método descrito en la sección previa, para ello se medirá la capacidad de predecir el interés de un usuario hacia un ítem no visto, es decir, se atacará la tarea de predicción de voto.

El segundo objetivo es estudiar el uso de un factor de peso en el error promedio, donde los pesos indicarán la importancia subjetiva asociada a cada predicción.

- **Conjuntos de entrenamiento y prueba:** Para validar el modelo propuesto, y de forma similar a muchas metodologías de evaluación, se divide aleatoriamente el conjunto de datos en conjuntos de entrenamiento (contiene el 80% de los votos) y prueba (contiene el 20%). Para reducir la variabilidad de los resultados, se ejecuta una validación cruzada de 5 particiones de datos diferentes. Los resultados que se muestran en los resultados obtenidos, representan el promedio de los valores obtenidos sobre las cinco particiones.

- **Medidas de evaluación:** En el primero de los objetivos, para comprobar el desempeño del modelo se medirá qué tan bien el sistema predice los votos o preferencias de los usuarios, es decir, la precisión del sistema. Siguiendo [54], se propone el uso de tres diferentes métricas de evaluación (Hamming, MAE y RMSE), las cuales como ya se mencionó en el capítulo anterior básicamente difieren en el costo asociado a cada error.

Para el segundo de los objetivos, la métrica se computará de la siguiente forma:

$$wMAE = \frac{\sum_{i=1}^U \sum_{j=1}^{N_i} w_{i,j} \times \text{abs}(p_{i,j} - r_{i,j})}{\sum_{i=1}^U \sum_{j=1}^{N_i} w_{i,j}}, \quad (3.3)$$

donde U indica el número de usuarios; N_i el número de ítems predichos por el i -ésimo usuario; $r_{i,j}$, el voto dado por el i -ésimo usuario al ítem I_j ; $p_{i,j}$, el voto predicho por el modelo y $w_{i,j}$ representa el peso asociado a cada predicción. Nótese que cuando todas las diferencias individuales tienen igual peso, $wMAE$ coincide con la MAE.

3.2.2. Resultados y análisis

Definida la metodología de experimentación a utilizar, seguidamente se describen los resultados obtenidos al comprobar cada uno de los objetivos definidos.

3.2.2.1. Evaluación de la calidad del vecindario seleccionado para la predicción

Se debe mencionar que algoritmos diferentes se estudiaron para seleccionar los vecinos: el primero requirió en el momento de realizar las recomendaciones, que los usuarios candidatos hubiesen evaluado en común al menos 10 de los ítems que en el pasado el usuario activo

Tabla 3.3: Desempeño de la métrica RMSE con al menos 10 votos en común.

RMSE				
Vecinos	LM-Raw	LM-NormR	BL-Raw	BL-NormR
5	1,08927	1,0613	1,15479	1,09998
10	1,05966	1,0240	1,12668	1,07184
20	1,04834	1,0073	1,11382	1,05874
30	1,04993	1,0020	1,11124	1,05751
50	1,05622	0,99986	1,11319	1,05750

Tabla 3.4: Desempeño de la métrica Hamming con al menos 10 votos en común.

Hamming				
Vecinos	LM-Raw	LM-NormR	BL-Raw	BL-NormR
5	0,62461	0,60277	0,6293	0,61414
10	0,61331	0,58816	0,6205	0,60283
20	0,61165	0,58316	0,61876	0,59550
30	0,61398	0,5814	0,61938	0,59460
50	0,61877	0,58155	0,62199	0,59384

evaluó. En el segundo algoritmo esta restricción fue eliminada, requiriéndose solamente un ítem en común.

Para ilustrar el desempeño del modelo que se propone, se han considerado dos alternativas diferentes para identificar el vecindario: la primera ha sido propuesta en [50] y se considera como un modelo de referencia (BL), en donde los mejores vecinos son obtenidos empleando la correlación de Pearson. En la segunda alternativa los vecinos son seleccionados empleando la ecuación de pérdida esperada (EL) sobre todos los votos estimados (Ecuación 3.2).

Además, para estimar el voto a sugerir, se consideran dos formas de computo: una empleando el valor de los votos dados por los usuarios (definida en el capítulo anterior como voto bruto), en lo adelante Raw, y la otra, NormR, normalizando los votos sugeridos usando la media de votos del usuario (definida como sugerencia centrada en la media), con esta última variante se suelen obtener muy buenos resultados. Nótese que la combinación BL-NormR es el modelo propuesto en [50], el cual sigue siendo un modelo del estado del arte en los algoritmos de filtrado colaborativo basados en memoria [38, 57].

Otro elemento que se debe destacar antes de mostrar los resultados obtenidos, es el referente al tamaño del vecindario. Se consideró optar por un número fijo de vecinos. Se emplean diversas configuraciones con cada modelo, seleccionando los mejores, 5, 10, 20, 30 y 50 vecinos.

Las Tablas 3.3, 3.4, 3.5, 3.6, 3.7 y 3.8 muestran el desempeño obtenido con las diferentes métricas en el momento de seleccionar los vecinos. Las tres primeras se enfocan para cuando se requiere que los vecinos seleccionados hayan evaluado al menos 10 ítems en común con el

Tabla 3.5: Desempeño de la métrica MAE con al menos 10 votos en común.

MAE				
Vecinos	LM-Raw	LM-NormR	BL-Raw	BL-NormR
5	0,79471	0,76085	0,82851	0,78397
10	0,76839	0,72839	0,8042	0,75786
20	0,76051	0,71505	0,79519	0,74457
30	0,76323	0,71075	0,79402	0,74301
50	0,77072	0,7096	0,79745	0,74218

Tabla 3.6: Desempeño de la métrica RMSE con al menos 1 voto en común.

RMSE				
Vecinos	LM-Raw	LM-NormR	BL-Raw	BL-NormR
5	1,14236	1,08232	1,12510	1,07305
10	1,09832	1,03733	1,08319	1,03235
20	1,07205	1,01638	1,06364	1,01324
30	1,06560	1,01160	1,05865	1,00746
50	1,06347	1,00865	1,05650	1,00402

Tabla 3.7: Desempeño de la métrica Hamming con al menos 1 voto en común.

Hamming				
Vecinos	LM-Raw	LM-NormR	BL-Raw	BL-NormR
5	0,6464	0,61836	0,63809	0,6237
10	0,63113	0,60448	0,62742	0,60901
20	0,62683	0,59466	0,62173	0,60051
30	0,62666	0,59285	0,62097	0,59622
50	0,62654	0,5915	0,62164	0,59464

usuario activo, mientras en las Tablas 3.6, 3.7 y 3.8 se muestra el comportamiento cuando los vecinos seleccionados tienen al menos solo un ítem en común valorado previamente. En este sentido se ha empleado un peso para devaluar las similitudes que se basan en un número pequeño de ítems evaluados en común, al igual como se describió en el capítulo anterior en la sección de filtrado colaborativo (Sección 2.2). En todas las tablas se destacan en negrita los mejores resultados para cada experimento.

Al analizar los resultados obtenidos pueden extraerse las siguientes conclusiones: primeramente, los mejores resultados han sido obtenidos en todos los experimentos cuando es considerada la sugerencia basada en la media de los votos; segundo, el tamaño del vecindario ha tenido un impacto significativo en la calidad de los resultados obtenidos, las medidas empleadas mejoran según se incrementa el número de miembros del vecindario desde 5 hasta

Tabla 3.8: Desempeño de la métrica MAE con al menos 1 voto en común.

MAE				
Vecinos	LM-Raw	LM-NormR	BL-Raw	BL-NormR
5	0,84489	0,78695	0,82666	0,78524
10	0,8055	0,74912	0,79271	0,74995
20	0,78668	0,72933	0,77678	0,73208
30	0,78248	0,72501	0,77317	0,72543
50	0,781	0,72204	0,77277	0,72203

50 vecinos. Estas diferencias reflejadas son estadísticamente significativas, comprobadas al emplear el test estadístico de Wilcoxon.

Por último, si analizamos los métodos de referencia (BL) para cada una de las métricas, al considerar las dos opciones de ítems en común (Tablas [3.3 y 3.6], [3.4 y 3.7], [3.5 y 3.8]), se obtienen los mejores resultados para cuando un vecino puede tener hasta un solo ítem en común. Como en la opción de al menos un ítem en común, se considera que un usuario con pocas valoraciones en común con el usuario activo (menos de 10) puede formar parte del vecindario, en estos modelos se llegará a tener en el vecindario algunos usuarios con una correlación muy alta pero con pocas valoraciones. Los usuarios que cumplan con la condición anterior (alta correlación y pocas valoraciones) no formarán parte de los vecindarios en la opción donde se condiciona que se tengan al menos 10 ítems en común, afectando las medidas globales de los métodos de referencia.

Sin embargo, el no considerar los usuarios anteriores (alta correlación y pocas valoraciones) en la forma de seleccionar el vecindario en nuestra propuesta, se ve reflejado en la mejora de las medidas globales. Una interpretación de este resultado es que un usuario confiará más en aquellos vecinos con los que comparte más experiencias en común.

Como conclusión general se puede decir que nuestra propuesta de criterio de selección de vecinos es competitiva y se comporta de forma superior respecto a la estándar basada en la correlación de Pearson, todo esto en términos de eficiencia y eficacia (existiendo diferencias significativas entre los resultados obtenidos). Estos elementos confirman la hipótesis que la medición de la capacidad predictiva de los vecinos puede ser útil para propósitos de predicción.

3.2.2.2. Evaluación de las dependencias del error en la predicción

Para comprobar el segundo objetivo de esta experimentación, presentamos los resultados obtenidos al emplear cuatro estrategias de recomendación. En la primera, las predicciones se estiman computando el promedio de los votos de cada usuario (se identificará con *Means*), \bar{r}_a . En la segunda estrategia empleamos el modelo definido en la Tabla 3.2 para N igual a 20 vecinos, al que ya identificamos como **LM**. Siendo los modelos *Slope One* y *SVD*, en sus implementaciones en Mahout [4], las restantes estrategias utilizadas.

Debemos señalar que aunque trabajamos con los mismos conjuntos de datos, en las métricas que se computan no se consideran los usuarios con menos de 10 ítems en el conjunto de prueba. Según la consideración anterior, se podrán encontrar diferencias con los resultados de la sección previa.

Por otra parte consideramos dos diferentes factores para evaluar la dependencia del error en la predicción, uno relacionado con los ítems y el otro con las preferencias del usuario. En ambos casos asignamos pesos en el intervalo $[0, 1]$.

- Popularidad del ítem: investigaremos si el error en las predicciones depende del número de usuarios que valoraron los ítems. En el capítulo anterior, cuando analizamos el conjunto de datos MovieLens, mostramos una gráfica con la popularidad de los ítems, la cual sigue la teoría de la larga cola (*Long Tail*) [3]. Vamos a considerar dos alternativas diferentes:

i+ En este caso los pesos impondrán más penalización a las malas predicciones cuando el ítem ha sido votado frecuentemente (ítems que tienen un alto número de votos). También se penalizan las malas predicciones cuando éstas tienen un pequeño número de votos, pero no tanto como cuando tenemos ítems con más votos. Con la idea anterior asumimos que el poco número de votos no proporciona mucha información acerca de los factores latentes que influyen en los votos de los usuarios. Particularmente, para cada ítem I_i podemos considerar este peso como la probabilidad de que este ítem fuese votado en el conjunto de entrenamiento, es decir, $w_i = pr(I_i)$.

i- Esta es el inverso del criterio anterior, donde ponemos mayor énfasis en las predicciones sobre aquellos ítems con menos votos, por lo que los pesos son $w_i = 1 - pr(I_i)$.

- Distribución de votos: Como es bien conocido los usuarios no suelen valorar los ítems de forma uniforme en un patrón de votos. Existen algunas tendencias en valorar los ítems que gustan con altos valores y utilizar muy poco las bajas puntuaciones. Esto último se ve reflejado de manera general en la mayoría de los conjuntos de datos (la distribución de votos para el conjunto de datos que se emplea en la experimentación fue analizada en el capítulo anterior, véase Figura 2.14). Al considerar esta característica podemos medir si el error depende o no de la distribución de votos. Nosotros consideramos cuatro alternativas para evaluar los pesos:

rS+ Considerando la distribución de votos de todos los ítems del sistema. Particularmente, en esta estrategia ponemos mayor énfasis sobre el error en la predicción de aquellos votos comunes. Por tanto, los pesos son $w_i = pr_S(r_i)$, siendo r_i el voto del usuario al ítem I_i .

rS- Inversamente a la anterior, se asigna más peso a los votos menos comunes, es decir, $w_i = 1 - pr_S(r_i)$.

rU+ En este caso, y dado que diferentes usuarios puede usar un patrón diferente de votación, consideramos la distribución de votos del usuario, de tal manera que aquellos votos comunes para un usuario tendrán mayores pesos, es decir, $w_i = pr_U(r_i)$.

rU- Esta última asigna más peso al voto menos frecuente del usuario, es decir, $w_i = 1 - pr_U(r_i)$.

Comenzando con el análisis de los resultados obtenidos, en las Figuras 3.3 y 3.4 presentamos los resultados de la MAE y la wMAE para cada uno de los cuatro modelos de recomendación utilizados. La primera de las figuras muestra tanto la MAE como los resultados obtenidos cuando los pesos están directamente relacionados a la cada una de las características (correlacionados positivamente, rU+, rS+, i+), mientras la Figura 3.4 presenta los resultados cuando los pesos están inversamente relacionados (correlacionados negativamente, rU-, rS-, i-).

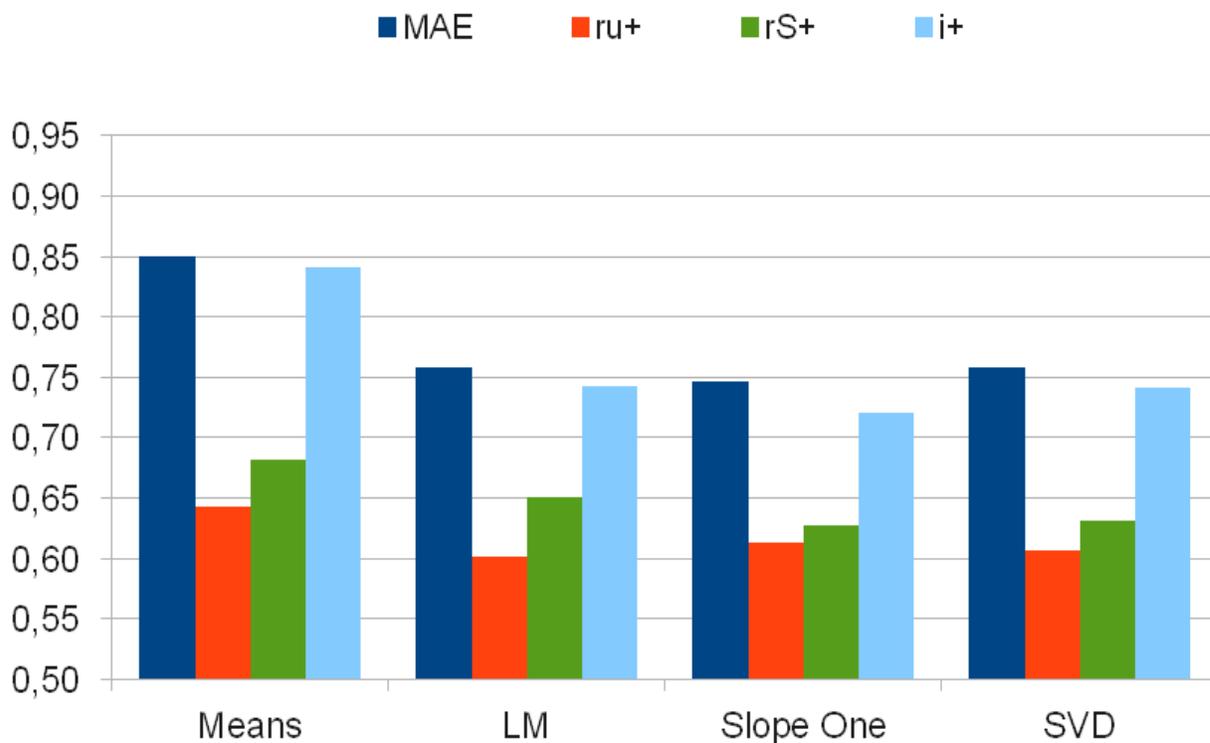


Figura 3.3: Pesos con correlación positiva en la wMAE.

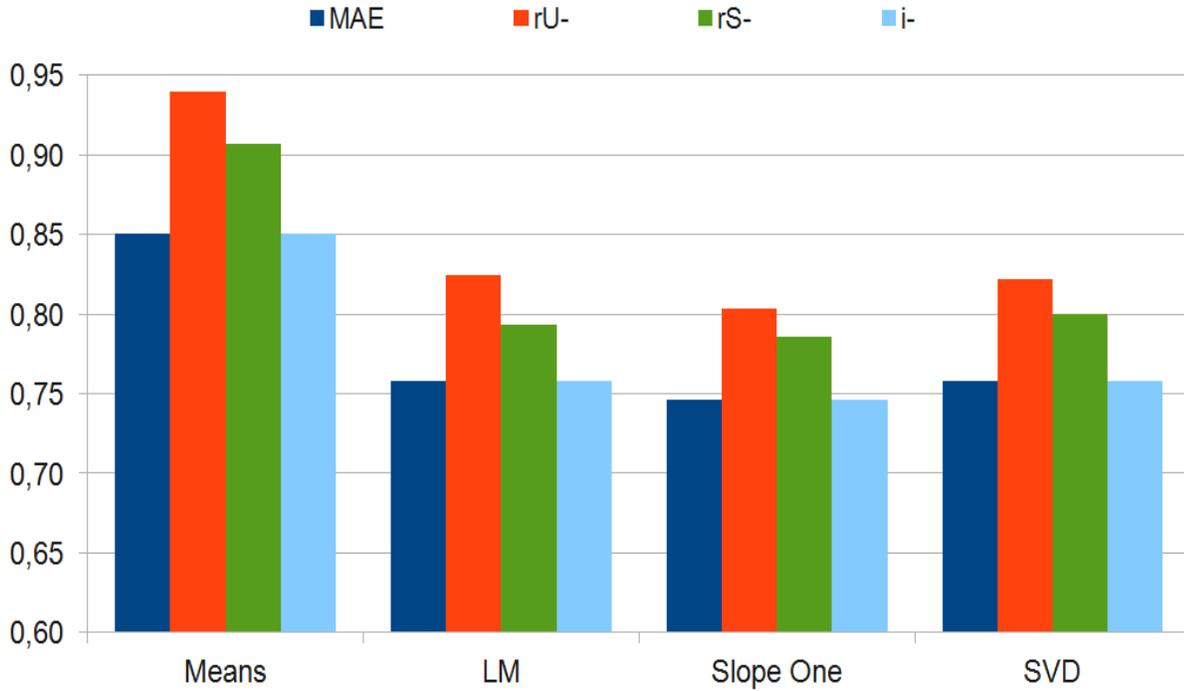


Figura 3.4: Pesos con correlación negativa en la wMAE.

Se observa que mediante el uso de la wMAE podemos determinar que el error es altamente dependiente del patrón de votos de los usuarios, y menos dependiente cuando consideramos la popularidad del ítem. Por otra parte, si comparamos las dos figuras anteriores, se puede observar que todos los modelos funcionan mejor para la predicción de los votos más comunes. En este sentido, son capaces de aprender las preferencias más frecuentes, y los grandes errores (mal desempeño) se obtienen cuando se centran en las predicciones sobre los votos con menos frecuencia. Cuando consideramos la popularidad del ítem, éstas diferencias son menos conclusivas. En cierto sentido, podemos decir que la forma que el usuario evalúa un ítem no depende de cómo de popular sea este ítem.

Comparación entre los pesos y el error relativo

Ahora exploraremos otra alternativa de evaluar el beneficio de utilizar la wMAE, para esto consideraremos el cociente entre la wMAE y la MAE ($wMAE/MAE$), siendo igual a:

$$wMAE/MAE = \frac{\sum_{i,j} w_{i,j} e_{i,j} / \sum_{i,j} e_{i,j}}{\sum_{i,j} w_{i,j} / N}, \quad (3.4)$$

donde $e_{i,j} = abs(p_{i,j} - r_{i,j})$.

Teniendo en cuenta que restringimos los pesos en el intervalo $[0, 1]$, el denominador

podría representar el porcentaje promedio de la masa de ítems que están relacionados con la dimensión que se considera, mientras que el numerador representa el porcentaje promedio del error proveniente de dicha característica. Tendríamos que cuando $wMAE/MAE > 1$, $wMAE > MAE$, el porcentaje de error proveniente de la característica es mayor que la masa asociada, por lo que podemos considerar que el sistema no es capaz de predecir adecuadamente dicha dimensión. Obviamente, lo inverso ocurre cuando $wMAE < MAE$. Cuando ambas métricas son iguales, esto implica que el error esperado es independiente de la característica.

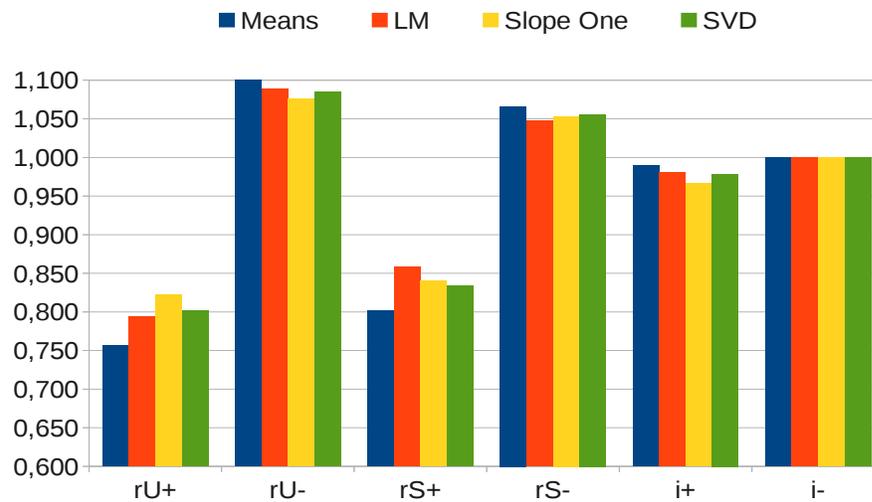


Figura 3.5: Relación entre la $wMAE$ y la MAE .

En la Figura 3.5, presentamos los valores de $wMAE/MAE$ para los cuatro modelos, donde, nuevamente podemos observar que existe una dependencia entre el voto y el error. El error asociado a los votos comunes es menor que la importancia relativa de esta característica en el sistema, mientras para los votos menos comunes el sistema no es capaz de realizar buenas predicciones, siendo mayor el error relativo de sus pesos asociados. Esta situación no se tiene cuando se considera la popularidad del ítem.

Estos son resultados globales, pero podemos considerar el comportamiento de las métricas para cada individuo. Para ello proponemos el uso de un diagrama de puntos, las Figuras 3.6 y 3.7 relacionan los pesos relativos (eje horizontal) con los errores relativos (eje vertical) para

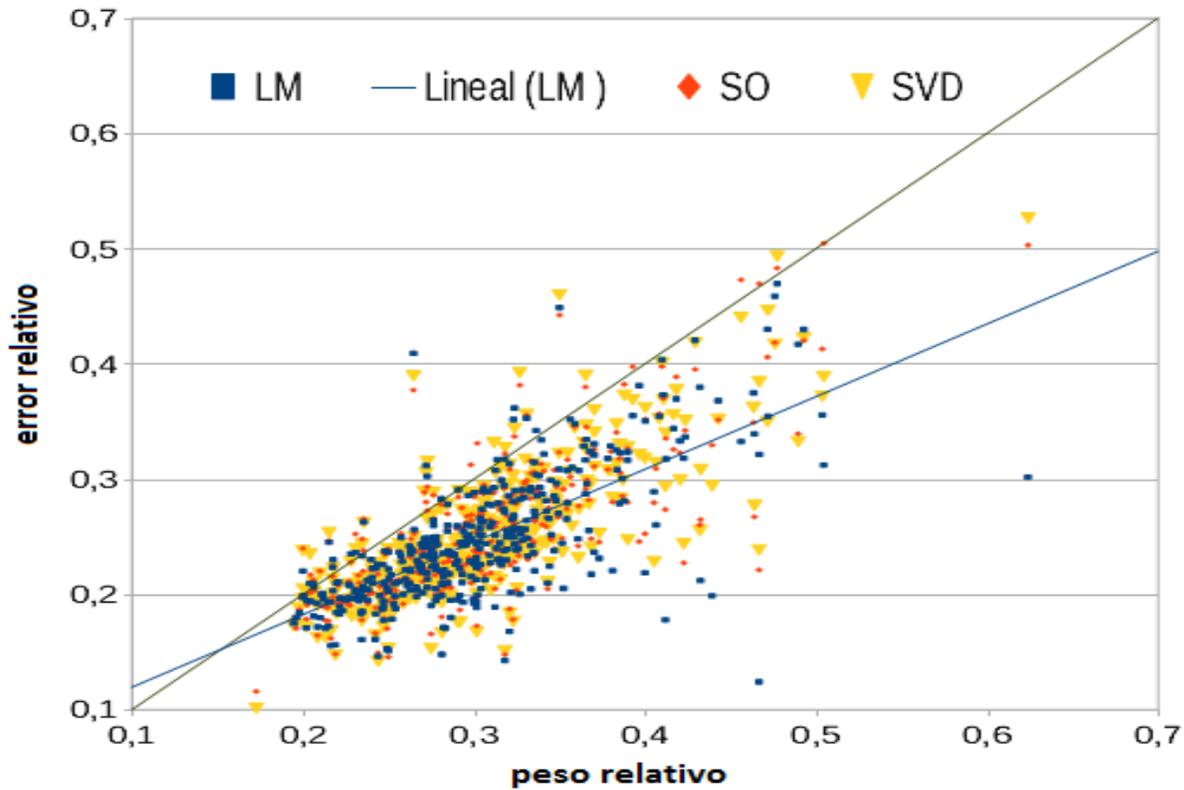


Figura 3.6: Relación entre el peso y el error utilizando $rU+$.

cada usuario en el sistema y por cada una de las estrategias de recomendación utilizadas¹. En la Figura 3.6 consideramos $rU+$ como factor de peso, mientras en la Figura 3.7 consideramos $i+$. Nótese que dado que son 5 posibles votos, el peso relativo es igual a 0,2 cuando todos los votos son igual de probables y este valor incrementa con la importancia de los votos más utilizados.

En la Figura 3.6 podemos ver que tanto el porcentaje de masa y el porcentaje de error están correlacionados positivamente, siendo $wMAE/MAE < 1$ para la mayoría de los usuarios. Además, existe una tendencia a mejorar las predicciones para aquellos usuarios con mayor masa relativa (por ejemplo, podemos ver como la regresión lineal para el modelo LM² se aleja de la línea $y = x$). En cierto modo podemos concluir que la utilidad de la recomendación de la distribución de votos es consistente para todos los usuarios y los modelos de recomendación utilizados. Por otro lado, en la Figura 3.7, si bien los pesos y el error están correlacionados positivamente, existen diferencias significativas entre los diferentes usuarios. Estos resultados no se reflejan en las medidas globales.

Comparación relativa entre los modelos

¹Son incluidos los usuarios con al menos 10 predicciones.

²El desempeño de los restantes modelos es similar, pero no fueron incluidos por razones de claridad en el diagrama.

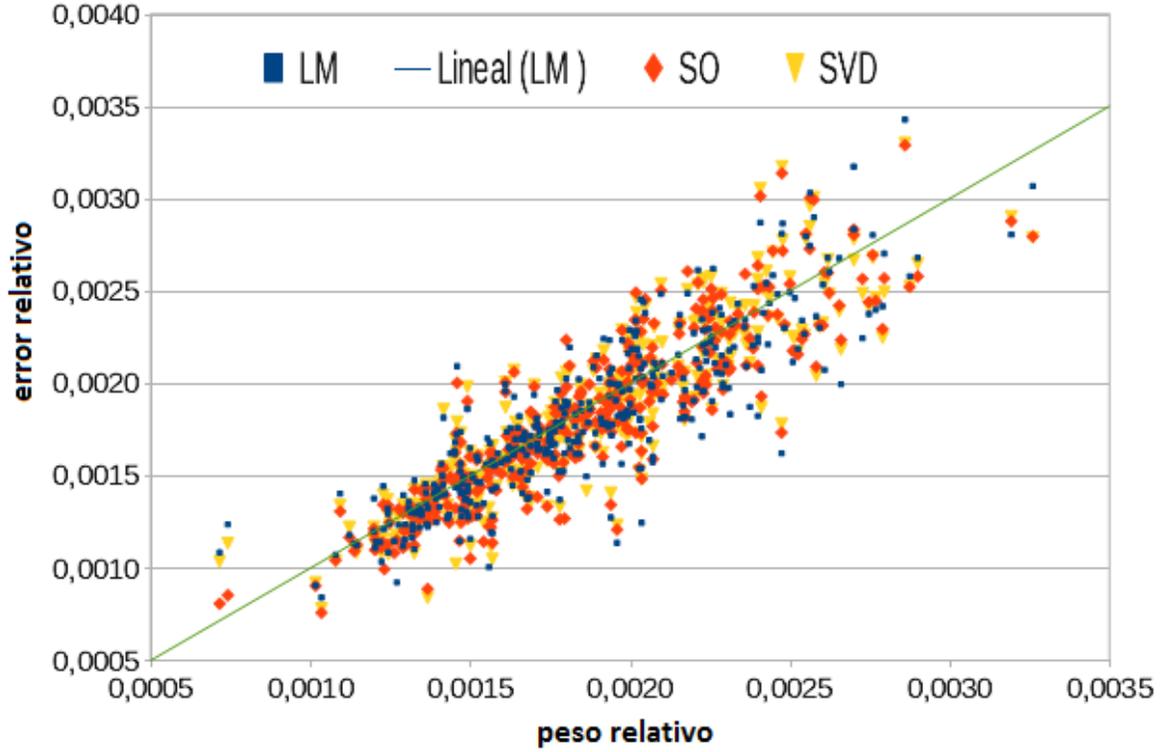


Figura 3.7: Relación entre el peso y el error utilizando $rI+$.

Para concluir este estudio nos gustaría valorar cuál es el significado del hecho de que un modelo alcanza una $wMAE$ igual a 0,6019. Aunque la $wMAE$ podría dar alguna información acerca de cómo se obtiene el error, existe un problema con esta medida, y es que no existe un criterio acerca de lo que es una buena predicción. Con el objetivo de atacar esta situación, proponemos el empleo del error relativo en lugar del error absoluto, es decir, el error medio absoluto porcentual ponderado, $wMAPE$.

Dado dos modelos, $M1$ y $M2$, la métrica relativa es definida como $wMAE_{M1}/wMAE_{M2}$, teniendo:

$$wMAPE_{M1} = \frac{\sum_{i=1}^U \sum_{j=1}^{N_i} w_{i,j} \times abs(p_{i,j}^{M1} - r_{i,j})}{\sum_{i=1}^U \sum_{j=1}^{N_i} w_{i,j} \times abs(p_{i,j}^{M2} - r_{i,j})}. \quad (3.5)$$

Cuando computamos esta métrica, mientras los resultados sean menores, mayores mejoras se tendrán. Retomando el ejemplo que que estamos analizando, si determinamos que $M2$ será el modelo que predice los votos del usuario empleando el promedio de sus votos (*Means*), obtenemos los resultados de la Figura 3.8 con el resto de los modelos.

Al analizar la primera tabulación del histograma de la Figura 3.8 ($rU+$), podemos comprobar cómo el modelo LM se ajusta mejor a las preferencias comunes del usuario. Sin

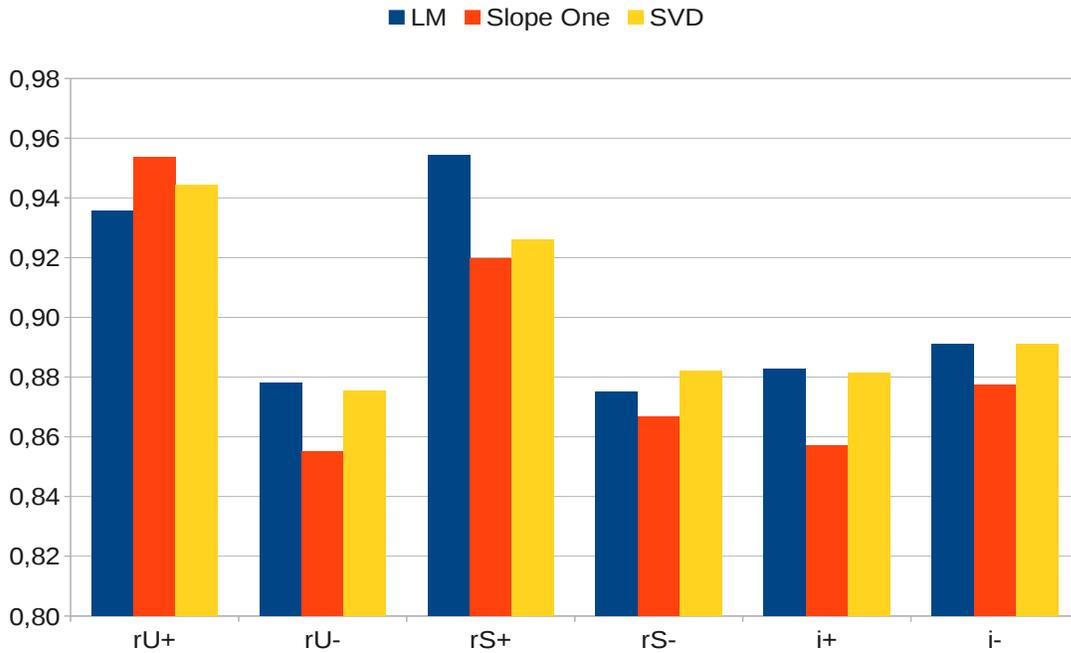


Figura 3.8: Comportamiento de modelos respecto a la métrica wMAPE.

embargo, en la tercera tabulación ($rS+$) comprobamos que los modelos de recomendación *Slope One* y SVD están más sesgados hacia la distribución de votos en general del sistema.

Analizando los resultados para la frecuencia de los votos menos frecuentes, tanto a nivel de usuario como de sistema ($rU-$ y $rS-$), se logran mejorar con respecto al voto promedio (*Means*), en un mayor grado al utilizar el modelo *Slope One*. Con respecto a la popularidad del ítem (últimas dos tabulaciones), podemos comprobar que todos los modelos mejoran su rendimiento al considerar los ítems más populares, aunque estas diferencias son menos significativas que cuando analizamos la frecuencia de votos.

3.3. Conclusiones

En el capítulo hemos propuesto y validado un criterio de selección de vecinos, el cual demostramos con los resultados alcanzados que es un método competitivo. El criterio se soporta en la idea de seleccionar los vecinos estimando su capacidad predictiva, lo que se convierte en una buena alternativa y confirma nuestra hipótesis inicial de que este vecindario dadas sus características, ofrece una mejor recomendación a los usuarios.

Independientemente del método seleccionado, el tamaño y calidad del vecindario tiene un impacto significativo en los resultados finales, obteniéndose mejores resultados para vecinda-

rios con un mayor número de vecinos y además, en el que estos vecinos hayan valorado un considerable número de ítems en común con el usuario al que se recomienda.

Finalmente proponemos una métrica para evaluar la importancia de determinadas características en la recomendación. Asignando un peso a las diferentes opciones según las características que se analizan, valoramos tanto el impacto de las distribuciones de votos (usuario y sistema) como la popularidad de los ítems.

El principal problema que presenta es que no se pueden comparar dos modelos con diferentes configuraciones de pesos, ya que no es una medida absoluta como la MAE y depende completamente del conjunto de pesos utilizado en la evaluación. Podemos decir que es una medida de evaluación relativa que permite conocer en las situaciones analizadas y en otras dada la posibilidad de su generalización, el origen de los errores y que ocurre internamente en el proceso de recomendación de los SR.

Capítulo 4

Explicación para sistemas de recomendación basados en filtrado colaborativo

En el capítulo atacamos uno de los principales problemas de los SR, de cara a los usuarios la desconfianza de éstos con las sugerencias recibidas. Por lo que hacemos una propuesta de explicación para SR de filtrado colaborativo.

De forma general en un proceso de recomendación social, los usuarios consideran las recomendaciones/opiniones de sus amigos y al mismo tiempo la experiencia previa que tienen sobre los productos. En general, la propuesta muestra a los usuarios una explicación, que le permita entender las razones por las que el sistema le hace dicha recomendación. Por un lado, consideramos la importancia que le dan los usuarios normalmente al consenso y opinión de los amigos y, por otro lado, el interés que muestran en confiar en aquellos productos de los que tienen experiencia previa. Ambos criterios son mezclados con la idea de mostrar a un usuario, cuál es el criterio de un determinado vecindario, del que sus miembros tiene criterios similares a él.

Creemos que un usuario antes de aceptar una recomendación, estaría satisfecho de conocer cuan relacionadas están sus experiencias pasadas con las necesidades presentes y, además, cuál es la opinión y el comportamiento de sus vecinos respecto a dichas relaciones. Para comprobar lo anterior, realizamos una experimentación con usuarios reales, donde empleamos además una interfaz de explicación clásica de gran aceptación por los usuarios. Los resultados experimentales son analizados en detalles en las secciones finales del capítulo.

Primeramente describimos aquellas características generales de los modelos que sustentan estos sistemas, características que son utilizadas en la propuesta. Posteriormente detallamos todos los componentes de la propuesta de explicación, y finalmente mostramos una interfaz de explicación para un SR de películas, donde insertamos nuestra propuesta.

4.1. Características del usuario activo, ítems y vecindario para la explicación

En el capítulo anterior comprobamos la utilidad de determinar la capacidad predictiva de los usuarios para formar parte del vecindario. Donde consideramos que si un usuario era bueno en predecir los votos pasados, entonces lo sería también para predecir los nuevos ítems. Apoyándonos en la idea anterior, nos preguntamos entonces si estos vecinos serían capaces o no de explicar las recomendaciones a los usuarios.

Retomamos los ejemplos del capítulo anterior para mostrar algunas ideas y dar respuesta a la interrogante anterior. En la Tabla 4.1 se tienen los votos de un conjunto de usuarios y el objetivo principal es la recomendación que se le dará a Ann (usuario activo) para el ítem I_7 .

Tabla 4.1: Votos de usuarios. El objetivo es proveer a Ann de una explicación para el ítem $I_t = I_7$.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	...	I_{18}
Sara	5	4	5	3	3	3	2	2	2	2	2	2	...	2
John	4	3	4	2	2	1	2	1	1	1	1	1	...	1
Bill	3	2	3	1	1	1	5	1	3	3	3	3	...	3
Juan	2	1	2	1	2	1	3	3	2	2	2	2	...	2
Katy	3	2	3	2	2	2	4	2	1	1	1	1	...	1
Karl	5			2		1	2	2	2	1	2	2	...	2
Jane		3			2		2	3	3	3	1	1	...	1
Josh			4	2			5	4	2	2	2	2	...	2
Roge		3		2			3	4	1	2	3	3	...	3
Rina	4				2		4	4	4	4	2	2	...	2
Ann	4	3	4	2	2		?							

Imaginemos que ante diferentes condiciones y para predecir el voto del I_7 , dos vecindarios se pudieran formar: por un lado, los usuarios Sara, John, Bill, Juan y Katy formarían el primer vecindario; mientras en el segundo vecindario, encontraríamos a Karl, Jane, Josh, Roge y Rina. Independientemente de la recomendación que reciba Ann, podemos ofrecerle una explicación clásica con los votos de los vecinos, que, con cualquier de los vecindarios anteriores, sería la misma (Tabla 4.2).

Si consideramos la interfaz anterior (la podemos encontrar también en forma de histograma), independientemente de la decisión de Ann de aceptar o no la recomendación, su forma de ver el SR sigue siendo como una caja negra.

Este ejemplo nos confirma que no toda la información es útil para explicar las recomendaciones. Por lo que nos preguntamos, ¿cuál sería la decisión de Ann si conociera la opinión del vecindario hacía sus experiencias pasadas?.

Tabla 4.2: Ejemplo de Interfaz de Explicación para Ann. Voto de los vecinos para el ítem I_7

Votos	Número de vecinos
★	
★★	2
★★★	1
★★★★	1
★★★★★	1

La visualización del comportamiento de los vecindarios anteriores, permitiría hacer notar a Ann en el primer vecindario, valoraciones de todos los usuarios para todos los ítems desde I_1 hasta I_6 ; en el segundo vecindario:

- que para el ítem I_4 , Karl, Josh y Roge valoraron con $2★$, similar a lo votado por ella,
- que para el ítem I_3 , solo Josh le daría una valoración, pues el resto de usuarios no votó el ítem,
- y para el resto de los ítems valorados, solo 2 de los 5 usuarios emitieron una valoración.

Evidentemente si Ann conoce que el vecindario no tiene suficientes razones para ofrecerle una explicación convincente, como es el caso del segundo vecindario, entonces no confiará en la recomendación. La situación será diferente para el primer vecindario, el cual al menos puede proporcionarle diversas evidencias positivas que serán bien recibidos por Ann.

Con ésta idea en mente diseñamos una representación del comportamiento de un vecindario para ilustrar mejor diferentes situaciones (véase la Figura 4.1). Consideramos en lo adelante que Ann sigue siendo el usuario al que se le hacen recomendaciones. En la primera fila, cada celda representa una de las valoraciones de Ann, donde el color está en relación con la escala de la Figura 4.2. Las restantes filas representan el criterio de cada uno de los vecinos respecto a cada uno de los ítems, donde cada celda según la escala anterior tiene un color, que representa la diferencia real entre el voto de Ann y el sugerido por el vecino para dicho ítem (color blanco cuando el usuario no valoró el ítem). En la gráfica podemos ver claramente los casos en los que el valor sugerido hacia un ítem en específico por un vecino es mayor que el voto de Ann y, por tanto, la diferencia es negativa, indicador de que el vecino sobrevaloró con respecto al valor votado por Ann. Los colores en una degradación azul en las celdas de la matriz, indican la sobrevaloración del ítem por parte del vecino, mientras la escala empleada para representar los votos de Ann, representan los casos de subvaloración. El color verde representa acierto total entre los votos.

Con la misma información utilizada para lograr la gráfica de la Figura 4.1, podemos obtener otras representaciones del comportamiento del vecindario, por ejemplo:

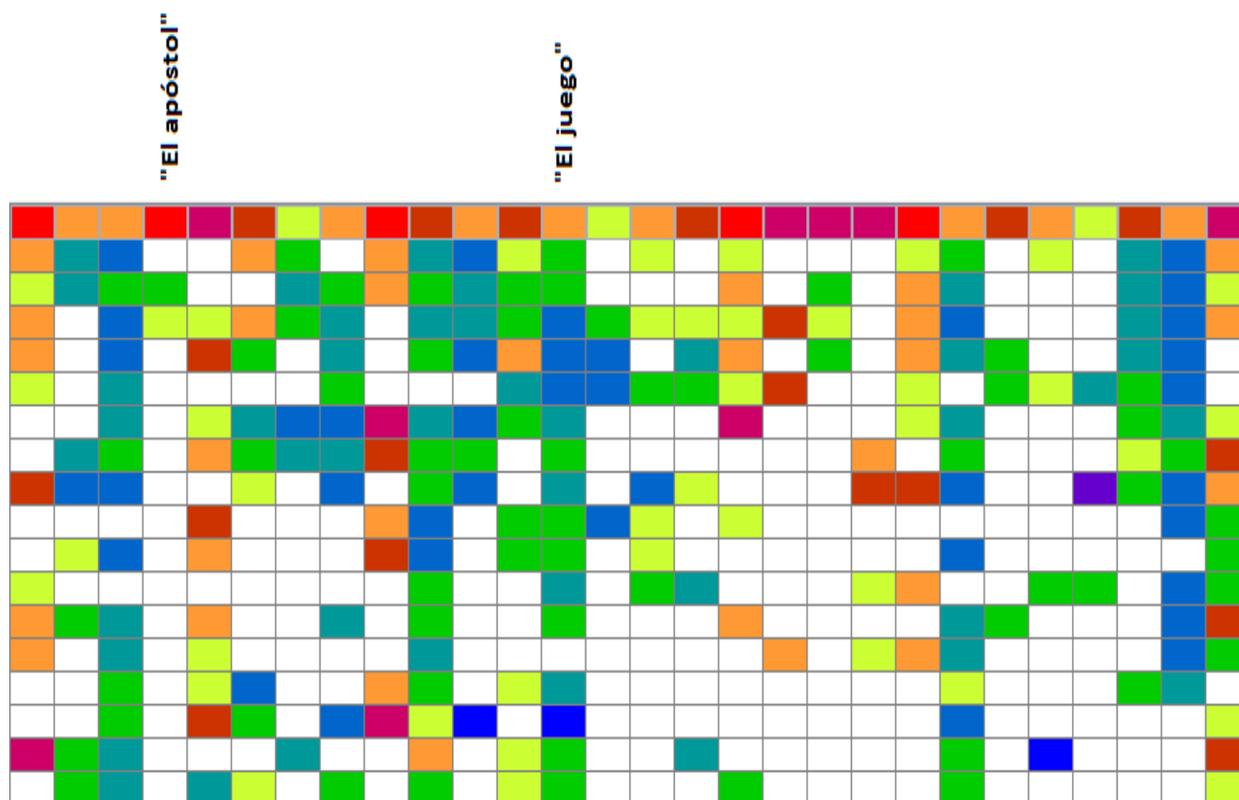


Figura 4.1: Relación entre los votos del usuario activo y el vecindario (ítems en común).



Figura 4.2: Escala de colores para representar: i) el voto del usuario activo y ii) la diferencia de votos.

- mostrar en cada celda el voto del vecino sobre el ítem que corresponde a la columna,
- mostrar la diferencia absoluta entre el voto del usuario y el sugerido por el vecino (Similar a la gráfica que mostramos en el capítulo anterior).

Nosotros seguiremos con la gráfica de la Figura 4.1 para ilustrar comportamientos interesantes por parte de los vecinos seleccionados. Si nos centramos en la cuarta columna, que representa la película “El apóstol (1997)” del director Robert Duvall, podemos observar que solo dos vecinos (segundo y tercero de los vecinos seleccionados para formar el vecindario) valoraron dicha película. Ann, que valoró dicha película con un voto de 5★, sabría que esos

dos vecinos la valoraron con 5★(dado que el color verde intenso indica que la diferencia entre los votos es igual a 0) y 4★ (dado que el color verde claro indica que la diferencia entre los votos es igual a 1), respectivamente.

Para la película “El juego (1997)”, votada por Ann con 2★, columna decimotercera, tenemos que los vecinos la votaron con esa misma puntuación o algo superior en todos los casos. Estos elementos pudieran servir al usuario respecto a algunas películas (películas de referencia), considerando que si el vecindario de forma general coincide en esas películas en especial con la votación que él ha dado, entonces aceptará la recomendación ofrecida. De los 17 vecinos que forman el vecindario, 8 de ellos coinciden con la votación dada para esa película por Ann.

Una situación ideal es que prevalezca en la matriz la tonalidad de verde o cercana a los colores que utilizamos para indicar una diferencia mínima (+1 o -1), esto nos indica que se tiene un buen vecindario, o al menos, con preferencias similares.

Si consideramos la forma de representar el comportamiento del vecindario como el de la Figura 4.1, y además las características de las votaciones de Ann, el vecindario y(o) los ítems, podemos lograr otras gráficas interesantes que comentamos brevemente.

Respecto a las características de los votos de Ann y(o) los ítems podemos considerar lo siguiente:

- Ordenar por fecha. Se ordenan las columnas por la fecha en que Ann votó cada ítem.
- Ordenar por voto. Se ordenan las columnas según el valor del votó que Ann dio a cada ítem. En la Figura 4.3 observamos una representación de este agrupamiento.

Las ventajas y desventajas de esta aproximación para explicar una recomendación serían las siguientes:

- Ventajas: el usuario tendrá rápido acceso a los ítems considerando el criterio por el que se ordena. En el caso de ordenar por fecha, Ann puede ver el comportamiento del vecindario hacía los ítems que votó más recientemente.

Como se observa en la Figura 4.3, Ann puede ver rápidamente el comportamiento del vecindario hacía un voto en específico.

- Desventajas: cuando se tienen muchos ítems, la representación de todos los ítems es más difícil y más compleja su comprensión por parte del usuario activo.

En relación con las características del vecindario consideramos varios criterios de ordenación de los datos, éstos son los siguientes:

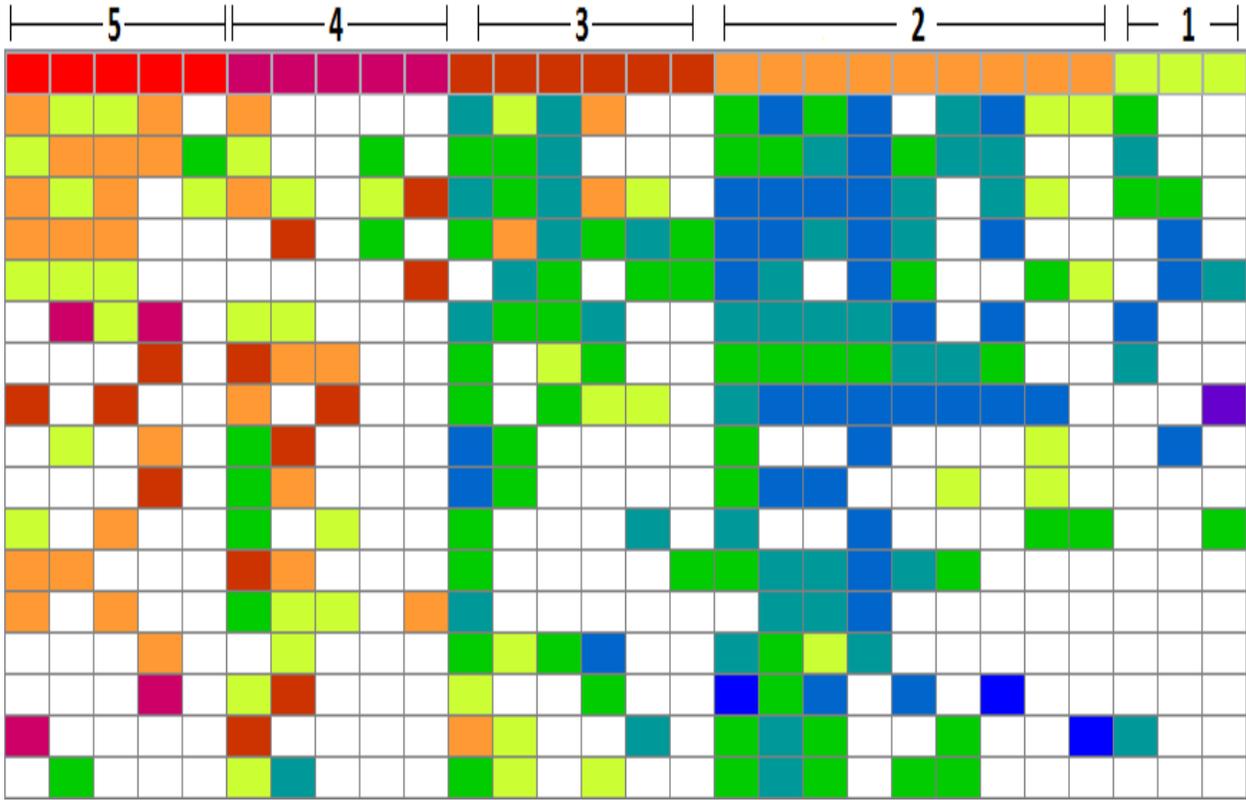


Figura 4.3: Representación del comportamiento del vecindario al ordenar por el voto de Ann.

- Ordenar por error en las votaciones de los vecinos. Similar al criterio anterior, pero para cada vecino y considerando las películas en común con Ann, se computa el error promedio de la votación (Ecuación 4.1). Los vecinos se ordenarían de forma tal que en las filas superiores/inferiores, Ann encontraría usuarios que en promedio suelen estimar de forma más/menos similar a sus votos previos:

$$e = \frac{\sum_{i \in N} |r_{a,i} - \hat{r}_{v,i}|}{n}. \quad (4.1)$$

- Ordenar por predicción del vecindario. Se ordena cada ítem según el voto con el que el vecindario lo recomendaría a Ann.
- Ordenar por error en las predicciones de los ítems. Se calcula primeramente para cada ítem el error promedio de votación de los vecinos (Ecuación 4.2) que han votado dicho ítem, luego se ordenan según los valores que se obtienen:

$$h(i) = \frac{\sum_{V \in N} |r_{a,i} - \hat{r}_{V,i}|}{n}. \quad (4.2)$$

En la Figura 4.4, donde los ítems están primeramente agrupados por el voto que dio Ann, y luego, dentro de cada grupo, ordenados por el criterio anterior (Predicción del vecindario), podemos ver para la película “El juego” (valorada con 2★), el voto que darían los 17 vecinos: 8 votarían con 2★, 4 con 3★, 3 con 4★ y 1 con 5★. Con estos votos tenemos que el error promedio en las predicciones de los vecinos hacía esta película es:

$$h = (8 * |2 - 2| + 4 * |2 - 3| + 3 * |2 - 4| + 1 * |2 - 5|) / 16 = 13 / 16 = 0,8125$$

Mientras, para “El apóstol”, donde los votos de los dos vecinos serían de 5★ y 4★, tendríamos, $h = ((5 - 5) + (5 - 4)) / 2 = 0,5$.

- Ordenar por similitud en las predicciones respecto al ítem objetivo. Partimos de tener dos vectores de votos por cada ítem, el primero y común para todos los ítems, con los votos sugeridos por los vecinos para el ítem que se recomienda, el segundo, con los votos que los vecinos sugieren hacía el ítem del perfil de Ann que se analiza. Tomando estos vectores, se computa una medida que nos permite determinar que tan similar los vecinos en conjunto, sugieren cada uno de los ítems previos a como sugieren el ítem actual. Se computa la similitud en la forma de sugerir dos películas diferentes de la forma $d(i, t) \propto \sum_{v \in N} |sg(a, v, i) - sg(a, v, t)|$.

Al usar los datos de la Tabla 4.1 como ejemplo, para la recomendación del ítem I_7 a Ann, sus vecinos Sara, John, Bill, Juan y Katy le sugieren 2★, 2★, 5★, 3★ y 4★ respectivamente. Para los ítems desde I_1 hasta el I_5 se calcularía la similitud $d(i, t)$, donde $d(I_1, I_7) = 3 + 2 + 2 + 1 + 1 = 9$, $d(I_2, I_7) = 10$, $d(I_3, I_7) = 9$, $d(I_4, I_7) = 9$ y $d(I_5, I_7) = 8$, siendo por tanto el ítem I_5 más similar que el resto de los ítems.

En la Figura 4.4, tenemos a la izquierda una columna con los votos que cada uno de los vecinos daría a la película que se recomienda. Siguiendo con la escala de colores que hemos empleado, tendríamos el voto de 2★, 2★, 4★, 2★, ..., 2★, por cada uno de los vecinos.

Para la película “El apóstol”, tenemos el voto de los vecinos en el siguiente vector, [5,4]. Al computar la medida obtenemos, $d = |5 - 2| + |4 - 4| = 3$

Una vez se tiene esta medida para cada uno de los ítems, logramos tener un orden que nos permite observar las películas en las cuales el patrón de voto de los vecinos, es más similar al que utilizan en la película objetivo.

Las ventajas y desventajas de esta nueva aproximación son las siguientes:

- Ventajas: podemos determinar la calidad de cada miembro del vecindario.

Comprobamos cómo se comporta en media el vecindario hacía un ítem en específico.

Logramos una medida de similitud entre las predicciones del vecindario.

- Desventajas: cuando se tienen muchos ítems, la representación de todos los ítems es más difícil y más compleja su comprensión por parte del usuario activo.

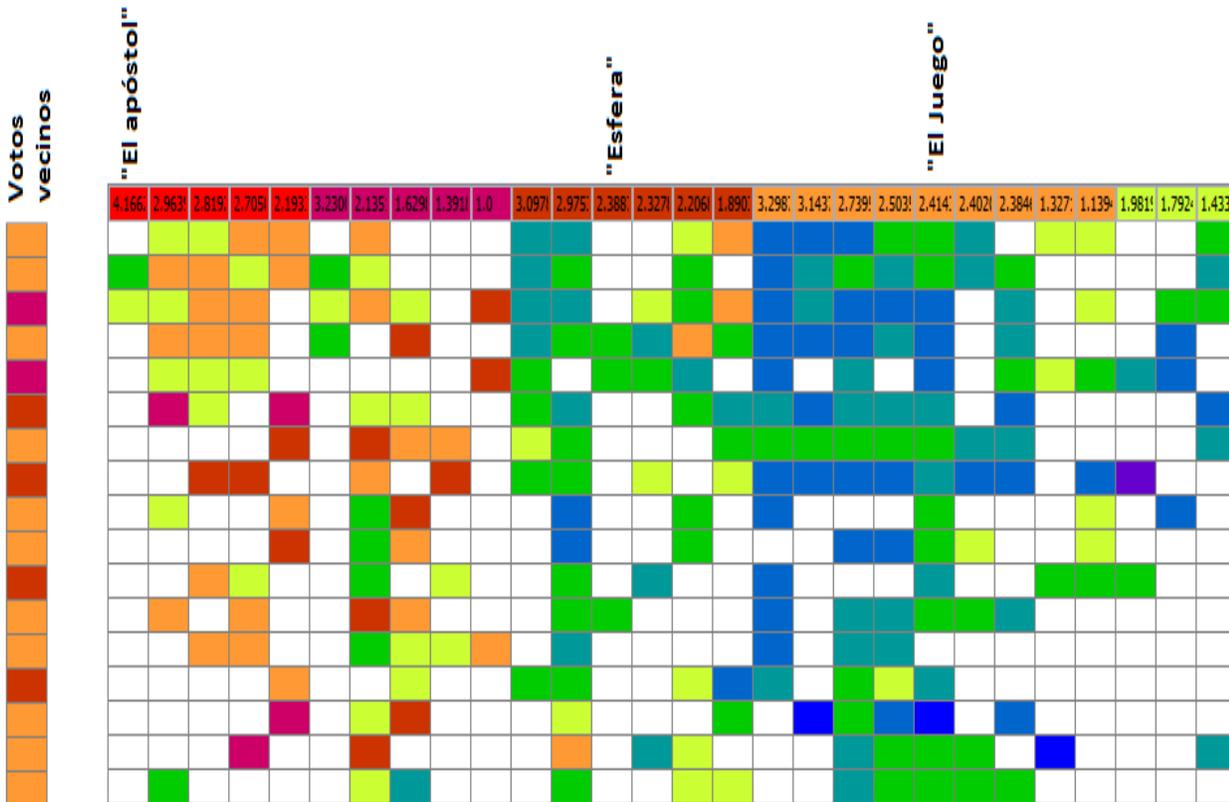


Figura 4.4: Representación del comportamiento del vecindario al agrupar los votos de Ann y ordenar dentro de cada grupo por la predicción del vecindario.

Los criterios analizados anteriormente pueden ser mezclados y lograr representaciones que permitirían comprobar a Ann otros elementos, por ejemplo, cómo es el comportamiento de los vecinos respecto a las películas más recientes, etc.

Si analizamos las ventajas y desventajas de las representaciones anteriores, encontramos como desventaja común, que la utilización de estas representaciones es más difícil cuanto mayor sea el número de ítems valorados previamente. No obstante, la información que nos brinda cada una de ellas, la consideramos muy útil para explicar las recomendaciones.

Para mostrar otra forma visual y destacar información que puede resultar de interés a un usuario, nos centramos en las dos últimas características que analizamos del vecindario. En la Figura 4.5 encontramos una representación diferente, en este caso se ordena verticalmente por el error en las predicciones de los ítems, donde destacamos la ubicación de la película “Esfera (1998)”, votada originalmente por Ann con un voto de 3★ y recomendada por el vecindario con un voto de 2★. Esta película fue valorada por 3 vecinos del vecindario seleccionado, pero

La lista de Schindler (1993)

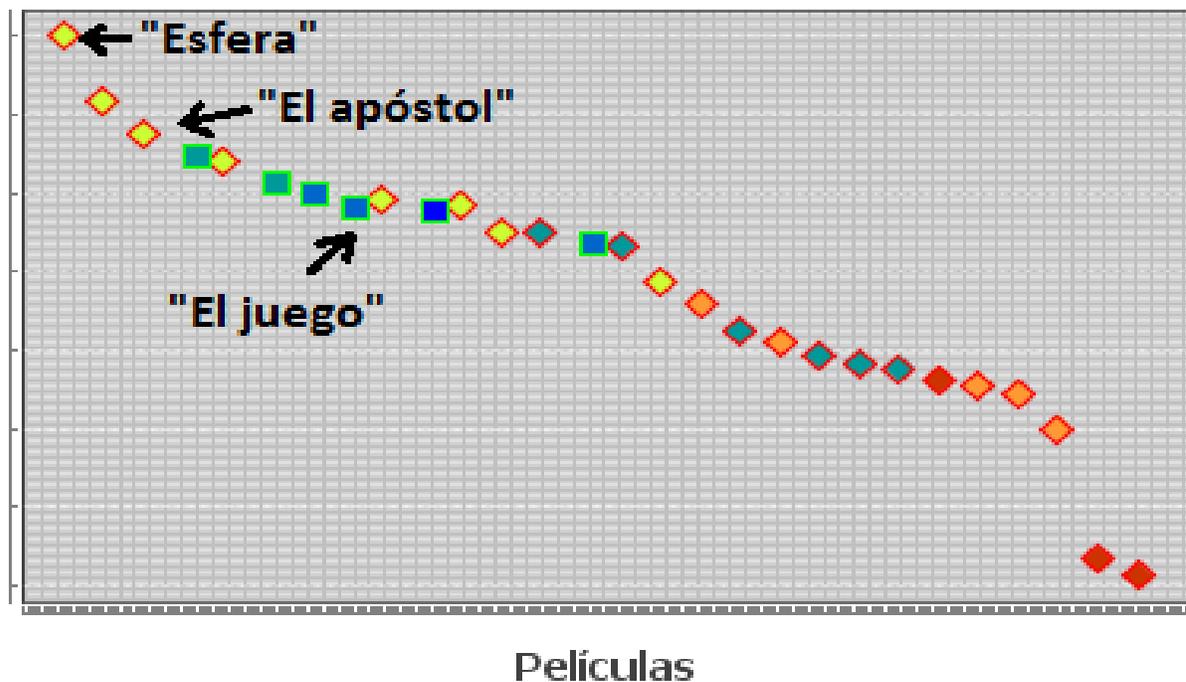


Figura 4.5: Representación al ordenar según el error de las predicciones de los ítems por el vecindario.

su posición está dada en que dichos vecinos coinciden completamente con el voto original de Ann (Figura 4.4).

En la Figura 4.6 mostramos los ítems ordenados verticalmente según la similitud en las predicción respecto a la película que se recomienda, “La lista de Schindler (1993)”. Mientras la película “Esfera ” en la Figura 4.5 es la primera que tenemos, en este nuevo gráfico encontramos primero a la película “El apóstol ”. En la gráfica de la Figura 4.6 observamos que es “El apóstol ”, la película donde los vecinos votan de forma más similar a la que recomiendan.

Cada una de los ítems en los gráficos anteriores, están representados por una figura geométrica, diamante o cuadrado, lo que representa fallo o acierto del vecindario hacía el voto dado por Ann, en la próxima sección, al describir la propuesta final se detalla esta representación. El color de cada figura está en correspondencia con la escala de colores empleada en las representaciones anteriores para representar la diferencia entre el voto del usuario activo y las predicciones de los vecinos.

Dentro de las ventajas y desventajas de estas aproximaciones encontramos las siguientes:

- Ventajas: podemos observar la calidad o el criterio del vecindario hacía cada uno de los ítems.

La lista de Schindler (1993)



Figura 4.6: Representación al ordenar según la similitud en las predicciones respecto al ítem objetivo.

A diferencia de las aproximaciones anteriores, en estas se pueden mostrar un mayor número de ítems.

- Desventajas: solo es posible representar un solo criterio verticalmente.

4.2. Propuesta de explicación para sistemas de recomendación basados en filtrado colaborativo

En la sección anterior nos apoyamos en los datos que se tienen en un proceso de recomendación colaborativo y describimos la información que se puede extraer al estudiarlos. En lo adelante formalizamos nuestra propuesta de explicación considerando criterios a los que los usuarios prestan especial atención, y que además, ya comentamos al finalizar la sección previa.

En un esquema basado en vecindad, tenemos un usuario al que realizamos una recomendación sobre un ítem específico. Esta sugerencia la computamos seleccionando el vecindario que dado un criterio determinado, mejor recomendación puede dar al usuario. En el criterio para seleccionar los vecinos, se considera la valoración sobre los ítems previos del usuario

[50, 52]. En este esquema resaltan los siguientes elementos:

- vecindario seleccionado para predecir el interés del usuario activo sobre un ítem específico,
- usuario sobre el que se realiza la recomendación,
- ítem que recomendamos,
- ítems votados previamente por el usuario activo y por los miembros del vecindario.

Imaginemos que en el paso posterior a la recomendación, el usuario solicita argumentos del “porqué” de la recomendación recibida. Apoyándonos en la idea de que un usuario se basa tanto en sus experiencias previas como en sus amigos para tomar decisiones, en nuestra propuesta de explicación, consideramos estos elementos para mostrar al usuario la opinión del vecindario, en aquellos ítems valorados previamente. Nuestra hipótesis es que si un usuario fue seleccionado como vecino, entonces será capaz de explicarle al usuario activo las sugerencias que este reciba.

La propuesta destaca el criterio de la vecindad, considerando la relación entre los votos pasados y las necesidades presentes. Se muestra una representación conceptual en la Figura 4.7. En la Figura 4.7 encontramos al usuario activo en el centro, recibiendo una recomendación de cada uno de los vecinos hacía un ítem objetivo. Podemos observar como el usuario puede comprobar además la opinión de cada uno de los vecinos respecto a cada uno de los ítems que el valoró previamente. Considerando estos criterios, se muestra al usuario lo que sería la opinión del vecindario respecto a sus experiencias previas.

En el caso de Ann (Tabla 4.1), a ella le gustaría conocer cuál es el criterio de sus vecinos respecto, por ejemplo, al ítem I_3 . Si los vecinos fuesen Karl, Jane, Josh, Roge y Rina, solo Josh le podría dar un criterio al respecto, pues el resto no lo valoró. Si de cinco vecinos, solo uno puede comentar sobre un ítem, Ann pudiera no confiar en la recomendación que le han dado sus amigos. Sin embargo, si los vecinos fueran Sarah, John, Bill, Juan y Katy, todos ellos pudieran darle aunque fuesen criterios variados a Ann acerca, no solo de I_3 , sino también de todos los valorados por ella.

La información que obtuvimos en las representaciones de las Figuras 4.5 y 4.6 (comportamiento del vecindario hacía cada experiencia previa (Figura 4.5) y similitud entre la predicción de cada experiencia previa y el ítem objetivo (Figura 4.6)), nos llevó a pensar en mezclar estas dos representaciones de forma tal que se muestre al usuario un comportamiento más general del vecindario. Con esta explicación el usuario podría evaluar a sus vecinos y determinar si está de acuerdo o difiere con ellos en cuanto a la recomendación recibida.

En la Figura 4.8 mostramos nuestra propuesta. Este esquema lo logramos al mezclar las gráficas de las Figuras 4.5 y 4.6. El eje horizontal muestra la relación que logramos en la

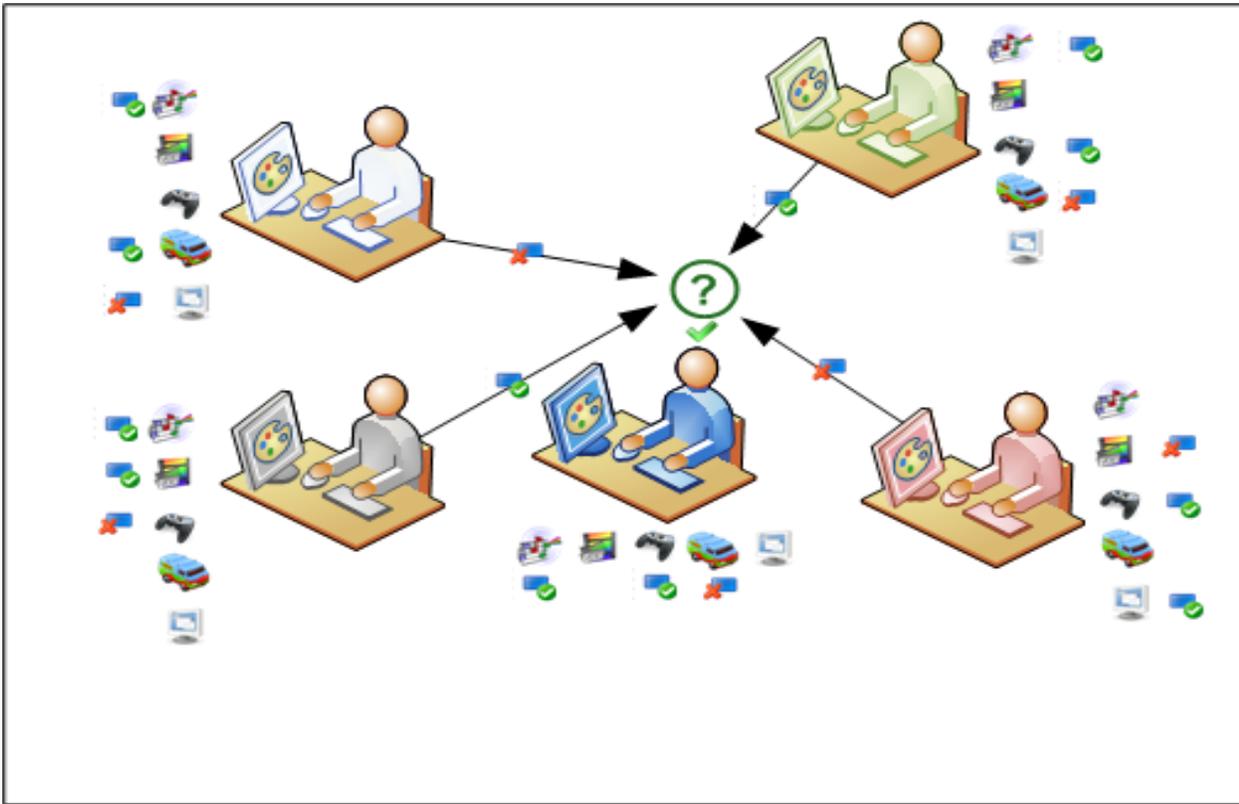


Figura 4.7: Criterio del vecindario acerca de los votos previos del usuario activo.

Figura 4.6 en relación a la película que se recomienda, mientras el eje vertical muestra el comportamiento del vecindario hacia cada experiencia previa (Figura 4.5). Se destacan la ubicación de las películas “El apóstol” y “Esfera”. En el caso de la película “El Juego”, la encontraremos justo debajo de la película “El apóstol”.

Se pueden encontrar los ítems que guardan mejor relación con el indicador del eje, en el caso horizontal más a la izquierda, y en caso vertical en posiciones superiores. La relación anterior fue determinada asumiendo el criterio de como el ser humano capta espacialmente mejor la información. El cerebro naturalmente prioriza los órdenes “arriba-abajo” e “izquierda - derecha”. En la gráfica de la Figura 4.8, se tienen 4 cuadrantes bien delimitados. Cada cuadrante refleja una mejor/peor relación de los ítems respecto al criterio que representa cada eje.

Para cada ítem se muestra la calidad de predicción, empleando cuadrados (\square) o diamantes (\diamond) para representar el acierto o no del vecindario. Cada una de estas figuras tiene un color que representa el voto dado por el usuario. Dicho color está en correspondencia con los colores utilizados hasta el momento en las representaciones anteriores.

En la Figura 4.9 se muestra una propuesta similar a la de la Figura 4.8, pero donde el color de cada figura representa la diferencia del voto que sugiere el vecindario con el voto

La lista de Schindler (1993)

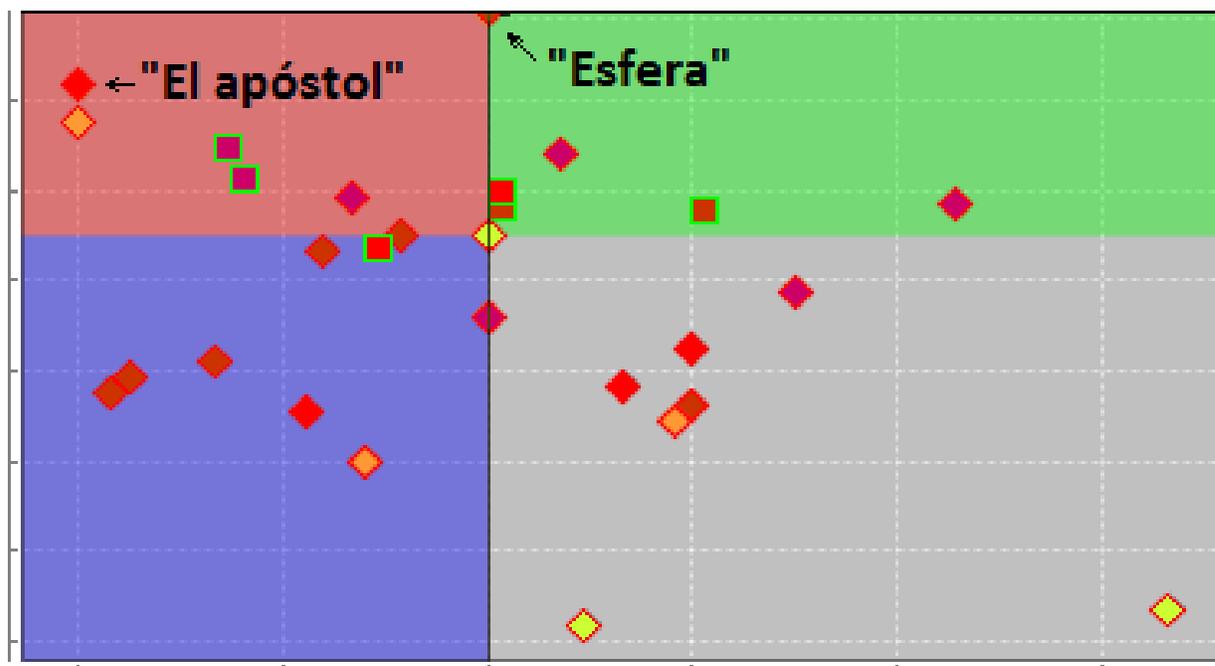


Figura 4.8: Propuesta de explicación. Error promedio de votación vs Similitud en las predicciones.

del usuario. Aunque la idea de la Figura 4.9 resulta interesante, consideramos más intuitivo utilizar el voto dado por el usuario para colorear cada figura.

Por otra parte, nuestra propuesta permite capturar ciertas relaciones que en otras propuestas no era tan sencillo, e incluso permite al usuario validar criterios como:

- Si la película “El Padrino” le gusta al vecindario, seguro que la que me recomiendan me gustará.
- Qué opinión tiene este vecindario sobre la película “El Padrino”.
- Cuáles de las películas que he visto, son valoradas en conjunto por el vecindario igual que esta que me recomiendan.

Este último criterio lo podemos observar en la Figura 4.10, donde tenemos una explicación para la película “Desperado”. El usuario puede notar que, de todas las películas valoradas previamente, el patrón de votos del vecindario para la película que recomiendan, es más similar con el que utilizan para la película “Entrevista con el vampiro” y más diferente con el de “El imperio contraataca”.

En la Figura 4.11 mostramos una explicación para la recomendación de la película “Scarface”, en la que podemos observar a la izquierda, tres películas rodeadas mediante líneas

La lista de Schindler (1993)

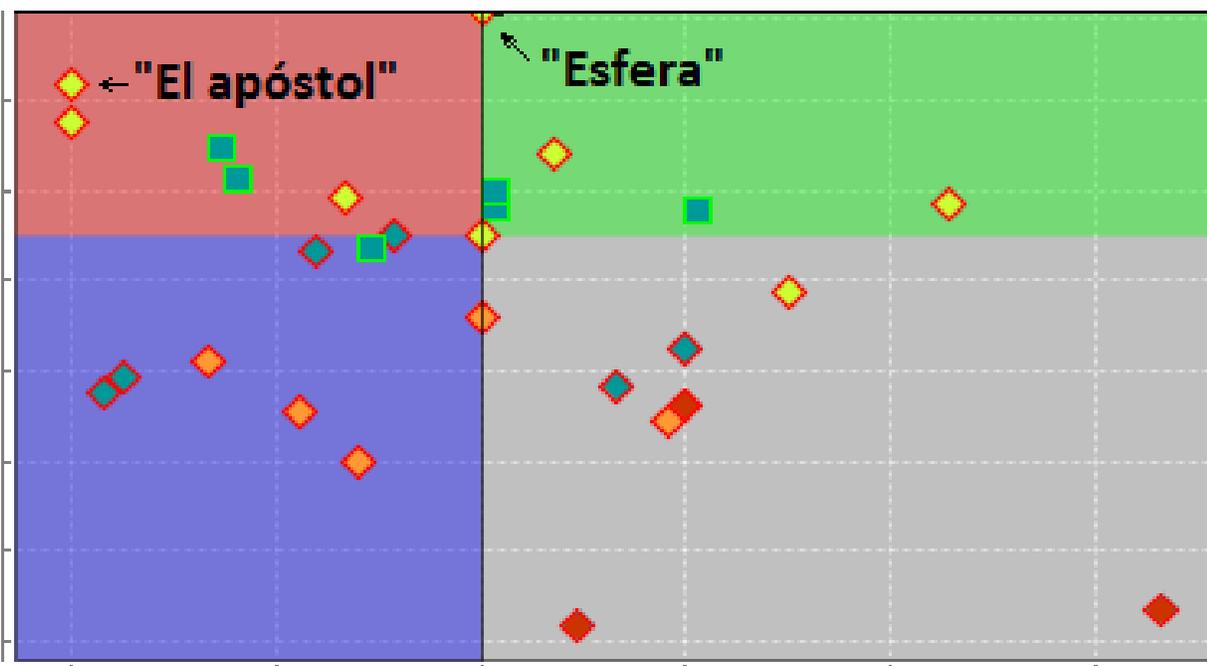


Figura 4.9: Propuesta de explicación. Se utiliza la diferencia del voto sugerido por el vecindario y el voto dado por el usuario para colorear cada figura.

discontinuas. Estas películas, que son predichas por el vecindario de forma muy similar a la que recomiendan, corresponden a la trilogía de “El Padrino”. El usuario puede encontrar situaciones como la anterior en la que ítems muy relacionados semánticamente estén ubicados muy cercanos entre sí en el gráfico.

Las ventajas y desventajas de nuestra propuesta de explicación son las siguientes:

- Ventajas: podemos observar la calidad y el criterio del vecindario hacia cada uno de los ítems valorados previamente.

Se mezclan varios criterios en la propuesta, brindando información útil al usuario activo.

- Desventajas: inicialmente puede resultar una explicación compleja por toda la información que el usuario activo recibe.

Una vez comentada en general nuestra propuesta de explicación y los datos utilizados en ella, damos paso a describir una interfaz de explicación para un SR de películas que la implementa.

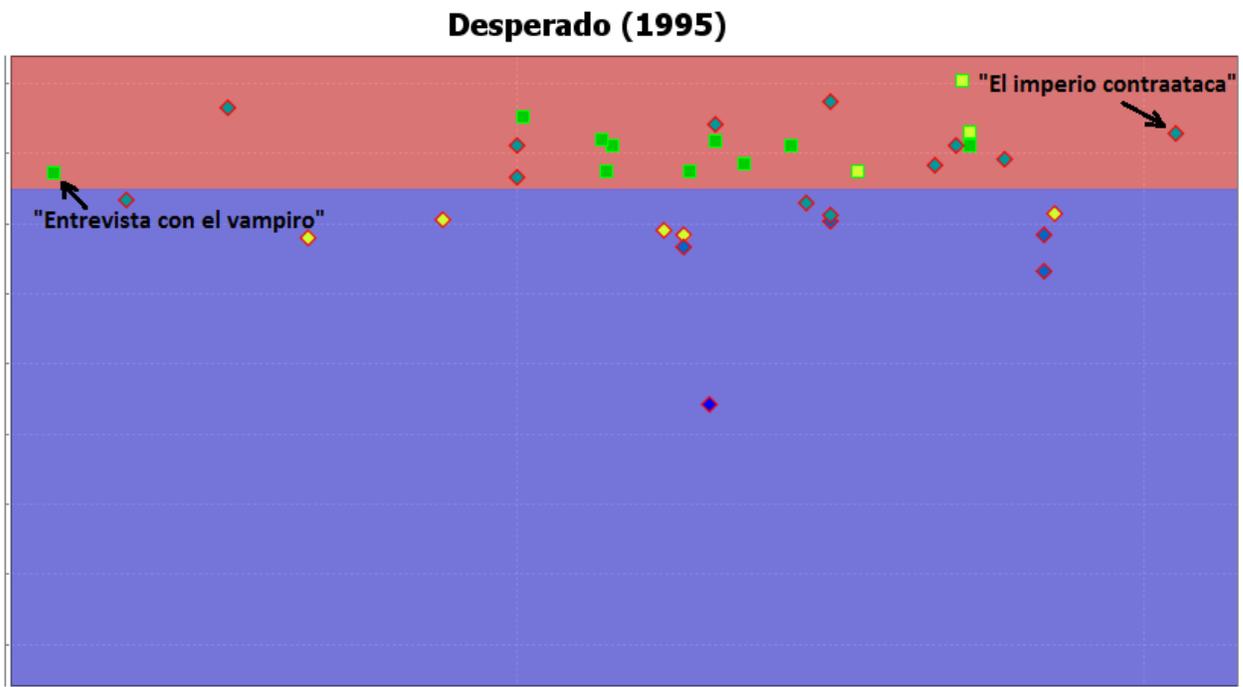


Figura 4.10: Ejemplo de explicación para la película “Desperado”.

4.3. Interfaz de explicación para un SR de películas

En esta sección se describe en detalle la interfaz de explicación diseñada para un SR de películas. Nos basamos en nuestra propuesta de explicación, que como se ha visto hasta ahora, puede ayudar al usuario a evaluar la calidad de la recomendación de acuerdo a sus preferencias. En la Figura 4.12 observamos la interfaz de explicación que proponemos para un SR de recomendación de películas.

La interfaz cuenta con cinco componentes o partes, A, B, C, D y E. La parte A muestra una interfaz clásica de explicación (histograma con los votos del vecindario), donde se utiliza una barra para cada posible categoría (1★ a 5★) más una barra adicional (“?”) que representa que se ha tenido problemas cuando se intentó encontrar un mínimo conjunto de vecinos que hubiesen evaluado el ítem para el que se realiza la recomendación (se considera que esta información es importante para el usuario activo). Suele suceder que para ítems poco valorados, no siempre se encuentran suficientes vecinos para completar el vecindario, por lo que se considera que el número de vecinos que faltarían por completar el vecindario es un dato valioso y que debe ser conocido por el usuario. En el caso de la interfaz de la Figura 4.12, la explicación del histograma se enfocaría en expresar “los usuarios del vecindario sugieren 3 para la película, porque la valoración hacía dicha película ha sido ...”. Si consideramos que se seleccionan de 20 vecinos, en el anterior histograma se ha podido encontrar un vecindario con la totalidad de los miembros.

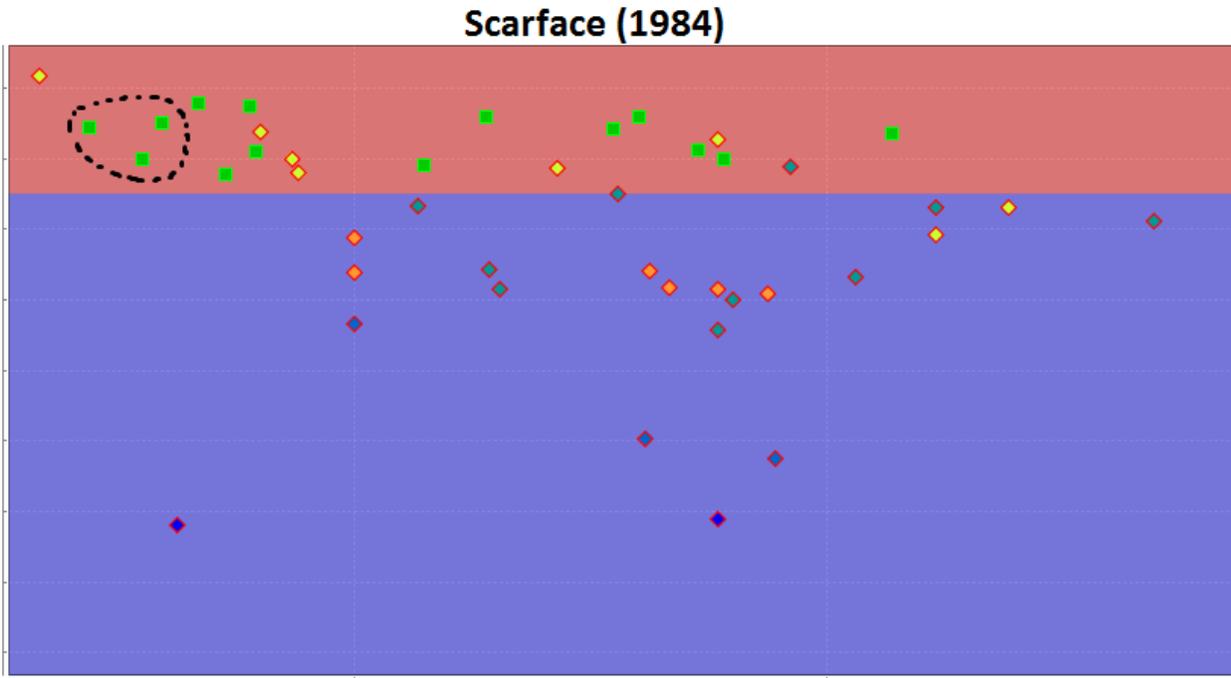


Figura 4.11: Ejemplo de explicación para la película “Scarface”.

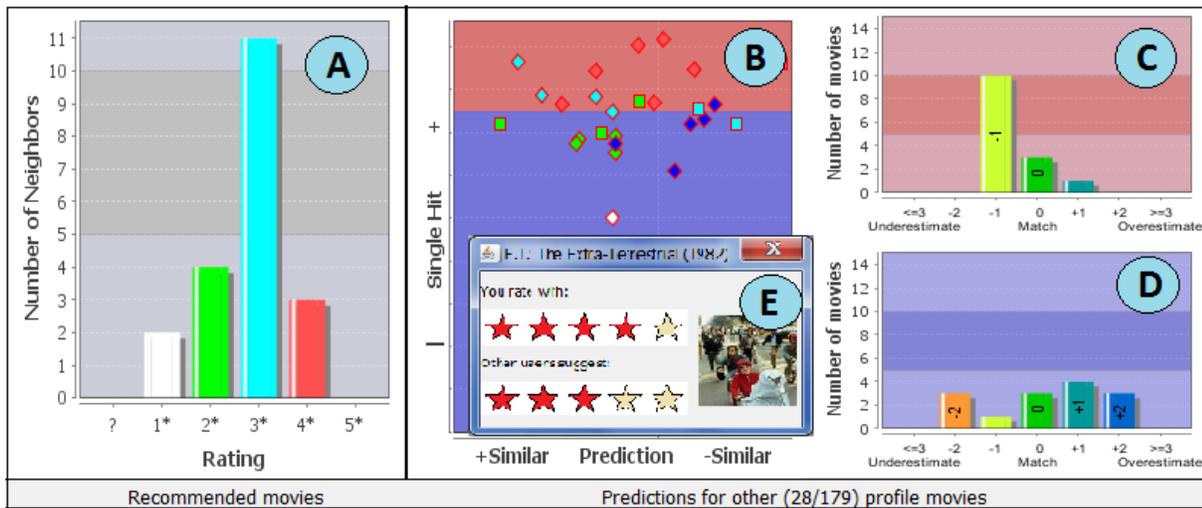


Figura 4.12: Propuesta de interfaz de explicación para SR de películas.

Aunque la interfaz con el histograma explica de alguna manera cómo el voto estimado se computa, un usuario se puede preguntar: ¿por qué debo creer en la sugerencia de 3★ recibida por el sistema? En caso de auxiliarse en un amigo, el creería o no la recomendación porque usualmente conoce sus preferencias sobre un conjunto de ítems evaluados en común. Al incluir nuestra propuesta en esta interfaz estamos considerando la idea anterior. Por lo tanto, el problema se reduce a si incluimos nuestra propuesta sin ningún cambio o le hacemos

algunos ajustes.

Dentro de la Figura 4.12, se encuentran las partes B, C y D que muestra la información del comportamiento del vecindario. Debemos comentar que el color de relleno de las figuras geométricas en la parte B, está en correspondencia al voto del usuario, y al mismo tiempo, a los colores utilizados para representar cada voto en el histograma de la parte A. En las partes C y D se emplean colores diferentes para representar el error cometido en las predicciones. Al posicionarnos sobre una de las figuras geométricas de la parte B, aparece una ventana (parte E) con información que describe dicho ítem, en el caso que se muestra se tiene que para la película “E.T 1982”, el usuario la valoró con 4★ y el vecindario le sugiere 3★. De esta forma el usuario podría rápidamente ver cuál sería el comportamiento del vecindario hacia sus valoraciones previas.

En la parte B de la interfaz se consideran las siguientes configuraciones:

- No todas las películas previas se muestran. Por una lado, un usuario puede valorar muchas películas en el pasado y mostrarlas todas restaría claridad en la interfaz, y por otro lado, nosotros no esperamos que un vecindario determinado se comporte bien en predecir todos los ítems previos. En la experimentación que hemos realizado, se requiere que para que un ítem forme parte de la explicación, éste debe ser evaluado por al menos un % de los individuos del vecindario. Obviamente, el número dependerá del algoritmo empleado para seleccionar el vecindario.
- El gráfico se divide en dos cuadrantes o espacios, en los cuales se encuentran aquellos ítems que tienen cierta relación (mejor/peor) con las medidas que se analizan en cada eje. El usuario encuentra en el cuadrante superior aquellos ítems donde los vecinos tienen un alto acierto individual. En el cuadrante inferior serán encontradas aquellas películas en donde los vecinos no suelen tener un criterio homogéneo o no suelen ponerse de acuerdo. A la izquierda de cada cuadrante, se podrán encontrar los ítems que son sugeridos de forma muy similar al ítem que se analiza.

Este parámetro de configuración es un tema sobre el que se pretenden realizar futuros estudios, pero para esta interfaz, consideramos que estarán en la parte superior aquellos ítems en los cuales el error promedio del acierto individual del vecindario sea menor que 1.

Las partes C y D muestran histogramas que representan el error en la predicción de los ítems valorados previamente en los cuadrantes superior e inferior respectivamente de la parte B. Para el ejemplo que se visualiza, en el cuadrante superior, de las 14 películas que dan soporte a dicha figura, 10 son sugeridas por los vecinos con un voto por debajo del voto original dado por el usuario, (Original-Sugerido, 5-4, 4-3, 3-2 o 2-1), una película es evaluada con un voto por encima, y en tres casos se acierta completamente.

En esta propuesta decidimos utilizar los histogramas para visualizar de forma consolidada el comportamiento del vecindario en las regiones delimitadas de la parte B. De igual forma se pudiera emplear un histograma en general de ambas partes o representar en los histogramas, en lugar del error en las predicciones, las posibles predicciones del vecindario por cada valor de voto posible, es decir, para las películas votadas por 3★, el usuario vería cuantas son predichas con 1★, 2★, ..., y 5★. En general estos histogramas pueden ser útiles para consolidar cierta información que al usuario le resulte útil.

Retomando los histogramas de las partes C y D de la Figura 4.12, y al mismo tiempo la parte B, un usuario encontraría información acerca de un amplio número de situaciones, incluidas aquellas donde el sistema acierta con sus votos originales. Otras situaciones pueden sugerir que el voto que se recomienda no es el adecuado. Por ejemplo, considerando la parte C, el usuario puede entender que los vecinos tienden a subestimar los votos que dan a los ítems que él valoró y, por tanto, pensar en corregir la predicción dada hacía arriba con un voto de 4★ en lugar de 3★. Puede encontrarse una situación similar en el gráfico de la izquierda de la Figura 4.13. La Figura 4.13 muestra tres posibles histogramas que un usuario pudiera observar. En el centro se tiene una situación donde los vecinos aciertan en la mayoría de los votos. Mientras en la figura de la derecha se observa claramente una situación de sobreestimación, donde los vecinos tienden a valorar muchas películas por encima del valor original del usuario.

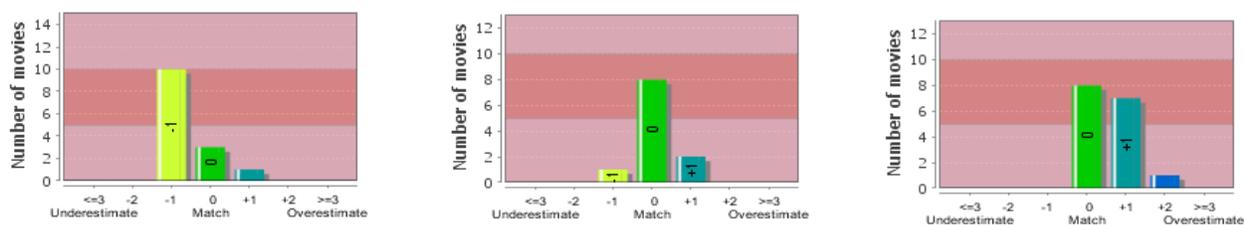


Figura 4.13: Histogramas de predicción de errores. Subestimación (Izquierda), Acierto (Centro) y Sobreestimación (Derecha).

4.3.1. Ejemplos de uso de la interfaz de explicación

Hasta aquí hemos descrito cada una de las partes de la interfaz de explicación que proponemos para un SR de películas. No obstante, y para comprenderla mejor, expondremos nuestro análisis general de algunos ejemplos.

En la Figura 4.14 encontramos una explicación para una película que se recomienda ver con un voto de 5★. El histograma de votos del vecindario refleja una muy buena opinión en general de los vecinos, 13 de ellos la valoran con 5★, 5 con 4★ y 2 con 3★. En el centro

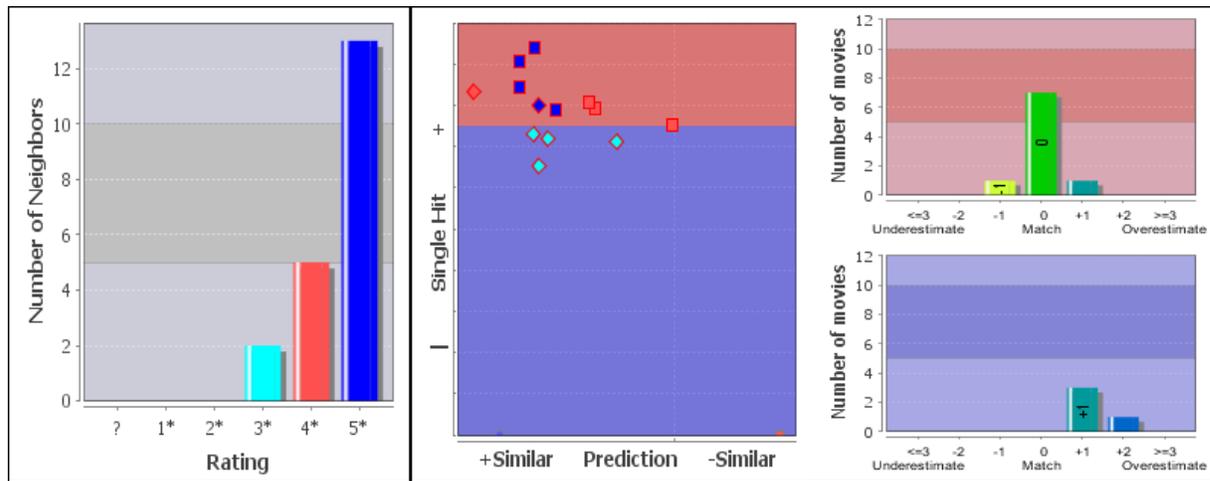


Figura 4.14: Ejemplo de interfaz de explicación para SR de películas. Caso donde la recomendación se aceptaría.

de la figura encontramos la opinión del vecindario hacía 13 películas de las valoradas que encontramos en el perfil del usuario, y donde podemos destacar:

- El vecindario acierta totalmente en 7 películas, todas votadas con 4★ y 5★ por el usuario.
- La película más similar (más a la izquierda) en la forma de votarla respecto a la que recomiendan, es “Indiana Jones en busca del arca perdida”. Esta información da al usuario la idea de cuan similar, al menos en la forma de votarla, es la película que le recomiendan respecto a las que él ya ha visto.
- En la película que en la que suelen estar más de acuerdo (más arriba) con respecto a la votación del usuario es “El Padrino”. Ésta perfectamente pudiera ser una de las películas de referencia del usuario, en donde la opinión del vecindario tuviera un fuerte peso en su decisión. Si un usuario observa que el vecindario tiene una opinión muy diferente a la suya en una película determinada, puede ser una razón de rechazar la recomendación.

Toda la información anterior de la opinión del vecindario hacía las películas valoradas previamente, la encontramos consolidada en los histogramas de la derecha. Lo primero es el número de películas que encontramos en cada parte del gráfico del centro de la figura, 4 películas en la parte inferior y 9 en la parte superior. Luego, de estas 9 películas en la parte superior, donde los vecinos suelen estar de acuerdo con el voto del usuario, en 7 de ellas además se recomienda el mismo voto del usuario. Al analizar estos histogramas, observamos que no reflejan una tendencia del vecindario a sobreestimar o subestimar los votos del usuario.

Nuestro análisis de la explicación de la Figura 4.14 en su totalidad, nos lleva a aceptar la recomendación recibida.

En la Figura 4.15 encontramos otro ejemplo de explicación, pero para una película que se recomienda con un valor de 3★.

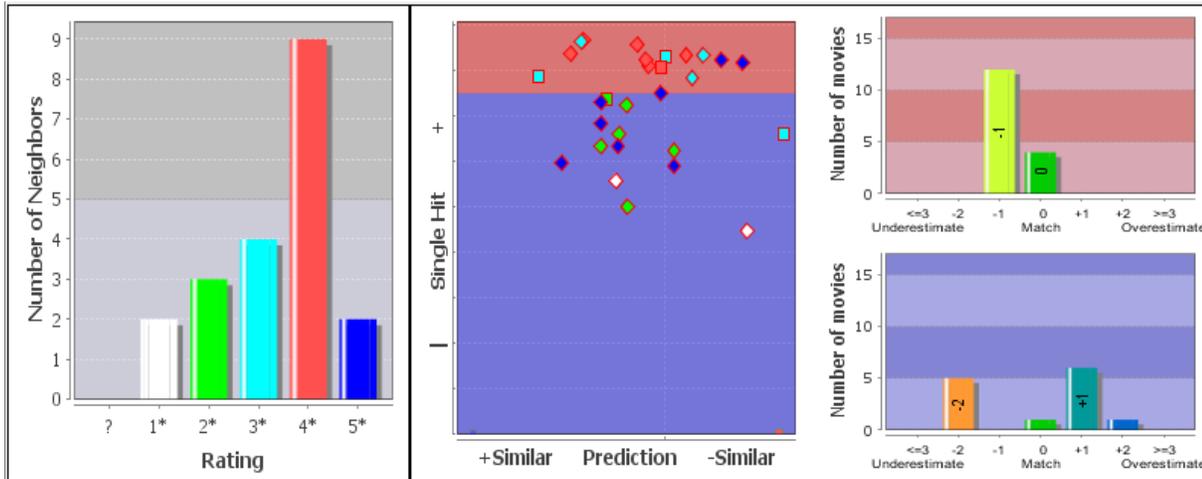


Figura 4.15: Ejemplo de interfaz de explicación para SR de películas. Caso donde la recomendación se pudiera modificar.

Lo primero que observamos al analizar el histograma de la izquierda, es que la opinión de los vecinos está distribuida en los posibles valores, aunque es el voto de 4 en el que un mayor número de vecinos coinciden. Al analizar en el gráfico central la opinión del vecindario sobre las películas valoradas previamente, podemos observar:

- Que la forma en que votan la película que recomiendan, es más similar a la forma en que votaron “Duro de Matar (1995)”, la cual además recomiendan con el mismo voto del usuario (3★).
- Que en total solamente hay 5 películas en las que los vecinos aciertan con el voto del usuario. Lo que nos hace pensar que el vecindario tiene una opinión no tan similar a la del usuario.

Pudiéramos analizar otras películas y su ubicación en este gráfico, pero nos centraremos en estas dos observaciones y en los histogramas de la derecha para destacar como el vecindario tiene una cierta tendencia a subestimar los votos del usuario. Si analizamos de los histogramas de la derecha, el superior, observamos como de 16 películas, en 12 de ellas, el vecindario a pesar de estar individualmente de acuerdo con el voto del usuario (están en la parte superior de la figura central), las valora colectivamente con un voto inferior. El histograma inferior nos refleja una tendencia tanto de subestimación como de sobreestimación.

Al analizar por un lado el histograma de los votos del vecindario, donde dada la distribución, bien pudieran los vecinos haber recomendado 4★, y por otro lado, la información en

conjunto del gráfico del centro y los histogramas de la derecha, donde se observa una cierta tendencia a subestimar los votos del usuario, consideramos que el vecindario se equivoca al recomendar el voto de 3★ para la película. Los elementos que hemos visto anteriormente, nos llevan a corregir el voto a 4★.

Debemos mencionar, que en los ejemplos que hemos utilizado para describir la interfaz de explicación, exponemos nuestra opinión al interpretar cada una de las gráficas, pero esta no será la interpretación de todos los usuarios. Al interpretar cada gráfico influyen elementos subjetivos por parte de cada usuario en el momento de tomar una decisión.

4.4. Experimentación

Una vez descrita la interfaz gráfica de explicación para un SR de películas, resta evaluar el impacto de esta en los usuarios. En lo adelante se establecen los objetivos y parámetros de la evaluación y se muestran los resultados que obtuvimos con un grupo de usuarios reales.

- **Objetivos de la experimentación:** Como objetivo general de la evaluación se planteó medir el impacto de la interfaz de explicación en un SR de películas respecto a la persuasión, eficacia y la satisfacción en los usuarios.

Teniendo como objetivos más específicos:

- Medir la capacidad que brinda la explicación al usuario, de evaluar un ítem recomendado luego de consultarla.
 - Medir la capacidad de persuasión que tiene la interfaz de explicación.
 - Evaluar la satisfacción de los usuarios con la interfaz de explicación.
- **Medidas de evaluación:** En el primero de los objetivos comprobaremos el desempeño de cada una de las interfaces similar a [10, 118], registrando la valoración del usuario luego de su consulta y comparándola con la votación original:

$$e = |r_o - r_x|, \quad (4.3)$$

donde para cada interfaz que el usuario consulta, r_o representa el voto original dado por el usuario y r_x es el voto obtenido para el ítem que se analiza en la interfaz de explicación. Empleamos finalmente la MAE como medida general para evaluar el desempeño general de cada una de las interfaces.

Para el segundo de los objetivos, comprobamos como varía el criterio de los usuarios al obtener para una misma recomendación, diferentes interfaces de explicación. Realizamos además un análisis de la influencia de la explicación en el comportamiento de los usuarios.

Y finalmente para el tercero de los objetivos, tras recopilar las opiniones en un cuestionario sobre los elementos utilizados en la explicación, aplicamos el método Delphi para determinar el criterio en conjunto de los usuarios. El cuestionario lo encontramos en el apéndice A.

- **Explicaciones utilizadas:** Utilizamos tres explicaciones de recomendación a lo largo de todo el estudio, las cuales se presentan a un grupo de usuarios para evaluar los objetivos declarados:
 - Histograma del voto de los vecinos, en lo adelante, $E - HV$. Esta explicación resultó adecuada para los usuarios en otros estudios [53].
 - Propuesta presentada en la sección 4.2 con el comportamiento del vecindario, $E - CV$.
 - Interfaz de explicación para un SR de películas presentada en la sección 4.3, $E - HCV$

Dado que la propuesta perfectamente puede ser generalizada, inicialmente se utilizó más de un modelo de recomendación para evaluar la efectividad de las explicaciones. Uno de los modelos empleados, fue el propuesto por Herlocker et al. en [52], mientras el otro fue el algoritmo utilizado en el capítulo anterior¹. Finalmente decidimos utilizar en la experimentación el segundo de los algoritmos de recomendación.

Para comprobar los objetivos marcados, desarrollamos un prototipo, el cual permitió coleccionar y procesar toda la información de interacción con los usuarios. El grupo de usuarios reales que interactuó con la aplicación estuvo compuesto por 35 personas, de ellas 17 del sexo femenino.

4.4.1. Interacción de usuarios

A los 35 usuarios que participaron en el estudio, como paso inicial le facilitamos un mecanismo para valorar un conjunto de películas del conjunto de datos MovieLens. Como mínimo debían valorar un número de 60 películas en la escala del conjunto de datos (1★ a 5★). Una vez valoradas las películas, las valoraciones se incorporaron en el conjunto de datos y cada usuario recibió un identificador para interactuar de forma personalizada con el prototipo. Debemos señalar que estas películas que han sido votadas por el usuario, para mostrarlas en las diferentes interfaces del prototipo, las seleccionamos aleatoriamente. Previamente a la interacción del usuario, se le ha descrito detalladamente cada interfaz de explicación.

¹En estos algoritmos se reportan valores medios de 27.7 y 20.6 películas respectivamente que se muestran en la explicación que proponemos para los parámetros considerados.

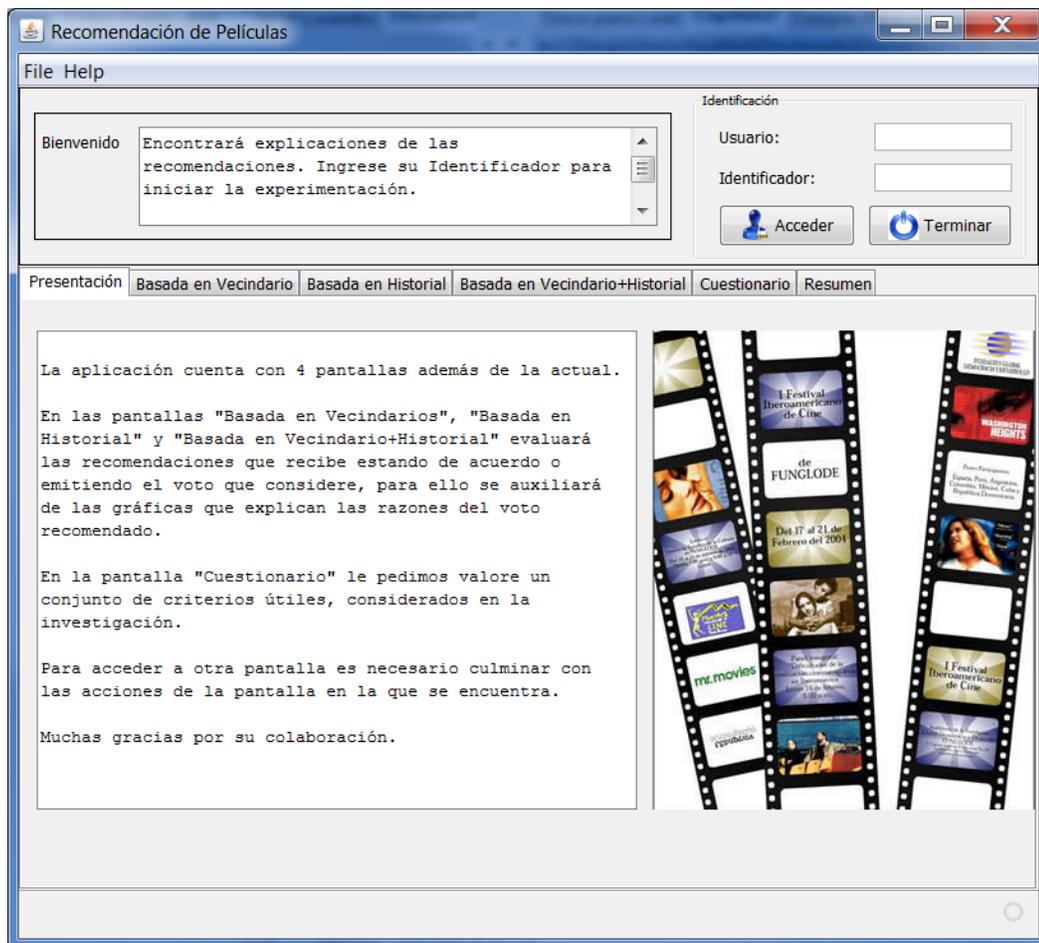


Figura 4.16: Interfaz de presentación del prototipo desarrollado.

En la Figura 4.16 se muestra la interfaz principal a la que los usuarios acceden una vez iniciada la aplicación. La parte superior, que es común para todas las interfaces, queda reservada para mostrarle al usuario una breve descripción de lo que representa cada interfaz en la que se encuentre, mientras a la derecha podrá introducir su nombre e identificador, elementos que permiten personalizar la aplicación para cada usuario. La parte inferior de esta pantalla muestra brevemente la información del resto de opciones a las que el usuario se enfrentará. En total cada usuario encontrará 5 interfaces que conforman el prototipo: Presentación, Basada en Vecindario, Basada en Historial, Basada en Vecindario+Historial y Cuestionario.

Luego de que un usuario se identifique en el sistema, tendrá acceso a las restantes 4 interfaces del prototipo. Para la experimentación que hemos desarrollado, y con vistas a facilitar tanto el estudio experimental como mostrar los resultados alcanzados, hemos dividido todo el estudio en cuatro etapas bien delimitadas. Las etapas que son descritas a continuación, están asociadas a diferentes situaciones a las que el usuario se enfrenta en su interacción con la aplicación. Al ver cada explicación de recomendación, el usuario debe confirmar si está

de acuerdo o no con el voto sugerido por el sistema. En caso de discrepar, debe indicar una alternativa de valoración.

- En la primera etapa, que identificamos con $Etapa_1$, a los usuarios se les muestran explicaciones soportadas en histogramas con los votos de los vecinos ($E - HV$) (Véase Figura 4.17).
- En la segunda etapa, $Etapa_2$, se muestran explicaciones soportadas en el criterio de los vecinos hacia las experiencias pasadas ($E - CV$) (Véase Figura 4.18). La idea con esta etapa, es evaluar de forma independiente nuestra propuesta.
- En la tercera etapa, $Etapa_3$, mostramos la interfaz de explicación que diseñamos en la sección previa ($E - HCV$), donde están incluidas las explicaciones de las etapas anteriores (Véase Figura 4.19).
- En la cuarta etapa, $Etapa_4$, a diferencia de las tres anteriores, las recomendaciones son de películas que el usuario no valoró previamente. En esta etapa se emplean las explicaciones $E - HC$ y $E - HCV$.

Cada usuario navegará inicialmente por cada una de las tres interfaces (Figuras 4.17, 4.18 y 4.19) que contienen las diferentes explicaciones antes de comenzar realmente las etapas de la experimentación, lo que le permite comprender y familiarizarse con el funcionamiento del prototipo. Para esta etapa inicial de entrenamiento se le muestran cuatro películas en cada una de las explicaciones ($E - HC$, $E - CV$ y $E - HCV$), teniendo en todo momento, toda la información disponible de las mismas, tanto el título, la votación original y la recomendada por el sistema. En los ejemplos, y con el objetivo de que el usuario visualice todas las opciones posibles, se muestra situaciones donde se acierta/falla con la recomendación, sobre votos altos/bajos.

Luego de que el usuario se familiariza con las cuatro películas iniciales en cada una de las explicaciones, recibe 10 recomendaciones anónimas en cada una de las tres primeras etapas que hemos descrito anteriormente. El usuario recibe solamente la sugerencia (voto) dada por el sistema y la explicación correspondiente a cada etapa. Además no conoce en ningún momento el título de la película que le estamos recomendando. El usuario además desconoce que las películas que se le muestran en las dos primeras interfaces (Figuras 4.17 y 4.18) son las mismas, y que además son películas que él valoró inicialmente.

En la tercera interface (Figura 4.19), de las 10 recomendaciones que el usuario recibe, un número de ellas (4 o menos) son mostradas en las interfaces precedentes.

Finalmente recomendamos a cada usuario 7 películas seleccionadas aleatoriamente de las que él no ha visto (en estos casos se muestran los títulos), y se le pregunta si vería o no dicha

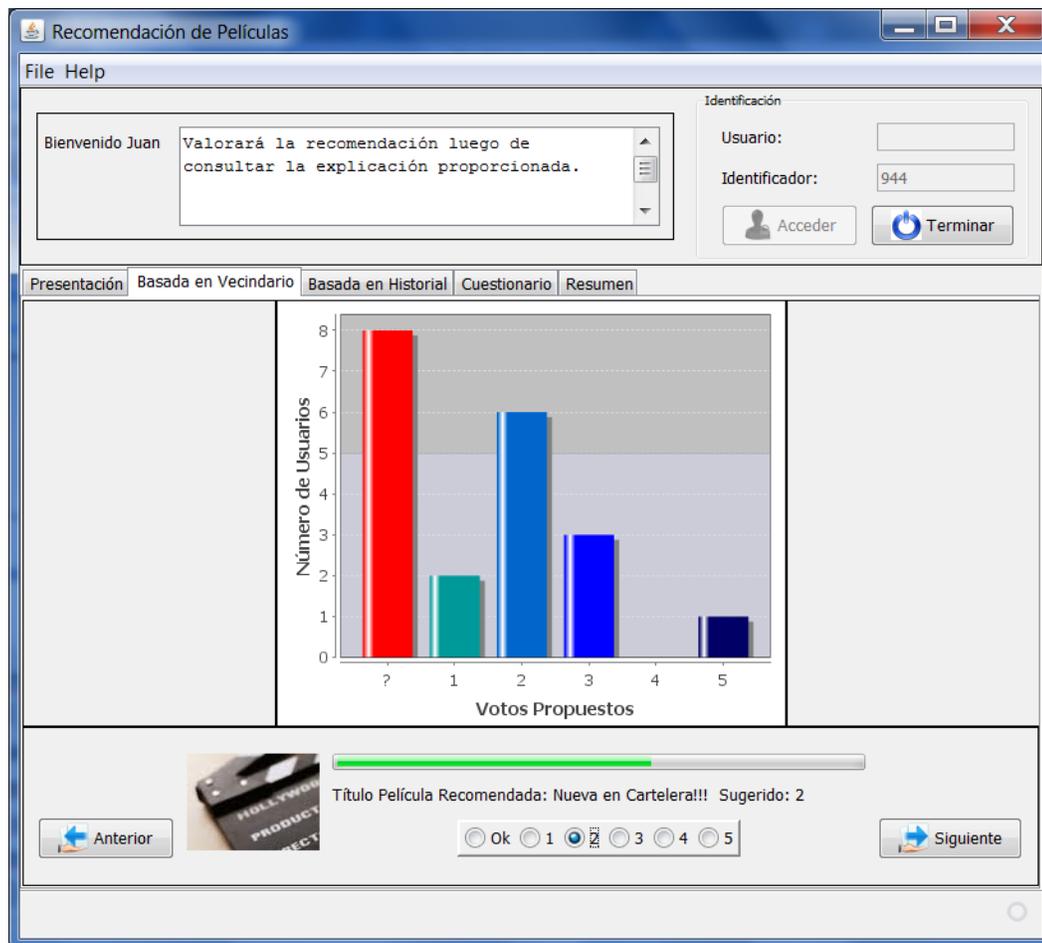


Figura 4.17: Ejemplo de interfaz de la primera etapa, Basada en Vecindario.

película dada la explicación suministrada. En esta cuarta etapa, al usuario le mostramos alternamente la primera y tercera interfaz con sus respectivas explicaciones.

Al finalizar con la evaluación de las explicaciones en las cuatro etapas experimentales, el usuario pasa a responder el cuestionario (Figura 4.20). Este cuestionario se enfoca básicamente en recoger la opinión de los usuarios, respecto a elementos presentes en las explicaciones que han sido suministradas.

4.4.2. Resultados y análisis

En la sección previa se ha descrito toda la metodología seguida para obtener las valoraciones de los usuarios respecto a las explicaciones diseñadas. En lo adelante se ilustran los resultados obtenidos al consolidar las valoraciones de todos los usuarios, tanto cualitativa como cuantitativamente. Todos estos resultados han sido validados estadísticamente.

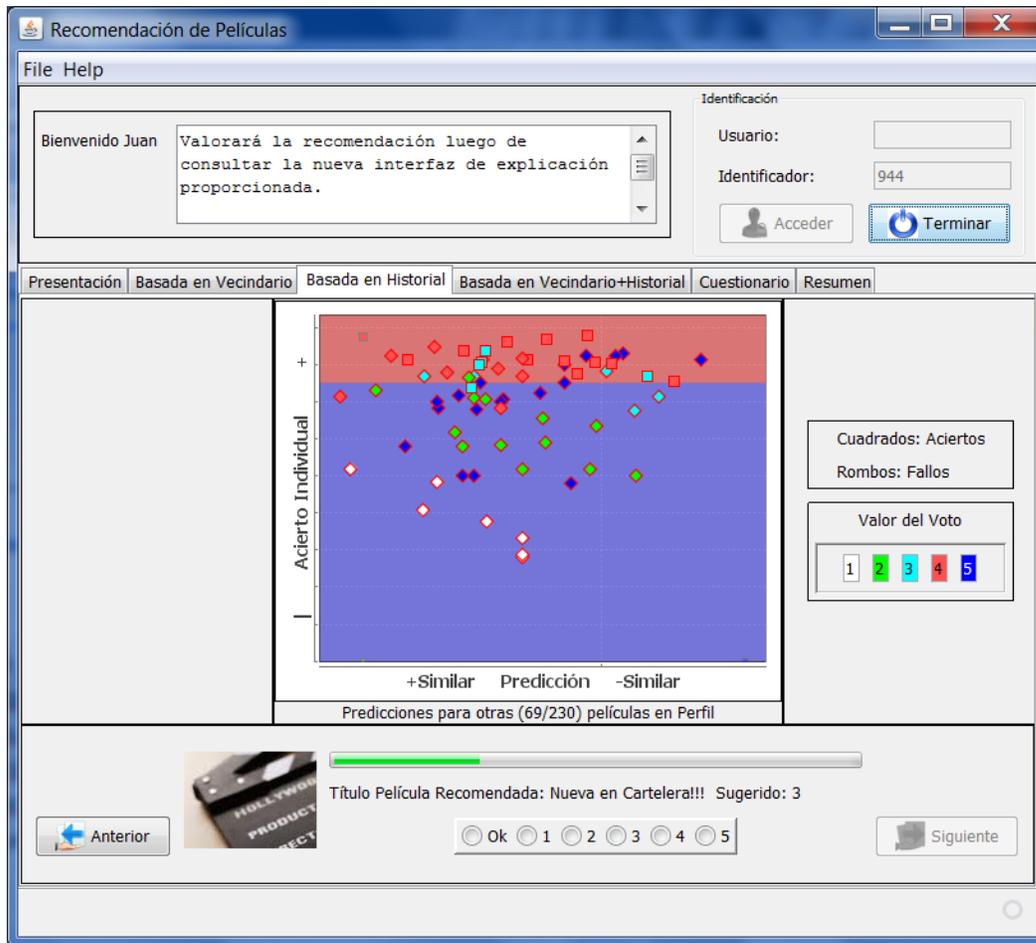


Figura 4.18: Ejemplo de interfaz de la segunda etapa, Basada en Historial.

4.4.2.1. Eficacia de la explicación

Para evaluar la eficacia de las explicaciones decidimos utilizar la MAE como medida general. En las Tablas 4.3, 4.4 y 4.6 mostramos los resultados obtenidos de la valoración por parte de todos los usuarios al recibir las explicaciones de las primeras tres etapas. Cada tabla muestra, en función de las 10 explicaciones que cada usuario recibió:

- la MAE que se obtiene con el modelo de recomendación,
- la MAE que se obtiene después de la votación del usuario,
- el número de recomendaciones aceptadas o rechazadas luego de consultar la explicación.

Análisis de las explicaciones $E - HV$ vs $E - CV$.

Primeramente comparamos los resultados que se obtienen en las dos primeras etapas y en el modelo de recomendación que se emplea (Modelo). La comparación es posible realizarla dado que en ambas etapas se emplea el mismo conjunto de películas.

Tabla 4.3: Resultados generales de la primera etapa.

Usuarios	MAE		Recomendaciones		
	Modelo	$E - HV$		Aceptadas	Rechazadas
1	0,4	0,6		8	2
2	0,9	0,8		9	1
3	0,3	0,4		7	3
4	1	1		8	2
5	0,9	0,5		6	4
6	0,6	0,6		10	0
7	0,7	0,7		8	2
8	0,6	0,9		7	3
9	0,8	0,7		9	1
10	0,7	0,8		7	3
11	0,3	0,5		6	4
12	0,9	0,9		10	0
13	1	0,8		8	2
14	0,9	0,9		8	2
15	1,1	1,1		10	0
16	1,1	1,3		8	2
17	0,5	0,4		7	3
18	0,8	0,6		6	4
19	0,6	0,7		7	3
20	0,7	0,6		9	1
21	0,4	0,6		4	6
22	1,3	1,3		6	4
23	0,8	0,8		10	0
24	0,6	0,7		9	1
25	1	1,7		3	7
26	0,6	0,3		7	3
27	0,8	0,9		5	5
28	0,9	1		7	3
29	1,1	0,6		5	5
30	0,6	0,6		10	0
31	0,9	0,7		8	2
32	0,6	0,6		10	0
33	0,6	0,9		4	6
34	1	1,4		7	3
35	0,4	0,6		6	4
\overline{MAE}	0,7543	0,7857	Total	259	91
Aciertos	135	138	%	74,0	26,0

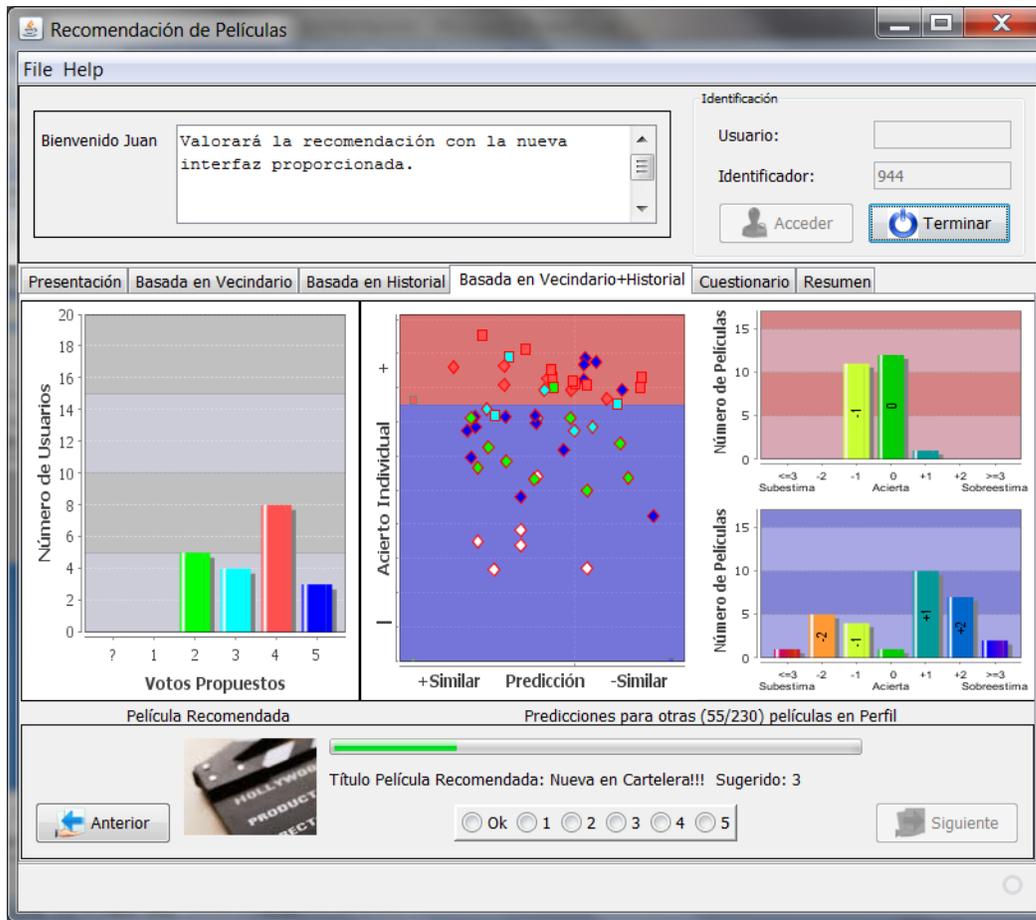


Figura 4.19: Ejemplo de interfaz de la tercera etapa, Pantalla Basada en Vecindario+Historial.

Como podemos observar, la MAE original del sistema de recomendación es de 0,75, mientras al utilizar los usuarios la primera explicación ($E - HV$), la MAE se incrementa hasta 0,79, y con nuestra propuesta en la *Etapa₂* ($E - CV$), donde se muestra el criterio de la vecindad hacia los ítems valorados en el pasado por el usuario, el valor de la MAE se reduce hasta 0,71. Estas diferencias son medianamente significativas en favor de la explicación propuesta en la *Etapa₂* de la experimentación. Entre los resultados del modelo empleado y la primera explicación no se tienen diferencias significativas. Con los resultados obtenidos hasta aquí, confirmamos que la explicación que proponemos ($E - CV$) brinda más ayuda para los usuarios que la interfaz con el histograma de los votos del vecindario ($E - HV$).

En las Tablas 4.3 y 4.4 encontramos para cada una de las explicaciones de las primeras dos etapas, el número de recomendaciones aceptadas o rechazadas por los usuarios (últimas dos columnas). Considerando los resultados obtenidos en cada etapa, a nivel de usuario se encuentran diferencias altamente significativas, siendo la explicación $E - CV$ en la que ocurren más cambios. Este resultado, unido con la MAE que obtuvimos en cada etapa, nos confirma la idea de que los usuarios al disponer de más información tienden a tomar decisiones

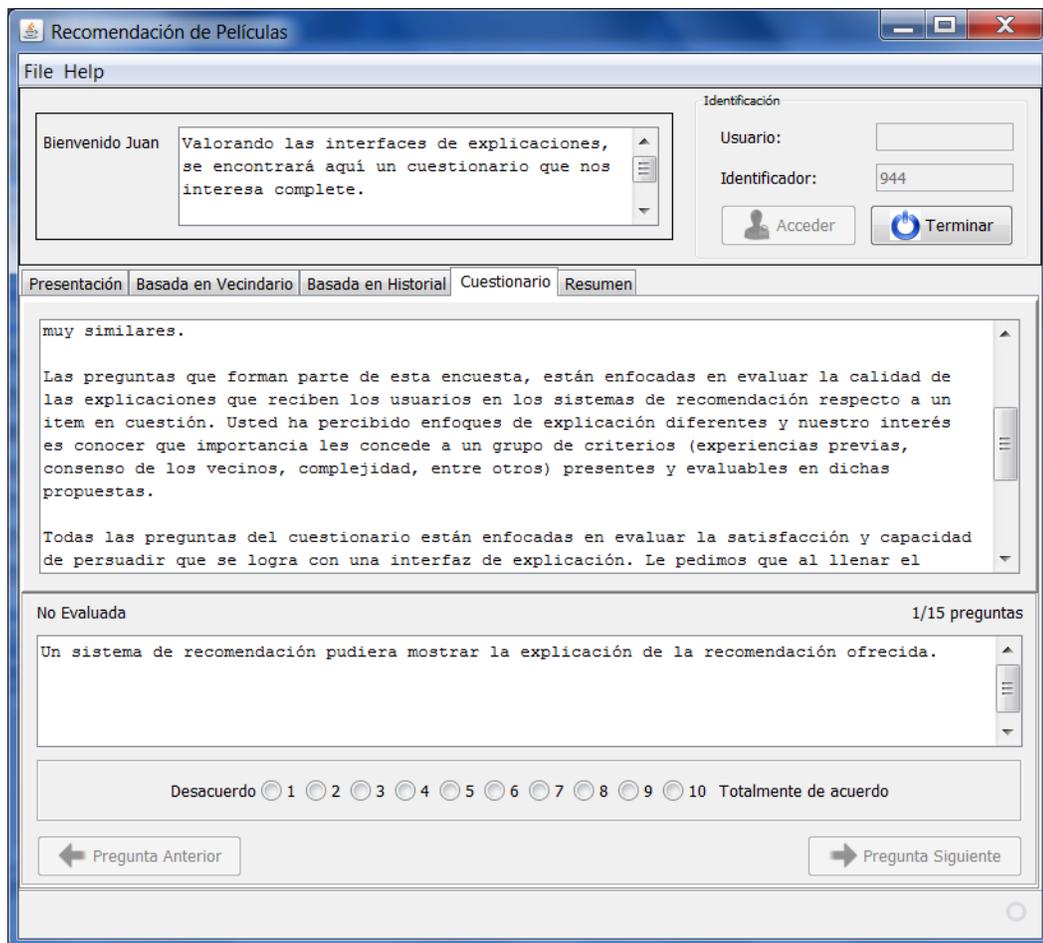


Figura 4.20: Ejemplo de interfaz de Cuestionario. Primera pregunta.

oportunas.

Por otro lado, nos podemos centrar en medir cómo se acierta o no al evaluar cada explicación con respecto al voto original. Para esto nos centraremos en la última fila de las tablas que muestran los resultados de cada etapa. Aunque estos resultados no son significativos, podemos ver que en las dos primeras etapas, al evaluar la segunda explicación, los usuarios logran acertar en el 42 % (147/350) de las ocasiones con el voto original, mientras los resultados del modelo y el histograma de los votos del vecindario ($E - HV$) son inferiores (38,6 % y 39,4 %).

Considerando que las películas en las dos primeras etapas son las mismas, evaluamos con cuál de las dos explicaciones el usuario obtiene mayor beneficio, es decir, con cuál de las dos, luego de analizar la explicación, el voto es más cercano al voto original. En la Tabla 4.5 resumimos el comportamiento para cada uno de los usuarios. Respecto a este análisis, de las 350 explicaciones en cada una de las dos primeras etapas, en 196 (56 %) casos no se tiene una explicación ganadora, dado que en ambas, los usuarios dan una votación similar. Del resto de casos, en 85 (24.3 %) recomendaciones la segunda explicación ($E - CV$) resulta ganadora,

Tabla 4.4: Resultados generales de la segunda etapa.

Usuarios	MAE		Recomendaciones		
	Modelo	$E - CV$	Aceptadas	Rechazadas	
1	0,4	0,4	4	6	
2	0,9	0,7	6	4	
3	0,3	0,4	9	1	
4	1	0,9	7	3	
5	0,9	0,9	4	6	
6	0,6	0,6	10	0	
7	0,7	0,3	6	4	
8	0,6	1	1	9	
9	0,8	0,8	6	4	
10	0,7	1,2	3	7	
11	0,3	0,6	5	5	
12	0,9	0,9	8	2	
13	1	1	10	0	
14	0,9	0,9	8	2	
15	1,1	0,9	5	5	
16	1,1	1,1	4	6	
17	0,5	0,6	9	1	
18	0,8	0,6	8	2	
19	0,6	0,5	5	5	
20	0,7	0,5	8	2	
21	0,4	0,5	7	3	
22	1,3	1,2	8	2	
23	0,8	0,6	8	2	
24	0,6	0,3	5	5	
25	1	1,1	8	2	
26	0,6	0,5	7	3	
27	0,8	0,8	2	8	
28	0,9	0,7	8	2	
29	1,1	0,7	4	6	
30	0,6	0,6	10	0	
31	0,9	0,7	6	4	
32	0,6	0,5	7	3	
33	0,6	0,7	9	1	
34	1	0,9	9	1	
35	0,4	0,4	6	4	
MAE	0,7543	0,7143	Total	230	120
Aciertos	135	147	%	65,71	34,29

Tabla 4.5: Beneficios al evaluar las explicaciones en las dos primeras etapas.

Usuarios	Beneficios		
	$E - HV$	$E - CV$	Empate
1	3	4	3
2	1	2	7
3	2	2	6
4	2	3	5
5	5	1	4
6	0	0	10
7	0	4	6
8	4	3	3
9	3	2	5
10	5	1	4
11	3	2	5
12	1	1	8
13	2	0	8
14	2	2	6
15	2	3	5
16	2	3	5
17	3	1	6
18	3	3	4
19	2	3	5
20	0	1	9
21	1	2	7
22	3	3	4
23	0	2	8
24	0	4	6
25	1	7	2
26	4	2	4
27	3	4	3
28	1	4	5
29	5	3	2
30	0	0	10
31	1	1	8
32	1	2	7
33	2	4	4
34	0	2	8
35	2	4	4
Total	69	85	196

mientras en 69 (19.7%), la explicación donde se muestran el histograma con los votos del vecindario ($E - HV$) fue con la que se obtuvo mayor beneficio.

Análisis de la explicación con histograma de votos del vecindario y el criterio del vecindario.

La Tabla 4.6 contiene el desempeño general obtenido para la explicación $E - HCV$, donde se evidencia que esta explicación mejora ligeramente los resultados del modelo de recomendación. Considerando el número de recomendaciones aceptadas o rechazadas, un 31 % aproximadamente de las sugerencias son rechazadas al consultar esta explicación, superior al 26 % de rechazo que se obtuvo con la explicación del histograma de votos del vecindario. Aunque estos valores no son completamente comparables pues se utilizan diferentes películas para recomendar a los usuarios.

En esta etapa tenemos un número de ítems para los cuales se muestran explicaciones que fueron también mostradas en las dos primeras etapas. De estas recomendaciones comunes en las tres etapas (100 en total), la Tabla 4.7 resume el comportamiento obtenido. Como se observa, la interacción de los usuarios con las explicaciones $E - HCV$ y $E - CV$, mejoran los resultados que se obtienen previamente con el modelo empleado y con la explicación donde mostramos el histograma con los votos del vecindario. En la interfaz $E - CV$ se realizan más cambios (38) respecto al valor recomendado por el modelo, y además, en 17 ocasiones esos cambios permiten acertar con el valor original dado por el usuario.

4.4.2.2. Eficiencia y persuasión de la explicación

En la cuarta etapa de la experimentación, donde se muestran a cada usuario aleatoriamente un total de 7 películas que no ha visto previamente, se le pregunta si dada la explicación le gustaría ver la película o no, valorando finalmente cómo las explicaciones $E - HV$ y $E - HCV$ pueden persuadir a los usuarios en la elección a elegir.

Al analizar anteriormente la variación de criterio del usuario ante cada explicación, evaluamos en cierta forma la capacidad de las explicaciones de persuadir al usuario. Esa evaluación fue implícita puesto que el usuario desconocía que evaluaba una misma recomendación, en dos diferentes explicaciones.

Ahora realizamos esta evaluación de persuasión, para recomendaciones que el usuario si conoce que las está evaluando para explicaciones diferentes. Como definimos en las medidas de evaluación, comprobaremos primeramente como varía el criterio de los usuarios al obtener para una misma recomendación, dos interfaces de explicación diferentes. De esa forma tenemos cuatro posibles opciones:

- La primera es que el usuario acepte la recomendación de ver o no la película luego de ver ambas explicaciones.

Tabla 4.6: Resultados generales de la tercera etapa.

Usuarios	MAE		Recomendaciones		
	Modelo	$E - HCV$		Aceptadas	Rechazadas
1	0,6	0,5		3	7
2	0,7	0,4		8	2
3	0,8	0,8		8	2
4	1	0,8		6	4
5	1,3	1		7	3
6	1,4	1,2		8	2
7	0,6	0,6		6	4
8	1,3	1,5		2	8
9	1	1		10	0
10	0,7	0,9		4	6
11	1,1	0,9		6	4
12	1	0,9		9	1
13	0,7	0,7		10	0
14	0,3	0,4		9	1
15	0,8	0,9		7	3
16	1,1	1,3		6	4
17	0,6	0,7		9	1
18	0,7	0,6		7	3
19	0,8	0,9		5	5
20	0,7	0,5		8	2
21	0,3	0,8		3	7
22	1	1,3		3	7
23	1	0,9		9	1
24	0,5	0,5		6	4
25	1,2	1,4		7	3
26	0,5	0,5		9	1
27	0,8	0,9		3	7
28	0,7	0,7		8	2
29	0,7	0,5		8	2
30	0,6	0,6		8	2
31	0,9	0,7		8	2
32	0,8	0,6		8	2
33	0,9	0,9		10	0
34	0,9	0,9		10	0
35	0,8	0,9		5	5
MAE	0,8229	0,8171	Total	243	107
Aciertos	120	129	%	69,43	30,57

Tabla 4.7: Resultados de las recomendaciones comunes entre las tres primeras etapas.

	Modelo	$E - HV$	$E - CV$	$E - HCV$
MAE	0,84	0,84	0,74	0,8
Aciertos	34	35	43	37
Cambios	-	22	38	31
Aciertos al cambiar	-	8	17	12

- La segunda es que rechace la recomendación que reciba para ambas explicaciones.
- La tercera es que acepte la recomendación al ver la primera explicación y luego al ver la segunda cambie de opinión.
- La cuarta y última es que al ver la primera explicación, rechace la recomendación, y luego la acepte al ver la segunda.

La Tabla 4.8 resume el comportamiento que los usuarios tienen en las 245 recomendaciones recibidas. Se muestra para cada una de las posibles opciones, la distribución de valores que el usuario recibió como sugerencias (1★ a 5★). Se consolidan además las recomendaciones aceptadas y rechazadas para cada explicación. Un análisis de estos datos nos permiten destacar que:

- Los usuarios deciden aceptar la recomendación de ver la película en 123 ocasiones (50,20%), independientemente del voto y la explicación mostrada. Mientras 80 de las recomendaciones ofrecidas fueron rechazadas al consultar ambas explicaciones, incluso 48 de ellas fueron sugeridas con votos de 3★ o 4★. Esto representa, que en un 82,85% de las recomendaciones, luego de ver el histograma de votos del vecindario ($E - HV$), cuando se muestra una explicación soportada en el criterio del vecindario ($E - HCV$), los usuarios reafirman su criterio.
- Los usuarios en 18 ítems recomendados con 3★ o 4★, luego de consultar la explicación del histograma de votos del vecindario ($E - HV$), decidieron ver las películas, pero cambiaron de opinión luego de consultar la otra explicación ($E - HCV$). La información que brinda el vecindario sobre los ítems valorados previamente y la similitud de votación de la película que se recomienda con las votadas previamente, permitieron en estos casos corregir la decisión anterior.
- En 24 ítems recomendados, de ellos 21 con votos de 3★ o 4★, los usuarios al ver la explicación $E - HV$, rechazaron la sugerencia de ver la película, cambiando de opinión posterior a consultar la segunda interfaz, $E - HCV$. En este caso, el histograma de votos del vecindario le brindó información al usuario para rechazar la recomendación, pero luego al compararla dentro de la explicación ($E - HCV$), se tuvieron argumentos para cambiar la opinión.

- De las 36 recomendaciones con votos bajos (1★ o 2★), los usuarios en 35 ocasiones están convencidos que no la verían al ver la explicación con el histograma de votos del vecindario, sin embargo, en 3 de esas ocasiones, la explicación $E - HCV$ logra convencer a los usuarios para que las vean.
- La distribución de votos de las recomendaciones, muestra que fueron recomendadas películas con votos de 3★ o 4★ en mayor medida, siendo de este grupo de películas, las recomendadas con 3★ las que más se rechazan, independientemente de la explicación.

El análisis de las opciones anteriores nos permiten concluir que la interfaz $E - HCV$ muestra información útil a los usuarios para tomar decisiones, como las que se evidencian en las situaciones previas. Los usuarios en un 17,15 % de las recomendaciones, al ver la explicación $E - HCV$, toman una decisión contraria a la considerada al ver la primera de las interfaces.

Tabla 4.8: Resultados generales de la cuarta etapa. Elección de las recomendaciones recibidas.

Comportamiento	Recomendación					Total	%
	1★	2★	3★	4★	5★		
Aceptada en $E - HV$ y $E - HCV$		1	44	75	3	123	50,2041
Rechazada en $E - HV$ y $E - HCV$	5	27	40	8		80	32,6531
Aceptada en $E - HV$ y rechazada en $E - HCV$			14	4		18	7,3469
Rechazada en $E - HV$ y aceptada en $E - HCV$	1	2	13	8		24	9,7959
Total	6	30	111	95	3	245	
Aceptada en $E - HV$		1	58	79	3	141	57,5510
Rechazada en $E - HV$	6	29	53	16		104	42,4490
Aceptada en $E - HCV$	1	3	57	83	3	147	60,0000
Rechazada en $E - HCV$	5	27	54	12		98	40,0000

El análisis de las 245 recomendaciones mostradas a los usuarios en cada explicación, nos confirma la hipótesis de que en la interfaz $E - HCV$, los usuarios tienden a cambiar un gran número de sugerencias o a reafirmarlas, porque disponen de información que les permiten tomar adecuadas decisiones.

Incidencia de la explicación en el acierto o no del voto original

Analizamos a continuación el comportamiento de los usuarios al recibir las explicaciones en las tres primeras etapas de la experimentación. Nos centramos en medir si luego de analizar la explicación, al modificar la recomendación recibida, el usuario acierta o no con el voto original. En las Tablas 4.9 y 4.10 resumimos para todas las recomendaciones y explicaciones, el comportamiento de los usuarios en cada posible combinación (voto original - voto sugerido) de los valores posibles (1★ a 5★).

Tabla 4.9: Resultados para diferentes valores de recomendaciones. Primera y segunda etapa.

Voto		Total	$E - HV$		$E - CV$	
Original	Recomendado		Cambios	Aciertos	Cambios	Aciertos
1★	1★	0	0	0	0	0
1★	2★	2	1	0	2	1
1★	3★	2	2	0	0	0
1★	4★	1	0	0	1	0
1★	5★	0	0	0	0	0
2★	1★	0	0	0	0	0
2★	2★	7	5	0	4	0
2★	3★	15	5	2	7	4
2★	4★	12	4	0	1	0
2★	5★	0	0	0	0	0
3★	1★	0	0	0	0	0
3★	2★	14	3	3	6	6
3★	3★	46	12	0	17	0
3★	4★	29	6	4	9	5
3★	5★	4	1	0	2	1
4★	1★	0	0	0	0	0
4★	2★	10	3	0	6	0
4★	3★	53	16	11	22	18
4★	4★	72	13	0	16	0
4★	5★	5	0	0	2	2
5★	1★	0	0	0	0	0
5★	2★	2	0	0	2	0
5★	3★	17	6	0	9	2
5★	4★	47	13	12	13	9
5★	5★	12	1	0	1	0
Total		350	91	32	120	48

En la Tablas 4.9 y 4.10, donde tenemos por cada explicación, los cambios que realizaron los usuarios y el número de estos cambios que permitieron acertar con el voto original, podemos destacar lo siguiente:

- En la explicación $E - CV$ significativamente ocurren más cambios y es además la explicación donde más los usuarios aciertan al cambiar.
- En menos del 30% de los casos donde el voto recomendado y el original fueron iguales, en las tres interfaces el voto fue cambiado.
- En los casos donde el voto recomendado fue diferente del voto original, en la segunda de las explicaciones ocurrieron significativamente más cambios (38.5%) y al mismo tiempo más aciertos al producirse los mismos (58.54%).

Tabla 4.10: Resultados para diferentes valores de recomendaciones. Tercera etapa.

Voto		$E - HCV$		
Original	Recomendado	Total	Cambios	Aciertos
1★	1★	0	0	0
1★	2★	0	0	0
1★	3★	4	0	0
1★	4★	1	1	0
1★	5★	0	0	0
2★	1★	0	0	0
2★	2★	8	3	0
2★	3★	16	6	2
2★	4★	20	4	0
2★	5★	0	0	0
3★	1★	0	0	0
3★	2★	8	3	3
3★	3★	51	15	0
3★	4★	46	18	10
3★	5★	5	0	0
4★	1★	0	0	0
4★	2★	11	5	0
4★	3★	49	17	16
4★	4★	59	16	0
4★	5★	2	0	0
5★	1★	0	0	0
5★	2★	0	0	0
5★	3★	18	5	0
5★	4★	62	19	14
5★	5★	8	2	0
Total		368	114	45

- En los casos donde el voto recomendado fue mayor que el voto original (sobrestimación), es en la tercera de las explicaciones donde existe un mayor número de cambios, pero este número no es significativo respecto a los cambios producidos en el resto de las explicaciones. El número de cambios que producen acierto respecto al voto original, se manifiestan significativamente mejor en la segunda de las explicaciones.
- En los casos donde el voto recomendado es menor que el original (subestimación), es significativamente en la segunda de las explicaciones donde se producen un mayor número de cambios.

Cuando se analizan las tablas anteriores y extraemos (filtramos) las posibles combinaciones dado el voto original o el recomendado, se tiene:

- Para los casos que el voto original fue de 2★, los usuarios tienden a cambiar más al ver la explicación $E - HV$, pero aciertan al cambiar en un mayor % en $E - CV$.
- Para los casos donde el voto original fue 3★, se producen más cambios y aciertos al cambiar al analizar la explicación $E - HCV$.
- Para los casos donde el voto original fue 4★, se producen más cambios y aciertos al cambiar en la explicación $E - CV$.
- Para los casos donde el voto original fue 5★, los usuarios tienen a cambiar más en $E - CV$ y en $E - HCV$, pero aciertan más al cambiar en $E - HV$.
- Cuando se recomienda 2★ o 3★, en $E - CV$ es donde hay más cambios y aciertos al cambiar.
- Cuando se recomienda 4★, en la explicación $E - HCV$ es donde más cambios se producen, pero es en $E - HV$ donde los cambios son más efectivos.
- Cuando se recomienda 5★ se tienen muy pocos casos, pero solamente en la explicación $E - CV$ es donde se producen aciertos al cambiar.

Las situaciones anteriores evidencian la utilidad de la información que brindamos en la explicación $E - CV$ para todo tipo de votaciones (altas y bajas). También debemos reconocer la utilidad para los usuarios de la interfaz $E - HCV$, en la que disponen tanto de la explicación con el histograma de votos del vecindario, como de la explicación con el criterio de los vecinos hacia sus películas previas.

4.4.2.3. Satisfacción de los usuarios con las explicaciones

Los análisis realizados previamente, permitieron procesar información cuantitativa obtenida de la interacción de los usuarios con el prototipo diseñado. Para obtener la opinión cualitativa sobre los elementos considerados en las explicaciones diseñadas, aplicamos a los usuarios un cuestionario.

A cada uno de los usuarios se le mostró un cuestionario con 23 preguntas que debían ser evaluadas en una escala de **1** (Desacuerdo) a **10** (Totalmente de acuerdo) (Véase el cuestionario al final del capítulo). Dado que en situaciones como estas, donde se tiene un grupo de 35 usuarios, llegar a un acuerdo sobre una temática no es una tarea sencilla, se decidió emplear un método que permitiera establecer un consenso entre los usuarios acerca de las preguntas realizadas. La idea en general es evaluar la satisfacción de los usuarios con las explicaciones.

El método Delphi, donde se asume que los juicios del grupo son más válidos e importantes que los juicios individuales, fue empleado en este caso, aunque solamente aplicando el

cuestionario en una sola ocasión a los usuarios. Este método encuentra su inspiración en el antiguo oráculo de Delphos y fue ideado originalmente a comienzos de los años 50 del siglo XX [76, 98].

De forma general con el método empleado, obtenemos que los usuarios:

- Confieren a la transparencia en los sistemas de recomendación una vital importancia, consenso de 10.
- Están totalmente de acuerdo que una explicación debería mostrar como los usuarios similares han valorado el producto recomendado, consenso de 10.
- Consideran que mostrar las experiencias previas del usuario activo en una explicación ayuda para tomar una decisión, consenso de 9.
- Consideran que se ofrece una mejor explicación si se utilizan tanto las experiencias previas como los votos de usuarios similares, consenso de 10.
- Luego de interactuar con el prototipo, creen que las explicaciones han ayudado a incrementar la confianza en el vecindario, consenso entre 8-9.
- Consideran que en cierto grado las explicaciones proporcionan ayuda para tomar una decisión oportuna, consenso de 8.
- Todas las explicaciones mostradas contienen información relevante para tomar decisiones adecuadas, pero cada una en cierto grado. Para la explicación $E - HV$, consenso de 8, mientras para el resto de las explicaciones, consenso de 9.
- Preferirían la explicación $E - HV$ en sistemas de recomendación destinados a películas o prendas de vestir, mientras para entornos académicos, de libros y restaurantes, ellos prefieren las explicaciones $E - CV$ y $E - HCV$.

Además de las preguntas del cuestionario, los usuarios emitieron su opinión sobre otros elementos que consideraban importantes en el estudio. Destaca que inicialmente se llegan a considerar complejas las propuestas $E - CV$ y $E - HCV$, pero que éstas sin dudas contienen elementos útiles para tomar decisiones.

De forma general, los resultados obtenidos confirman la hipótesis de que las explicaciones son necesarias en los sistemas de recomendación, que las mismas deben ser facilitadas a los usuarios y que estos suelen emplearlas para tomar decisiones. Además, aunque en la opinión de los usuarios prevalecen elementos subjetivos asociados con la simplicidad de la interfaz primera donde se muestran histogramas con el voto del vecindario ($E - HV$), ellos prefieren consultar interfaces más informativas (de ser posible donde aparezcan ambos enfoques) en caso de dudas.

4.5. Conclusiones

En el capítulo obtuvimos una propuesta de explicación para los SR de filtrado colaborativo, que resulta competitiva respecto al histograma de votos del vecindario, interfaz de explicación esta última que ha sido bien valorada anteriormente por la comunidad. Las características esenciales en las que fundamentamos la propuesta, están relacionadas con dos elementos al que los usuarios confieren una gran importancia al tomar una decisión, la opinión sobre los productos previos y la opinión de los amigos.

En la explicación que proponemos, a un usuario se le muestran los ítems valorados previamente y al mismo el criterio del vecindario seleccionado respecto a dichos ítems. Estos ítems tienen una fuerte relación semántica con el ítem para el cual ofrecemos la explicación, siendo capaces de explicar la recomendación y convencer al usuario a aceptar la sugerencia recibida.

Los resultados experimentales que obtuvimos con usuarios reales, demuestran la utilidad de la propuesta en la recomendación de películas. La aceptación y opinión de los usuarios, nos confirma la idea de generalización en otros entornos, donde las recomendaciones deben estar adecuadamente fundamentadas para que los usuarios las acepten.

Resaltamos de la experimentación general con usuarios, que la propuesta de explicación:

- finalmente brinda más información al usuario (en cualquiera de las formas en que se utilizó, esta ayudó a reconsiderar el voto dado, por encima del 30 % de las recomendaciones, mientras el histograma ($E - HV$) ayudó en solo un 26 %), y,
- la información que brinda es más útil dado que ayuda a predecir mejor el voto que originalmente se dio (la MAE original del SR sin emplear interfaces de explicación fue de 0,75, mientras al usar el histograma de votos del vecindario este se incrementa hasta 0,79 y con nuestra propuesta este valor se reduce a 0,71).

Capítulo 5

Aprendizaje basado en la explicación para la mejora de la recomendación

El estudio con usuarios reales que mostramos en el capítulo anterior, nos permitió detectar por un lado, el comportamiento de los usuarios al recibir la información de la explicación, y por otro lado, el interés permanente en algunos datos.

Al mostrarle las explicaciones a los usuarios, no en todos los casos se reflejó el mismo comportamiento. Para un grupo de ellos se tenían explicaciones muy informativas y eficaces, donde al consultarlas se convencían de la sugerencia recibida o tenían argumentos para modificar la recomendación. Otros usuarios por su parte, encontraban oportunas las explicaciones pero su criterio de aceptación o no de la recomendación, dependía de otros factores.

Lograr detectar este comportamiento de forma automática y emplearlo en la mejora del desempeño del proceso de recomendación es lo que se persigue. Para esto la propuesta que se describe en el capítulo utiliza los datos de, i) la estrategia de recomendación, y ii) del modelo e interfaz de explicación.

En los resultados experimentales mostramos que aunque no para todos los usuarios, se logra predecir el error siguiendo la estrategia que proponemos, su oportuno empleo permite en un número de usuarios, mejorar las recomendaciones originales del SR.

En este capítulo primeramente describimos la idea que proponemos y su origen. Seguidamente analizamos en detalles el algoritmo que se utiliza en la experimentación, considerando algunos ejemplos para facilitar su comprensión. Finalmente en la experimentación, destacamos todos los resultados obtenidos y las principales conclusiones.

5.1. Evidencias empíricas que apoyan un aprendizaje de las explicaciones

Partimos de considerar una recomendación similar por dos vecindarios diferentes de 5 vecinos, por ejemplo, un voto de 3★, donde se asume además que cada uno de los vecinos

contribuye de igual forma a la valoración y que ésta es obtenida cuando las distribuciones de votos son $\{1,2,3,4,5\}$ y $\{3,3,3,3,3\}$. Al analizar los histogramas de los votos que se obtienen como interfaces de explicación, el usuario tendría situaciones diferentes pero muy interesantes.

La primera distribución de votos supondría una alta incertidumbre en el usuario, pues se evidencia una total discrepancia en la valoración hacia dicho ítem y el usuario pudiera desconfiar y no aceptar la recomendación recibida. Mientras en la segunda se evidencia un alto consenso de los vecinos y una casi segura aceptación de la recomendación por parte del usuario. Otras situaciones, como por ejemplo, la distribución $\{1,5,5,5,5\}$ para una sugerencia de 4★, podría hacer pensar al usuario en desechar el voto extraño en la distribución (1★) y modificar la sugerencia a un valor de 5★.

En las múltiples pruebas realizadas en el trabajo con interfaces de explicación en el Capítulo 4, hemos podido comprobar que los usuarios, tras observar una interfaz de explicación, modifican frecuentemente los votos propuestos y que con estas acciones se logra una mejora en las recomendaciones, y por tanto, en el desempeño del SR en general.

Esto nos llevó a pensar que deben existir algunos procesos internos en el pensamiento de los usuarios que los motive a realizar una acción determinada sobre la recomendación recibida. Es evidente que estas acciones dependen de la experiencia que adquiera el usuario con las recomendaciones previas, la sugerencia propuesta por el sistema y la información disponible en las interfaces de explicación, entre otros factores.

En el Capítulo 4 el usuario recibe una explicación para la recomendación ofrecida al ítem I_t , soportada en la recomendación que daría el vecindario para k ítems valorados previamente por el usuario activo, $\{I_1, \dots, I_k\}$. A lo largo de este capítulo mostraremos los elementos que nos permiten afirmar que esta información, extraída de las interfaces de explicación, puede ser utilizada para mejorar las recomendaciones que se ofrecen a los usuarios. Se pueden tener casos donde los vecinos tiendan a subestimar/sobreestimar el voto y el usuario note dichas tendencias, pudiendo corregir la sugerencia recibida.

Nosotros inicialmente decidimos comprobar si la información disponible al consultar un histograma de votos incidía en la recomendación y en los usuarios. El siguiente ejemplo ilustra nuestra propuesta en un proceso de recomendación.

Ejemplo

Suponga que el sistema recomienda para un ítem el voto 3,25 cuando proporciona como explicación la distribución de votos de los vecinos (información proporcionada por la interfaz); en la Figura 5.1, $\{3,4,8,3,2\}$.

Podemos considerar la entropía de dicha distribución como una alternativa para resumir dicha información, calculándose como:

$$H(o) = \sum_j p(r_j) \log_2\left(\frac{1}{p(r_j)}\right), \quad (5.1)$$

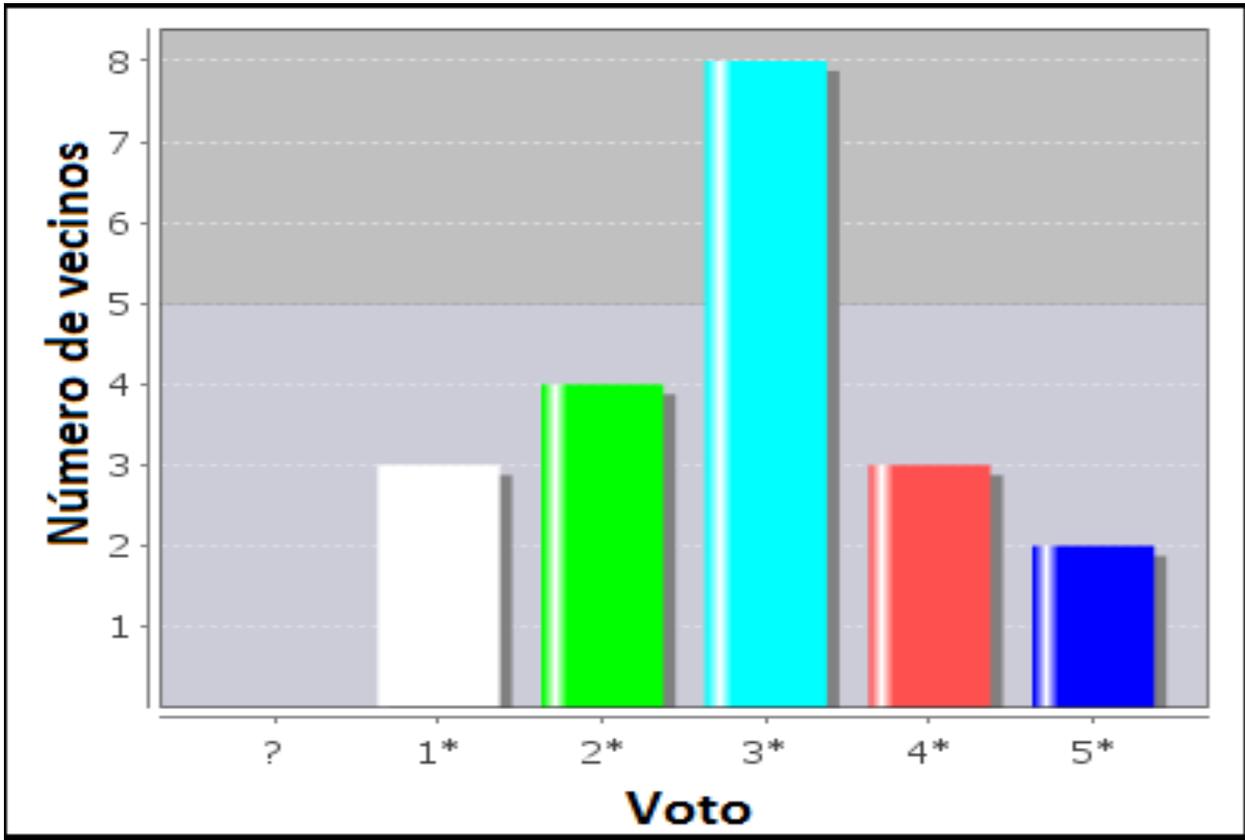


Figura 5.1: Ejemplo de histograma con las votos del vecindario.

donde la suma es sobre el conjunto de votos posibles ($1\star$ a $5\star$), representando $p(r_j)$ la probabilidad de que el vecindario vote con el valor de r_j el ítem. Según la distribución anterior tendríamos $H(o)$ igual a 2,14.

Nos preguntamos si existe alguna relación entre dicha información y la que proporciona el sistema en casos anteriores. Por ejemplo, imaginemos que para este usuario podemos conocer la explicación que se ha dado cuando observaba ítems previos (la Tabla 5.1 refleja dicha información). En dicha tabla para cada uno de los ítems mostramos el voto original dado por el usuario, el voto recomendado por el sistema, la distribución de votos del vecindario, la entropía, y el error cometido en la predicción.

Con esta información, podemos intentar estimar el error que se comete en la predicción, $\hat{e}(o)$. Imaginemos que al aplicar un simple algoritmo de regresión lineal (RL), se obtiene la siguiente ecuación para estimar el error cometido:

$$\hat{e}(o) = 0,4 * \hat{r}(a, o) - 0,25 * H(o). \quad (5.2)$$

Dado el caso que estamos considerando, donde el voto recomendado es de 3,25 y la entropía es de 2,14, el valor que se estimaría del error sería de 0,765. Si adicionamos este valor al voto recomendado por el sistema, obtenemos un voto de 4,015.

Tabla 5.1: Información de las recomendaciones en ítems previos para un usuario.

Ítem	$r(a, o)$	$\hat{r}(a, o)$	Votos del vecindario					$H(o)$	$e(o)$
			1★	2★	3★	4★	5★		
i_1	3	3,25	2	3	8	6	1	2,01	-0,25
i_2	5	4	0	0	0	20	0	0	1
i_3	4	3,1	0	0	16	4	0	0,72	0,9
i_4	3	2,6	0	4	10	4	2	1,76	0,4
i_5	2	1,84	4	10	4	2	0	1,76	0,16
i_6	4	3,3	1	2	12	5	0	1,49	0,7
i_7	2	2,15	4	4	4	4	4	2,32	-0,15
i_8	5	3	0	0	20	0	0	0	2
i_9	5	4,05	0	1	2	12	5	1,49	0,95

Con esta idea estamos considerando, que al disponer de todas las experiencias previas de un usuario y la información descrita anteriormente, ante una nueva recomendación, somos capaces de predecir el error y mejorar la recomendación.

La pregunta que nos hacemos ahora es si realmente esta corrección de la recomendación es útil para un usuario, en el sentido de que permita mejorar el comportamiento de nuestro sistema. Para ello, una alternativa simple sería ver si esta aproximación nos proporciona buenos resultados cuando consideramos las experiencias previas del usuario. Continuando con nuestro ejemplo, supongamos que tenemos el usuario U , para el que se han recomendado 12 ítems (ver Tabla 5.2), de los que conocemos tanto la valoración dada por el usuario a cada ítem (primera columna) como la recomendación dada por el sistema (segunda columna), y su posible explicación (representada en la cuarta columna mediante la entropía de la distribución).

Para cada uno de estos ítems recomendados, podemos realizar el proceso anterior (por simplicidad asumiremos que la recomendación se realizará con el mismo vecindario, y que dan lugar a la misma función de regresión), obteniendo las nuevas predicciones que es muestran en la última columna de la Tabla 5.2.

Obviamente, esta corrección no siempre tiene porque ser correcta, por ejemplo, para el ítem de la primera fila se empeora la predicción, pero en media para este usuario el realizar la corrección nos permite mejorar el rendimiento del sistema, logrando mejorar la MAE desde 1,0333 hasta 0,7063.

Esta sería una situación ideal, sin embargo, como veremos posteriormente, la utilidad de esta aproximación depende de los usuarios. Podemos encontrar un grupo de usuarios donde esta aproximación no es válida, las correcciones no permiten mejorar el rendimiento global del sistema, sino que incluso lo empeoran considerablemente.

Este ejemplo nos ha permitido motivar nuestra propuesta de aprendizaje basada en la explicación, focalizada en el uso del histograma de votos. Pero también es posible encontrar

Tabla 5.2: Comportamiento positivo de la propuesta para nuevas recomendaciones del usuario U .

$r(a, o)$	$\hat{r}(a, o)$	$e(o)$	$H(o)$	$\hat{e}(o)$	$\hat{r}(a, o) + \hat{e}(o)$
4	3,88	0,12	2,01	1,0495	4,9295
5	3,5	1,5	0	1,4	4,9
4	3	1	2,32	0,62	3,62
2	2,3	-0,3	2,32	0,34	2,64
3	2,31	0,69	1,76	0,484	2,794
4	2,8	1,2	1,49	0,7475	3,5475
4	3,05	0,95	2,32	0,64	3,69
4	2	2	0	0,8	2,8
3	3,15	-0,15	1,49	0,8875	4,0375
4	3,2	0,8	1,49	0,9075	4,1075
2	1,99	0,01	0	0,796	2,786
4	3,42	0,58	2,32	0,788	4,208
MAE	1,0333				0,7063

características en la interfaz de explicación propuesta en el capítulo anterior que permiten identificar el tipo de error que se está cometiendo en las predicciones. Así, ya en nuestros experimentos con usuarios reales, pudimos observar ciertas tendencias al consultar nuestra propuesta de explicación. Por ejemplo, a modo ilustrativo, podemos considerar el caso del usuario 22 de MovieLens, donde en la Figura 5.2, el número de ítems en la parte izquierda de la interfaz de explicación está altamente correlacionado (correlación = 0,58) con el error cometido en la recomendación. En la Figura 5.2 encontramos para un usuario, cada uno de los errores cometidos en la recomendación de sus ítems (información vertical), y para cada caso, el número de ítems que se muestran al usuario en la parte izquierda de la interfaz de explicación (información horizontal). En este caso, nos podemos percatar de que el sistema es propenso a sobrestimar el voto cuando el número de ítems en el parte izquierda es mayor que 10. Así, si el sistema predice la calificación 3,75 para un ítem que se analiza, y tenemos 25 ítems en el lado izquierdo de la interfaz de explicación, tendríamos $x = 25$ y entonces el error que estimaríamos sería 0,71 ($e(t) = 0,061x - 0,811$), con lo que el usuario pudiera modificar la recomendación inicial hacia 3★.

Las situaciones anteriores y toda la etapa experimental con las explicaciones diseñadas, inspiró esta propuesta de investigación, que pudiera ser formulada considerando la siguiente pregunta, ¿es posible aprender de las interfaces de explicación? Si la respuesta fuese positiva, nos preguntaríamos, ¿cómo podría emplearse la información aprendida de dichas interfaces para mejorar las recomendaciones? Y estas son las cuestiones que vamos a intentar solventar en el resto del capítulo.

Debemos destacar que no encontramos en la literatura estudios sobre el efecto de las explicaciones basadas en ítems pasados. Por medio de la propuesta que describimos en adelante,

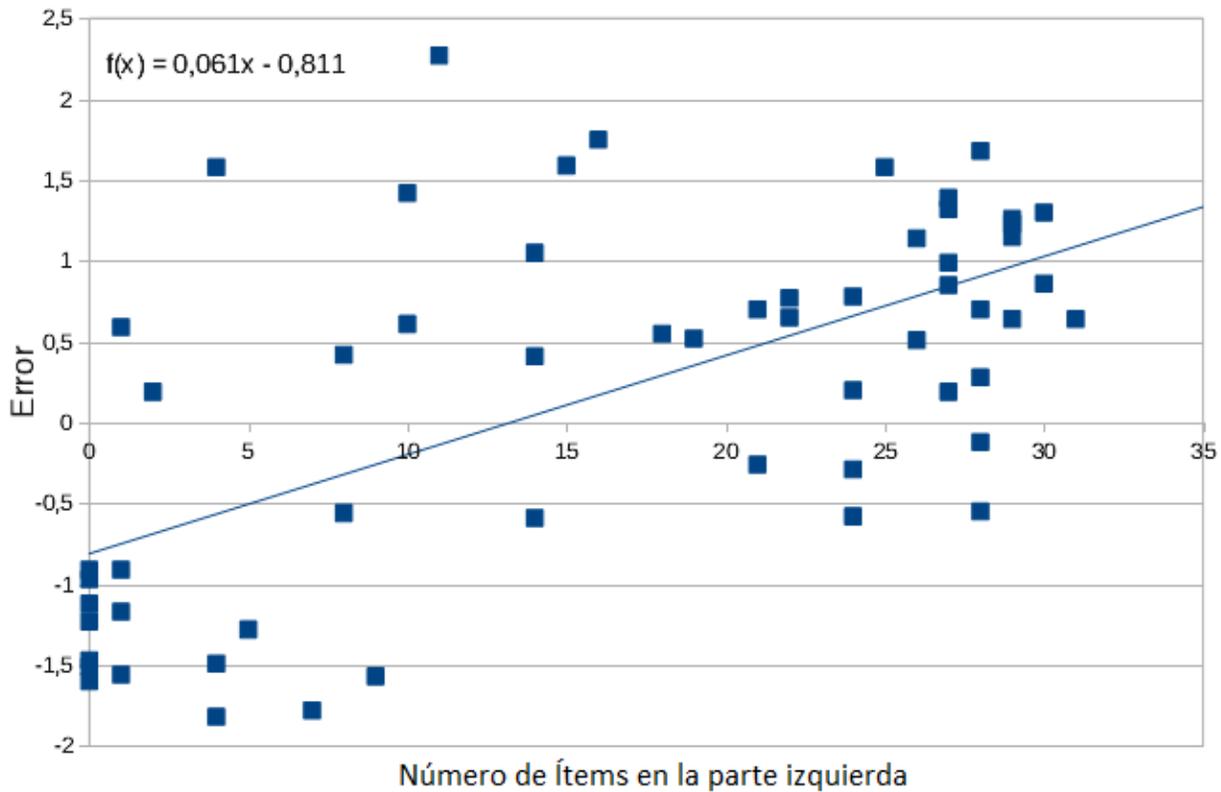


Figura 5.2: Ejemplo de correlación del error de la recomendación vs número de ítems en la parte izquierda de la interfaz de explicación propuesta en el Capítulo 4.

podemos explorar los costos y beneficios de las acciones que realizan los usuarios (al juzgar las sugerencias) para la mejora de las recomendaciones ofrecidas, lo cual es un problema abierto en el campo. Aunque la propuesta se enfoca en un sistema de recomendación basado en vecindario, esta puede ser aplicada en otras estrategias de recomendación.

Antes de gestionar toda la información para el aprendizaje, debemos destacar que aprender de las interfaces de explicación tiene algunos problemas:

- Primeramente, se necesita conocer toda la información de los ítems recomendados, tanto en los que el usuario muestre interés positivo como en los que no.
- Para determinados usuarios no se podrá realizar el proceso al no tener suficiente información.
- Por último, el aprendizaje es muy dependiente de la estrategia de recomendación, ya que resultará esencial para extraer toda la información relacionada con las acciones que realiza el usuario. Al considerar un SR basado en vecindario, puede ser muy probable que estas acciones dependan de los vecinos utilizados para obtener la sugerencia.

5.2. Selección de características

Algo clave en nuestra propuesta, es que nosotros disponemos de toda la información necesaria para los ítems observados, por un lado, conocemos el voto real y la sugerencia dada por el sistema para los ítems, por lo que si sabemos cuál sería la acción que haría el usuario (confiar en el voto o modificarlo), y por otro lado, tenemos la interfaz de explicación para cada recomendación. Cada explicación, al estar soportada en el criterio del vecindario seleccionado sobre los ítems valorados previamente por el usuario activo, nos facilita obtener la información asociada a ella en cualquier momento. En lo adelante identificaremos como F_{RS} la información que está directamente relacionada con los ítems observados (o) del usuario activo a y el histograma de votos del vecindario. Teniendo por tanto:

- $\hat{r}(a, o)$, la recomendación o voto sugerido.
- $e(o)$, el error en la predicción (atributo clase), definido como $e(o) = r(a, o) - \hat{r}(a, o)$. Nótese que este es el error que estamos tratando de aprender de la explicación.
- $H(o)$, la entropía que mide la variedad (incertidumbre) en la distribución de votos de los vecinos seleccionados para predecir un voto al ítem que se analiza.

Mientras que los atributos o características que disponemos en una explicación cualquiera S (véase un ejemplo en la Figura 5.3) para un ítem, o , los identificaremos como F_{Ex} . Estos son:

- $\#S_o$, número de ítems en el gráfico.
- $\eta(o)$, promedio del acierto individual en el conjunto de ítems en S_o :

$$\eta(o) = \frac{1}{\#S_o} \sum_{i \in S_o} h(i), \quad (5.3)$$

donde $h(i)$ es el error promedio en las predicciones de los ítems.

- $MAE(o)$, representa la media del error absoluto en las predicciones para aquellos ítems en S_o :

$$MAE(o) = \frac{1}{\#S_o} \sum_{i \in S_o} |\hat{r}(a, i) - r(a, i)|. \quad (5.4)$$

La diferencia con $\eta(o)$ es que cuando se considera la MAE los votos de los vecinos han sido combinados (como una mezcla ponderada) con el objetivo de determinar la predicción para el ítem.

Además, hemos observado que la distribución de los ítems en la interfaz de explicación puede ser relevante para determinar la calidad de la predicción. Como hemos mencionado, el comportamiento no era el mismo para todos los usuarios: se tenían explicaciones muy eficaces para algunos pero no tanto para otros. Al centrarnos en este detalle, pudimos detectar que los usuarios luego de observar varias explicaciones, tenían una tendencia a concentrarse en la información en general que percibían, pero fundamentalmente en la que procedía de algunas partes en particular.

Por ejemplo, para algunos usuarios era muy relevante el número de ítems mostrados en la interfaz de explicación, por ejemplo, el número de ítems que eran predichos similarmente por el vecindario; mientras para otros usuarios lo importante eran los ítems en los que se producían errores en las predicciones.

Así, al igual que en la Figura 5.3, hemos dividido el gráfico en cuatro partes considerando, por un lado, la similitud de las predicciones: hemos usado un umbral de 1 para dividir el eje horizontal, dividiendo el gráfico en una parte izquierda (más similares) y una derecha (menos similares). Recordemos que este eje establece que tan similar son las predicciones de cada una de los ítems que se muestran respecto al ítem objetivo (Ecuación 5.5), por lo que consideramos que aquellos ítems que al computo de la medida de similitud, excedan este umbral y por tanto sean menos similares en su predicción respecto al ítem que se recomienda, estarán en el lado derecho.

$$d(i, t) \propto \sum_{v \in N} | sg(a, v, i) - sg(a, v, t) |. \quad (5.5)$$

$$\text{Si } d(i, t) = \begin{cases} \leq 1 & \text{Lado izquierdo} \\ > 1 & \text{Lado Derecho} \end{cases}$$

Por otro lado, se han considerado aquellos ítems en los cuales el sistema puede predecir correctamente empleando un umbral de 1 para la medida de acierto (Ecuación 4.2), eje Y, delimitando el gráfico en una parte superior (buenas predicciones individualmente) y una inferior (peores predicciones individuales).

$$\text{Si } h(i) = \begin{cases} \leq 1 & \text{Parte superior} \\ > 1 & \text{Parte inferior} \end{cases}$$

Las características extraídas de la explicación, F_{Ex} , pueden ser computadas para cada una de las diferentes partes anteriormente descritas del gráfico. Con el objetivo de distinguir cada una de las características de dichas partes, emplearemos una tupla $-sh-$ como prefijo, donde en el primer componente $s = \{l, r, x\}$, l representa el lado izquierdo, r el derecho y x representa ambos lados. Similarmente, el componente $h = \{t, b, x\}$, donde t representa la parte superior, b la inferior y x ambas. Así, para una instancia, podemos encontrar

La lista de Schindler (1993)

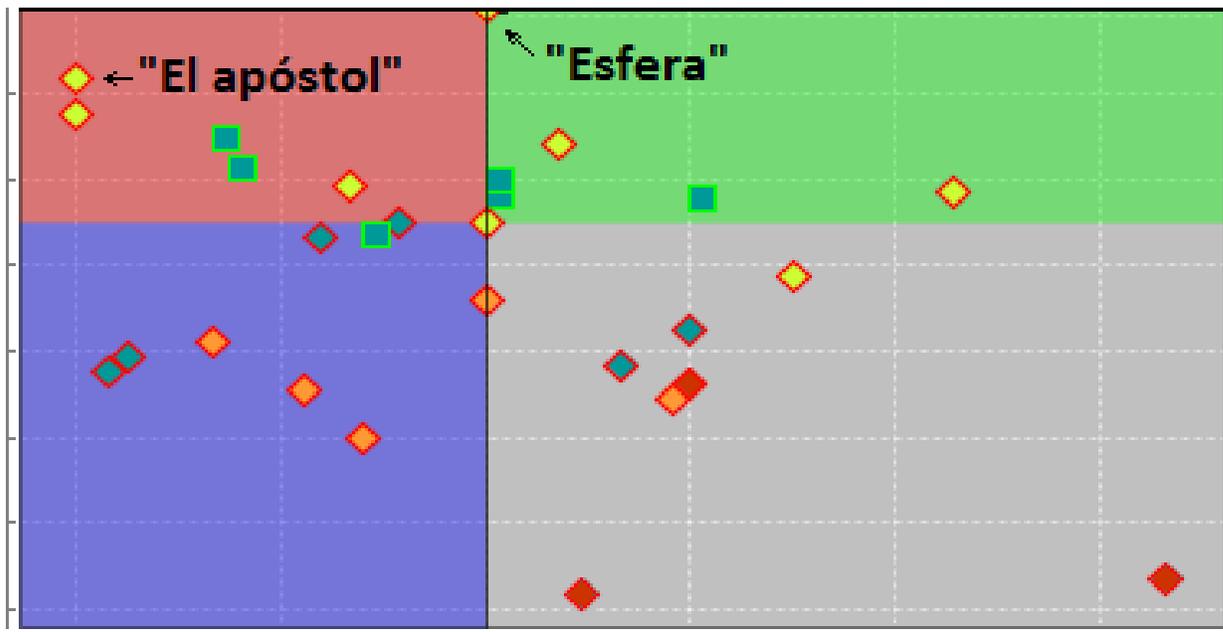


Figura 5.3: Ejemplo de interfaz de explicación.

$lt_\eta(o)$ representando la métrica η computada sobre aquellos ítems en el cuadrante superior izquierdo de la interfaz de explicación (los ítems más similares y mejor sugeridos), $xb_\eta(o)$ representando la métrica enfocadas en la parte inferior, y $xx_\eta(o)$ para denotar la métrica sobre el gráfico en general. En la Figura 5.3 tenemos que $lt_{\#S_o} = 5$, $lb_{\#S_o} = 8$, $rt_{\#S_o} = 5$, $rb_{\#S_o} = 9$, $lx_{\#S_o} = 13$, $rx_{\#S_o} = 14$, $xt_{\#S_o} = 10$, $xb_{\#S_o} = 17$ y $xx_{\#S_o} = 27$.

Por medio de esta propuesta obtendremos un conjunto de 30 características diferentes para cada ítem observado. Obviamente, para el ítem que se analiza calculamos todas las características excepto el error en la predicción que es el elemento que queremos determinar.

Disponer de toda la información analizada hasta aquí, nos permite responder las preguntas realizadas en la sección anterior y determinar si es posible o no aprender de las explicaciones.

5.3. Aprendizaje de la explicación

El estudio con usuarios reales descrito en el capítulo anterior, nos permitió comprobar cómo algunos usuarios, al analizar la información en la interfaz de explicación, eran capaces de corregir el error que se cometía en las recomendaciones.

Al analizar muchos casos como el de la Figura 5.2, consideramos interesantes los resultados que se obtenían, dado que éstos representaban una fuente de información importante para

ser incluidos en un modelo de recomendación. La propuesta que describiremos, cuyo objetivo fundamental es aprender automáticamente el error en la recomendación, aunque utiliza la información que se puede extraer de la explicación, resulta totalmente transparente al usuario.

En esencia, la idea es encontrar el error de la predicción del SR para mejorar la recomendación, enfocándonos en la información obtenida en el proceso de predicción y en la interfaz de explicación (Figura 5.4). Para ello, nos centramos primeramente en la tarea de encontrar una función para determinar el error, $e(t) = f(F_{Ex}, F_{RS})$. Al determinarse el error, éste puede ser empleado para modificar la recomendación, siendo incorporado al valor originalmente sugerido por el sistema, $\hat{r}'(a, t) = \hat{r}(a, t) - e(t)$.

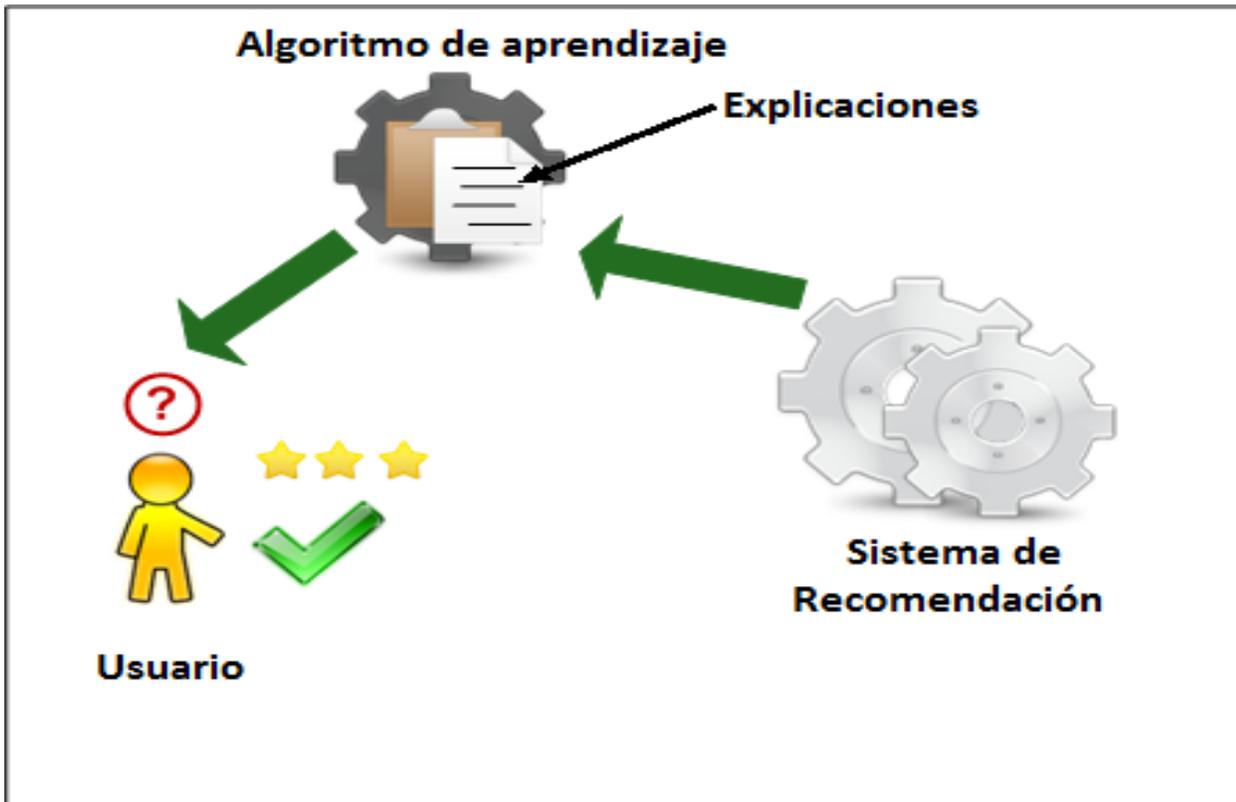


Figura 5.4: Visualización del modelo de aprendizaje del error.

Ahora bien, ¿cómo encontrar la función asociada al error, $f(F_{Ex}, F_{RS})$? Para ello empleamos una aproximación de aprendizaje inductivo que, a partir de un conjunto de ejemplos de entrenamiento (representando el conocimiento específico extraído de las interfaces de explicación), fuese capaz de obtener dicha función. Es importante destacar que en este proceso, en los ejemplos de entrenamiento del conjunto de datos, es conocido el error. El conjunto de entrenamiento lo conformamos al extraer la información de las explicaciones que pueden ser obtenidas de cada uno de los ítem, $o \in \mathcal{S}$. Dado que estos ítems han sido valorados previamente por el usuario activo, se conoce tanto el voto original dado por el

usuario como el que sugiere el sistema, es decir, se conoce además el error. Así, por medio de este procedimiento obtenemos un conjunto de instancias de entrenamiento de la forma $\{e(o) \cup F_{RS}(o) \cup F_{Ex}(o)\}$.

Estas instancias pueden ser empleadas como entradas en un algoritmo de aprendizaje automático, con el error de predicción, e , como una variable dependiente y el conjunto de características siendo las variables independientes, $F_{RS} \cup F_{Ex}$.

5.3.1. Modelo propuesto para predecir el error

Una de las partes que compone el área del aprendizaje automático es el aprendizaje inductivo o aprendizaje por ejemplos. En este campo, el algoritmo trata de inducir una regla general desde un conjunto de instancias observadas. El paradigma de este tipo de técnicas es la clasificación, la cual básicamente consiste de la asignación de una determina instancia a la clase a la que esta pertenece, existiendo una gran cantidad de aproximaciones para esta tarea en concreto.

Dentro de todas estas aproximaciones, se puede encontrar la familia de los árboles de decisión. Un clasificador se construye con la forma de un árbol, donde cada nodo representa un atributo del problema. Para la raíz y los nodos internos, se realizan comprobaciones de los correspondientes atributos para una instancia dada, de manera que a cada rama descendiente se le asigna uno de los valores posibles del atributo. Finalmente, los nodos hojas ofrecen un valor para el atributo clase.

Los árboles de decisión también se expresan en forma de reglas. La regla base se compone de tantas reglas como nodos hojas tenga el árbol. El antecedente está formado por la conjunción de todas las pruebas desde el nodo raíz al nodo anterior a la hoja. En el consecuente, el valor que la variable clase obtiene.

Los árboles de decisión son técnicas de clasificación eficientes, robustas y relativamente simples. Además una de los principales ventajas desde el punto de vista humano es que son entendibles. Se dividen dos grupos de árboles: clasificación o regresión. En los primeros, la clase es una variable discreta, mientras en los segundos, este atributo es continuo. Entonces los nodos hojas almacenan un tipo de promedio para el atributo clase según todas las instancias que alcanzan a este nodo hoja. Dentro de estos clasificadores se encuentran los clásicos algoritmos **ID3** y **C4.5** [92].

Si las hojas almacenan funciones de regresión lineal que predicen el valor de la clase, el árbol se llama árbol de modelo. En este caso podemos encontrar el *M5* [92], como uno de los algoritmos clásicos y que seleccionamos para poner en práctica nuestra propuesta, aunque más específicamente, se emplea el algoritmo *M5'* [121], contenido en la herramienta

de minería de datos Weka ¹. En este algoritmo se mejora la gestión de variables discretas y los valores ausentes, se evidencia además la reducción del tamaño del árbol resultante.

Estos “árboles modelos” son análogos a un conjunto de funciones lineales y su inspección es muy sencilla. El método construye un árbol binario de clasificación donde cada hoja puede predecir el error mediante una regresión lineal. Las predicciones son computadas trazando el camino a una hoja, empleando comprobaciones binarias para una sola característica.

El $M5'$ es un método que al igual que el algoritmo CART (*Classification And Regression Trees*), es utilizado para predecir valores continuos. Los árboles modelos como los generados con el $M5'$, generalmente son más compactos que los árboles de regresión y han demostrado ser más precisos [91].

Siguiendo la propia descripción del autor del algoritmo, Quinlan, el algoritmo $M5$ está compuesto básicamente de los siguientes pasos, asumiendo una colección de T instancias de entrenamiento:

1. Construcción de un árbol binario de búsqueda por medio de una aproximación divide y vencerás. Las instancias de entrenamiento son particionadas hasta que los ejemplos en las hojas formen conjuntos relativamente homogéneos. El criterio al dividir es elegir el atributo que minimiza la variación interna del subconjunto en los valores de la clase para cada uno de los valores de este atributo, es decir, la desviación estándar de los valores de la clase que llegan a un nodo se computa, $sd(T)$; entonces, esta misma medida se computa para cada conjunto T_i , correspondiente a los atributos que este atributo podría tomar, $sd(T_i)$. Finalmente, se selecciona el nodo que maximiza la reducción esperada en el error (diferencia entre $sd(T)$ y $\sum_i sd(T_i)$). Este proceso se repite recursivamente para cada uno de los restantes subconjuntos.
2. Construcción de un modelo lineal para cada uno de los nodos del árbol pero considerando solo los nodos contenidos en el subárbol correspondiente.
3. Computo del error estimado en cada nodo para las instancias no vistas.
4. Simplificación de los modelos lineales en cada nodo, eliminando las variables que tienen una pobre contribución al modelo.
5. Poda del árbol desde las hojas hacía la raíz. Los nodos pueden ser secuencialmente eliminados mientras que el error estimado disminuya (comparando el error del nodo con el obtenido en sus subárboles).
6. Suavizado del proceso para mejorar la precisión de la predicción de los modelos y compensar las discontinuidades que pudieran producirse en los modelos adyacentes en los nodos hojas.

¹<http://www.cs.waikato.ac.nz/ml/weka/>

De este proceso en general, al aplicar el modelo logramos extraer un conjunto de reglas que cumplen el comportamiento que tienen los usuarios, para luego ser aplicadas oportunamente con el fin de mejorar la recomendación que se ofrece a los usuarios.

En la implementación que se puede encontrar del algoritmo $M5'$ en Weka, controlamos los siguientes parámetros: el número mínimo de instancias a considerar en la división, el uso de árboles/reglas sin podar y el uso de suavizado o no en las predicciones.

Primeramente nos gustaría discutir sobre el parámetro de podar o no los árboles en el $M5'$. Al emplear el criterio de podar pueden obtenerse reglas simples, incluso algunas de ellas con una o ninguna variable. Esto se debe a que el error global se reduce después de la poda de la rama de un nodo. Esta situación generalmente funciona bien cuando es necesaria una predicción para una instancia cualquiera, pero este no es nuestro caso. Nosotros ya tenemos una recomendación, y podemos elegir entre actuar sobre esta recomendación o no. De este modo, mediante el uso de la estrategia de no podar, podemos preservar las ramas buenas, aunque el modelo aprendido podría sobreajustarse a los datos que se disponen (en el peor de los casos obteniendo una regla para cada instancia). Luego de múltiples experimentos y con el objetivo de evitar el sobreajuste, definimos un número mínimo determinado de instancia para poder dividir un nodo. En el ejemplo de la Figura 5.5 podemos observar un árbol obtenido con esta estrategia de no podar. Más adelante y según describimos el propio ejemplo, comprobamos la utilidad de esta estrategia para lanzar o no una regla.

En la Figura 5.5, mostramos un árbol obtenido como salida del algoritmo $M5'$ en Weka para el usuario 507 del conjunto de datos MovieLens, generado a partir de un conjunto de 58 instancias de entrenamiento, instancias que representan los ítems que el usuario evaluó. En este ejemplo se pueden destacar las siguientes características:

- Este árbol contiene un total de 4 variables representadas como nodos internos.
- Cada arista del árbol está etiquetada con el correspondiente valor del atributo.
- Las hojas, las cuales corresponden a la variable clase, error (e), o en términos de regresión, la variable que queremos predecir un valor, están etiquetadas con el identificador de un modelo lineal (LMi) aprendido, el número de instancias que se ajuste en el correspondiente nodo hoja (un tipo de soporte de la regla, es decir, el número de instancias que cumplen satisfactoriamente todas las pruebas de la raíz a la hoja) y el porcentaje de ganancia del error en este nodo con respecto a las instancias que soportan la regla.

Se pueden observar 5 reglas (una por cada nodo hoja), compuestas por la conjunción de las pruebas desde la raíz hasta el nodo hoja. Para cada regla, los modelos lineales aprendidos son:

$$\text{LM1: } e = 0,0205 * \text{rx_}\#S - 0,3349$$

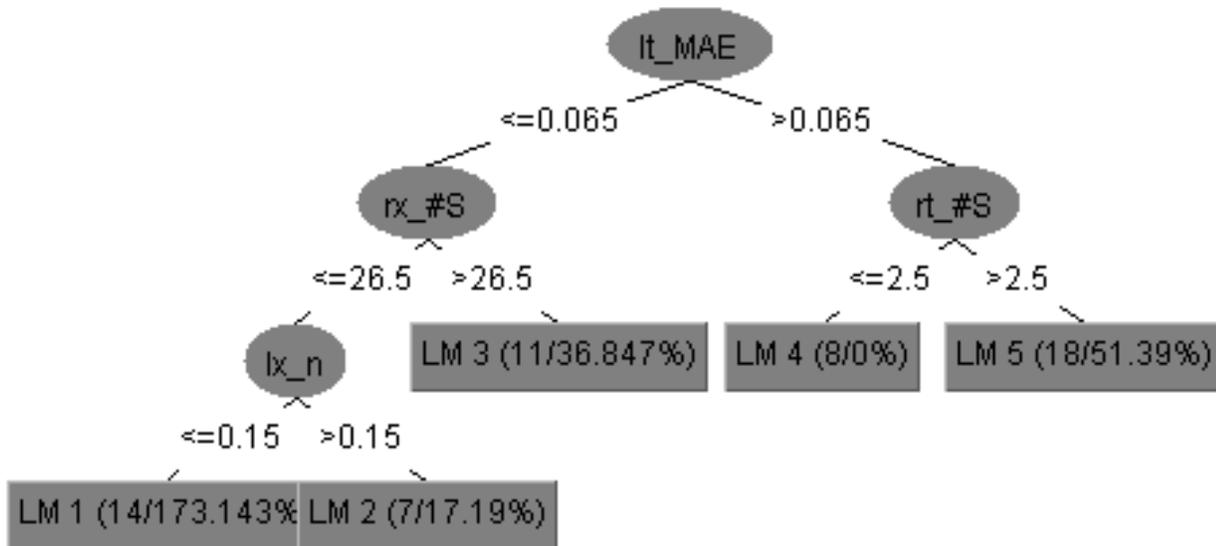


Figura 5.5: Árbol de modelo $M5'$ para el usuario 507

$$\text{LM2: } e = 0,0205 * \text{rx_#S} - 0,2952$$

$$\text{LM3: } e = 0,0249 * \text{rx_#S} - 0,1412$$

$$\text{LM4: } e = 0,0072 * \text{rt_#S} + 0,0103 * \text{rx_#S} - 0,2176$$

$$\text{LM5: } e = 0,0050 * \text{rt_#S} + 0,0103 * \text{rx_#S} - 0,2194$$

Con respecto al porcentaje de ganancia del error obtenido en el nodo hoja, este es la raíz del error cuadrado medio (RMSE) de las instancias en el nodo dividido por la desviación estándar global de la clase. Así, en aquellos casos donde el valor porcentual es inferior al 100%, la estimación que se está realizando se puede considerar buena.

Ahora nos preguntamos ¿cómo vamos a utilizar este modelo de árbol para redefinir la recomendación que se hace a un usuario? Básicamente, cuando una nueva instancia de un usuario llega, se comprueban las distintas ramas con respecto a los valores de los atributos, alcanzando finalmente un nodo hoja. En ese momento, se predice el valor de la variable de error mediante el modelo lineal correspondiente. Entonces la calificación predicha se vuelve a calcular.

Pero no siempre es interesante calcular el error esperado en cada nueva instancia, ya que no en todos los casos la aplicación de este modelo podría mejorar las calificaciones. De hecho, al aplicarse masivamente a todos los casos, es muy probable que se empeore la MAE resultante o no se logren grandes mejoras.

Entonces, tenemos dos factores que analizar para determinar si aplicar o no la regla. El primer caso es cuando el porcentaje de ganancia del error es mayor que el 100%. En

esta situación, conocemos que el modelo lineal aprendido no es bueno para dar un buen rendimiento y, por tanto, la regla no se aplica. En el ejemplo que analizamos, para el usuario 507, se aplicarían solamente las reglas en los modelos LM2, LM3, LM4 y LM5. El segundo caso es cuando el número de instancias que soportan la regla es bajo, entonces nuestra confianza con respecto al porcentaje de la ganancia del error es además baja, ya que hay pocas instancias almacenadas en el correspondiente nodo hoja. En experimentaciones preliminares nos percatamos de que si encontrábamos pocas instancias en un nodo hoja, no se obtenían buenos resultados generales al lanzar la regla, en estos casos aunque se tuviera un buen porcentaje de ganancia, un bajo número de instancias no es un elemento representativo para nosotros. Finalmente consideramos que 10 instancias es un requerimiento aceptable en la experimentación para este criterio. Por lo tanto, después de comprobar los dos criterios anteriores, solamente se aplicarían las reglas LM3 y LM5.

Con el objetivo de ilustrar completamente la idea, se continuará utilizando el ejemplo del usuario 507, donde se mide el desempeño usando una metodología de validación cruzada (*leave one out*) sobre las instancias (cada vez se deja una instancia fuera) que se tienen de cada usuario². Esta metodología de validación cruzada la empleamos pues es la forma más similar a como ocurre el proceso de recomendación para un usuario real, donde se disponen de todas las experiencias previas para aprender y se tiene una nueva experiencia a la que predecirle. Para este usuario en específico, el valor de la MAE general para el sistema de recomendación original es de $MAE = 0,564$ y se logra reducir este valor hasta $0,531$ aplicando el modelo lineal obtenido para todas las instancias (se aplican todas las reglas). Considerando un porcentaje menor o igual al 70% en el momento de aplicar las reglas, solamente se aplicarían 25 reglas y el valor de la MAE descendería hasta $0,485$. Los resultados que se obtendrían serían más significativos al considerar el porcentaje menor o igual al 50%, donde solo se aplicarían las reglas para 14 instancias y en donde el valor de la MAE que se obtiene sería de $0,477$. Al considerar aplicar las reglas para un porcentaje menor o igual al 40%, la MAE se incrementa hasta $0,482$ (se lanzan 10 reglas).

5.3.2. Consideraciones sobre eficiencia

En la propuesta que se ha descrito previamente, son necesarios cálculos adicionales al algoritmo clásico de recomendación, lo que evidentemente afecta el tiempo computacional que se invierte. Por un lado, es necesario computar los valores de las interfaces de explicación relacionados con los ítems que se recomiendan y además con los ítems que forman parte del perfil del usuario. De todas formas, estos cálculos no son complejos y pueden calcularse previamente, de forma que cuando el modelo se genera, el proceso de obtener la predicción final no se ve afectado.

²Nótese que por medio de esta propuesta se aprende un árbol de modelo diferente para cada uno de los ítems.

Cuando el proceso no es simplemente de predicción, sino por ejemplo, ofrecer una lista de artículos, esto sí podría implicar un alto costo computacional, al necesariamente realizar el proceso para todos los artículos de la lista de recomendación que se ofrezca.

5.4. Experimentación

Luego de ilustrar la idea que perseguimos, en esta sección comprobamos la utilidad y beneficios de la misma. En la experimentación consideramos los objetivos y parámetros de configuración que detallamos a continuación:

- **Objetivos de la experimentación:** Nuestro primer objetivo es medir la capacidad de la propuesta para corregir el error en las predicciones ofrecidas por un SR.

En el segundo objetivo nos planteamos identificar las características de la interfaz de explicación que resultan más importantes para corregir el error en las predicciones.

- **Conjuntos de entrenamiento y prueba:** Considerando que en la propuesta no es necesaria la interacción del usuario, se decidió validarla empleando una evaluación empírica sobre el conjunto de datos clásico MovieLens.

Para validar el modelo, siempre que el SR computa una predicción para el usuario activo, se aprende un árbol de decisión de sus experiencias pasadas. Al igual que muchas metodologías de evaluación y debido a que los usuarios no suelen evaluar un gran número de artículos, hemos aplicado una estrategia de validación cruzada dejando cada vez una instancia fuera (*leave one out*) sobre todos los datos de cada usuario.

- **Medidas de evaluación:** Para comprobar los objetivos marcados se consideraron las métricas de precisión: MAE y RMSE.

Debemos destacar que se obtendrán valores globales para estas métricas, y además valores asociados a un subconjunto de ítems, condicionados a que se lancen o no reglas (r). Por ejemplo, en el caso de la MAE, los valores absolutos obtenidos con el modelo de recomendación los identificaremos por MAE, y para el caso en que estén condicionados por un número m de reglas del algoritmo $M5'$, utilizaremos rMAE, siendo el cómputo de la siguiente forma:

$$rMAE = \frac{\sum_{i=1}^m |r(a, i) - (\hat{r}(a, i) + \hat{e}(i))|}{m}. \quad (5.6)$$

Otra métrica que se analiza es la ganancia en MAE que se obtiene al lanzar reglas para corregir el error en las predicciones, calculándose de la siguiente forma:

$$\%G = 100 - \frac{rMAE}{MAE} * 100. \quad (5.7)$$

5.4.1. Resultados y análisis

El primer paso en el diseño experimental para evaluar nuestra propuesta pasa por determinar qué parámetros podemos controlar del algoritmo $M5'$ y los valores que se le asignan. Así, nos encontramos que son dos: por un lado, la poda o no del árbol aprendido, y por otro el número de instancias mínimo necesarias para dividir, tal y como se comentó en la sección anterior. Con respecto al primero, se ha optado por no podar, mientras que para el segundo ($\#I$) proponemos evaluar el método con los valores $\#I = 15$, $\#I = 20$ e $\#I = 25$.

Luego, en el momento de analizar los resultados del modelo (árbol) resultante, se consideraran dos restricciones para lanzar la regla: i) el número de casos en la hoja (soporte de la regla) tiene que ser ≥ 10 y ii) el porcentaje de la ganancia de error tiene que ser menor que un umbral, representado en la experimentación por $\%$.

Se ha considerado además el empleo de diferentes modelos con el objetivo de validar la propuesta. Las aproximaciones seleccionadas emplean ambos enfoques de filtrado colaborativo y han sido descritas en detalle en el Capítulo 3 que aborda los modelos colaborativos. Una de las propuestas, **BL** [50], selecciona los mejores vecinos empleando la correlación de Pearson, mientras la segunda de las alternativas, **LM**, emplea una función de costo asociada a las predicción de las experiencias previas³.

De forma general, en la experimentación se intenta mejorar el desempeño de los modelos de recomendación. En las secciones siguientes, comprobamos por un lado, la capacidad de estimar el error cometido en las predicciones, y finalmente, las principales características que inciden en esta predicción.

5.4.1.1. Corrección del error en las predicciones para todos los usuarios

En este primer experimento, intentamos corregir el error en las predicciones del SR aprendiendo de las interfaces de explicación. Esta corrección la hacemos a todos los usuarios, y para cada uno de ellos comprobaremos todos sus ítems.

Al inicio del capítulo comentamos la idea de utilizar solamente la información de F_{RS} en un simple algoritmo de regresión lineal. Antes de analizar los resultados del $M5'$, mostramos brevemente los resultados obtenidos con el algoritmo de RL.

Los resultados de esta experimentación se reflejan en la Figura 5.6, donde se muestra como varía el desempeño en todos los usuarios en términos de mejora de la MAE, es decir, cuánto se mejoran los resultados del modelo de recomendación luego de aplicar la estrategia anterior. Como podemos observar no para todos los usuarios se logra mejorar en el mismo sentido, de

³Para mayores detalles, dirigirse a la sección 3.1 del capítulo 3.

943 usuarios del conjunto de datos que utilizamos, en 388 las mejoras son positivas y en 555 son negativas. De todas formas, aunque ligeramente, se logra mejorar en computo general la MAE, con el modelo de recomendación tenemos que la $MAE_{Modelo} = 0,7176$ y al intentar predecir el error cometido en la predicción, logramos que la $MAE_{Modelo+RL} = 0,7144$.

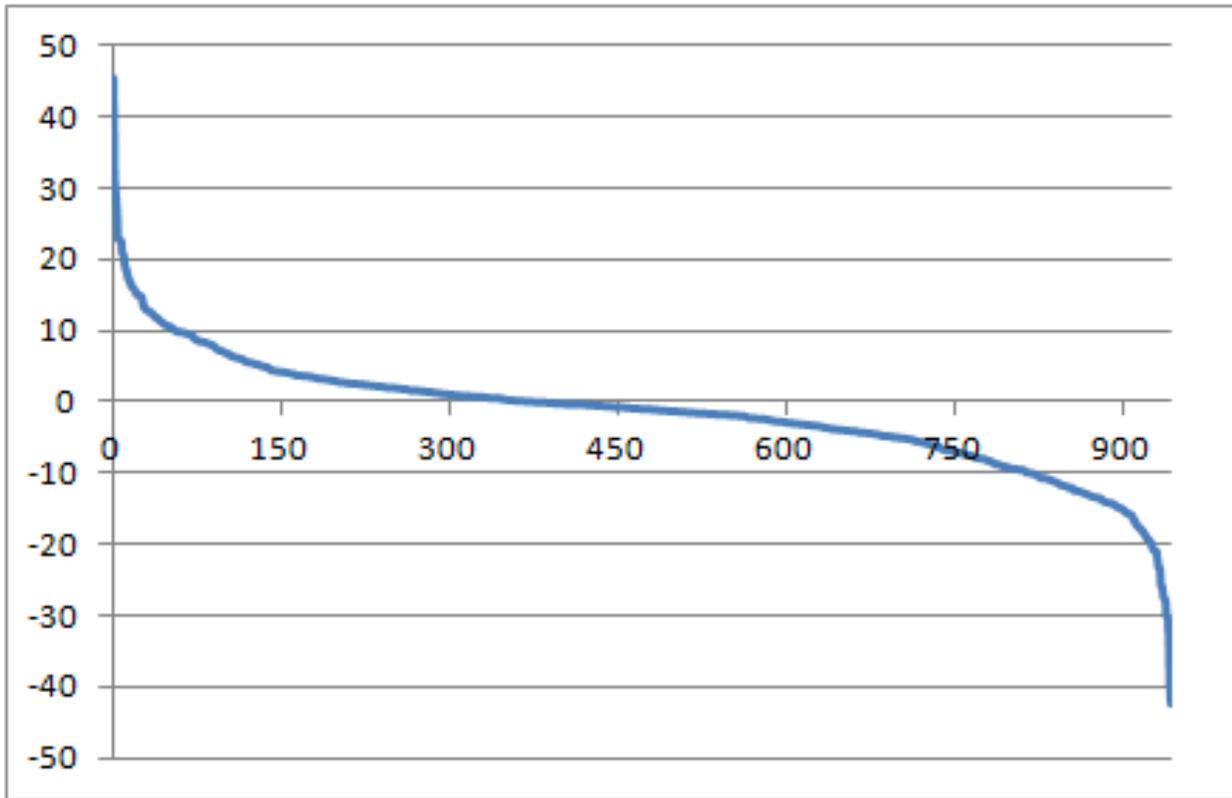


Figura 5.6: Mejora de la MAE al aplicar un simple algoritmo de regresión lineal.

Ya con el algoritmo $M5'$ y para ejemplificar el comportamiento que se obtiene, nos enfocaremos en los resultados para la configuración de $\#I = 20$ y el modelo LM, aunque las tendencias son similares en los resultados con las restantes configuraciones ($\#I = 15$ y $\#I = 25$). La Tabla 5.3 muestra los resultados obtenidos cuando condicionamos en el momento de lanzar o no la regla, el umbral que debe sobrepasar el porcentaje de la ganancia de error (%). En la tabla mostramos en la primera columna el número de usuarios a los que se le lanzan reglas, en la segunda columna la MAE general del modelo (MAE) y en la tercera la MAE que obtenemos después de predecir el error e intentar corregir la predicción (MAE_m). La cuarta y última columna muestra el número total de reglas que se lanzan. Podemos observar que la MAE general ($MAE = 0,718$) se reduce hasta $MAE_m = 0,705$ según se usa un % de umbral mayor para lanzar las reglas. Al mismo tiempo, según aumentamos el % de umbral, podemos observar cómo se incrementa el número de usuarios a los que se le intenta corregir el error en la predicción. Este valor, como comprobaremos más adelante, garantiza

un cierto nivel de calidad en cuanto a determinar qué regla lanzar o no. Los resultados, en general, pueden ser considerados no tan buenos, pero cuando analizamos más detenidamente los datos, observamos que el rendimiento entre usuarios difiere considerablemente.

Tabla 5.3: Resultados de la MAE y $rMAE$ sobre las predicciones en todos los usuarios, del modelo basado en las predicciones pasadas, LM.

#I=20				
%	Usuarios	MAE	MAE_m	Reglas
10	77	0,718	0,718	985
20	221	0,718	0,717	3854
30	416	0,718	0,715	8605
40	620	0,718	0,713	14886
50	748	0,718	0,710	22483
60	838	0,718	0,708	31276
70	879	0,718	0,707	40933
80	898	0,718	0,706	50320
90	907	0,718	0,705	58423

Así, la Figura 5.7 muestra cómo el desempeño varía con los usuarios (en término de mejora de la MAE). En el eje horizontal se tienen todos los usuarios analizados y en el eje vertical, mostramos el porcentaje de mejora que se logra para cada uno de los usuarios. En la gráfica mostramos los resultados para $\% = 20, 40, 60, 80$ y se puede notar la diferencia en la cantidad de usuarios a los que se le lanzan reglas en cada configuración. Mientras para $\% = 20$ a 221 usuarios se le lanzan reglas, para $\% = 80$ este número alcanza los 907 usuarios (Tabla 5.3).

Los resultados anteriores nos confirman que sí es posible aprender de las interfaces de explicación y mejorar la recomendación. Sin embargo, depende altamente del usuario y de sus experiencias previas, por lo tanto, puede ser detectado de antemano. Podemos detectar un grupo de usuarios para los cuales la información en las interfaces de explicación son beneficiosas para corregir las predicciones, y por tanto, nuestra propuesta es muy útil, mientras para otros usuarios al emplearla se empeoran las predicciones dadas por el SR. Esto puede explicar por qué ha sido tan difícil de entender/medir los efectos de las explicaciones en los SR.

5.4.1.2. Desempeño de las predicciones en usuarios candidatos a corregir el error

Como consecuencia del análisis anterior, en lo adelante, los valores de desempeño que se presentan, se obtienen considerando simplemente aquellos usuarios con al menos una regla lanzada y para los cuales a priori la información en las interfaces de explicación es beneficiosa

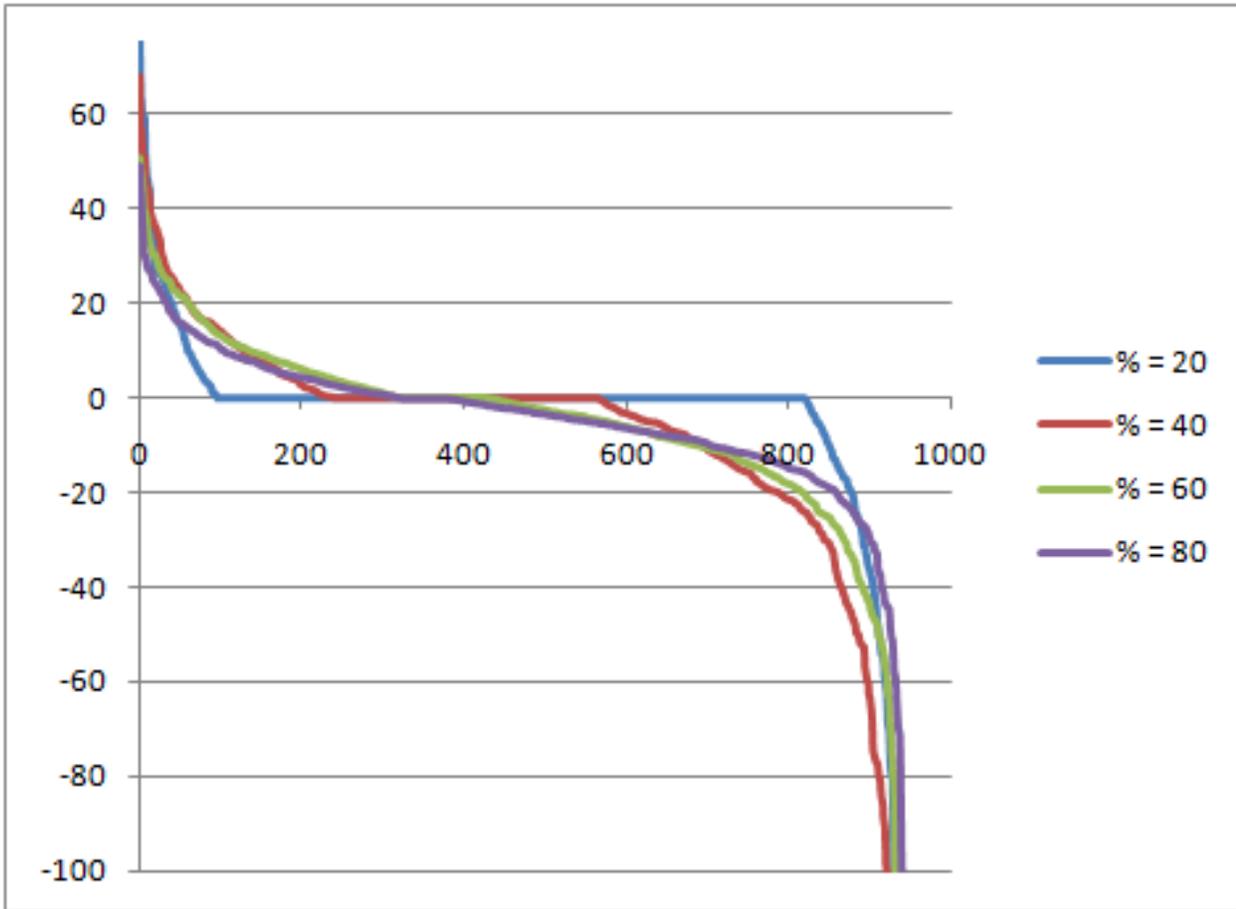


Figura 5.7: Mejoras en la MAE para ($\#I = 20$).

para corregir las predicciones (donde se obtienen ganancias positivas), por lo tanto, nos enfocamos en los beneficios de nuestra propuesta al ser aplicada sólo a un número concreto de usuarios, descartando aquellos en los que en la experimentación anterior se obtuvieron ganancias negativas.

Por ejemplo, en la sección anterior, encontramos en la Tabla 5.3 para $\% = 10$ que se tienen 77 usuarios a los que se le lanzan reglas, de ese número de usuarios, solamente en 34 las ganancias son positivas, siendo productivo aplicar en ellos nuestra propuesta y por tanto es en ese grupo de usuarios donde centraremos el análisis de esta sección.

Retomaremos nuevamente el empleo del algoritmo de RL simple para destacar el grupo de usuarios a los que les resulta beneficiosa la información de las interfaces y en los que se logran mejoras. Lo primero es que este grupo de 388 usuarios, valoró el 52,52% de instancias en el conjunto de datos, por lo que al aplicar la estrategia de RL simple lograríamos la mejora a un grupo importante de usuarios. En términos de MAE, se logra mejorar de $MAE_{Modelo} = 0,7110$ a $MAE_{Modelo+RL} = 0,6828$.

Para evaluar este aspecto con el algoritmo $M5'$, se decidió no lanzar reglas para todos

los usuarios donde el sistema no funciona y simplemente sugerir la recomendación original. Por medio de este criterio restringimos la siguiente experimentación para un subconjunto de usuarios.

Consideramos que estos resultados son buenos, porque para un importante porcentaje de usuarios el sistema mejora los resultados, mientras que el resto de los usuarios recibe las recomendaciones originales del SR. Con respecto al algoritmo $M5'$, hemos mostrado que este proceso puede ser muy provechoso para entre el 25 % y 35 % de los usuarios en el conjunto de datos empleado (véase Figura 5.7). Ciertamente el hecho de que para algunos usuarios los resultados sean mejores que para otros, es una temática en la que se tiene que seguir investigando.

Tabla 5.4: Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=15$, en modelo LM.

%	Usuarios	Ítems	Reglas	G10
10	48	11439	419	54,17
20	142	30223	1340	65,49
30	209	37607	2400	57,89
40	263	39034	4307	60,08
50	313	41964	7576	52,72
60	329	41616	11091	50,76
70	350	44383	15211	43,14
80	337	41079	16934	38,28
90	344	39989	18623	35,47

Tabla 5.5: Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=20$, en modelo LM.

%	Usuarios	Ítems	Reglas	G10
10	34	7604	319	67,65
20	101	20940	1088	62,38
30	168	32024	2109	63,69
40	244	38969	4227	54,92
50	289	44090	8533	55,02
60	326	45018	13489	42,64
70	334	43839	18022	39,82
80	340	44499	22690	32,06
90	334	41174	24654	32,34

Las Tablas 5.4, 5.5 y 5.6 muestran el número de usuarios involucrados en las predicciones para $\#I = 15$, $\#I = 20$ e $\#I = 30$, respectivamente, para el modelo LM. En cada tabla,

Tabla 5.6: Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=25$, en modelo LM.

%	Usuarios	Ítems	Reglas	G10
10	26	5850	237	53,85
20	75	16663	781	69,33
30	155	28962	1684	60,65
40	225	36687	3680	52,89
50	282	44189	7832	50,00
60	312	46070	13013	41,35
70	306	44074	17982	35,95
80	311	43998	23382	25,72
90	311	41499	26475	27,01

la primera columna representa, como hasta ahora, el umbral que debe sobrepasar el porcentaje de la ganancia de error (%). La segunda columna (Usuarios) muestra el número de usuarios en los que se lanza al menos 1 regla, la tercera columna (Ítems) presenta el número de ítems que han sido evaluados originalmente por los usuarios de la columna anterior, y la cuarta columna, el número de reglas lanzadas (Reglas), es decir, ítems en los que las recomendaciones originales del sistema han sido corregidas. La quinta columna (G10) presenta el porcentaje de los usuarios de la segunda columna que obtuvieron mejoras en las recomendaciones (mayores que el 10 %) después de aplicar las reglas (Reglas). Estos resultados junto con los de las Tablas 5.7, 5.8 y 5.9, permiten en su conjunto entender mejor el desempeño del sistema. Al enfocarnos en $\#I = 20$ (los resultados para otras alternativas son similares) y para un bajo % de umbral se obtienen muy buenos resultados, pero esto solo se aplica para un bajo número de usuarios. Si se incrementa el valor del % de umbral al 50 % esto permite que se aplique el modelo a un mayor número de usuarios (289, cerca de un 30 % de los usuarios del conjunto original de datos empleado, cuyas votaciones representan el 44 % de las valoraciones (44090)) con mejoras importantes.

Cuando el % es mayor que 70, el desempeño se comporta más errático porque se comienzan a lanzar reglas malas. Como consecuencia, encontramos usuarios en los que el desempeño es adecuado por debajo de este umbral pero cuando estas reglas se aplican, su desempeño decrece, siendo posible que se excluyan de las pruebas (dado que muestran en el global un desempeño negativo). Por ejemplo, imaginemos un usuario cuyo desempeño es muy bueno hasta % de 60 en 10 reglas lanzadas al corregir errores pequeños, y cuando aumentamos hasta 70 el % de umbral, se lanzan dos nuevas reglas al intentar corregir errores grandes, que finalmente resultan reglas malas. En este ejemplo observamos como las dos últimas reglas provocan que se excluya un usuario que en los % anteriores se lograban mejorar las recomendaciones en computo general. Por otro lado, nuevos usuarios son incluidos por el modelo aprendido al aumentar el % de umbral. Es decir, igualmente tendremos usuarios en

Tabla 5.7: Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo LM para $\#I=15$.

%	Usuarios	MAE	$rMAE$	%G
10	48	0,532	0,398	25,1
20	142	0,587	0,454	22,5
30	209	0,624	0,491	21,2
40	263	0,683	0,558	18,3
50	313	0,690	0,594	14,0
60	329	0,708	0,626	11,7
70	350	0,721	0,653	9,4
80	337	0,725	0,661	8,8
90	344	0,729	0,669	8,2

Tabla 5.8: Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo LM para $\#I=20$.

%	Usuarios	MAE	rMAE	%G
10	34	0,464	0,342	26,3
20	101	0,584	0,440	24,6
30	168	0,620	0,481	22,4
40	244	0,684	0,558	18,4
50	289	0,688	0,593	13,8
60	326	0,699	0,621	11,1
70	334	0,718	0,652	9,1
80	340	0,723	0,669	7,6
90	334	0,721	0,666	7,7

los que para un % determinado no se lanzan reglas y cuando se aumenta dicho %, nuevas reglas se comienzan a lanzar.

Similares tendencias respecto a las obtenidas anteriormente con respecto al % de umbral en las reglas lanzadas, las obtenemos con el modelo propuesto por Herlocker et al. [50]. Las Tablas 5.10, 5.11 y 5.12 muestran estas tendencias. Aunque podemos observar que en el modelo basado en las predicciones pasadas son necesarios muchos menos usuarios, ítems y reglas para obtener grandes mejoras.

Centrándonos ahora en los casos donde se decide lanzar las reglas, las Tablas 5.7, 5.8, 5.9, 5.13, 5.14 y 5.15 muestran los resultados obtenidos cuando se consideran diferentes números de ítems en la estrategia de dividir y diferentes valores de % de umbral para los modelos empleados. En las tablas se muestra el número de usuarios a los que se le lanzan reglas, la MAE que obtenemos con el modelo original del SR (MAE), la MAE empleando nuestra propuesta ($rMAE$) y las mejoras (en términos de porcentaje, %G).

Cuando analizamos los resultados de cada configuración, con respecto a $\#I$ no hay un

Tabla 5.9: Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo LM para $\#I=25$.

%	Usuarios	MAE	rMAE	%G
10	26	0,493	0,390	20,9
20	75	0,605	0,440	27,3
30	155	0,605	0,475	21,6
40	225	0,669	0,535	20,1
50	282	0,679	0,582	14,3
60	312	0,695	0,619	11,0
70	306	0,699	0,637	8,9
80	311	0,713	0,661	7,2
90	311	0,709	0,659	7,0

Tabla 5.10: Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para $\#I=15$, en el modelo BL.

%	Usuarios	Ítems	Reglas	G10
10	74	15287	861	62,16
20	174	32208	1948	64,37
30	255	42018	3267	66,27
40	335	47259	5377	62,69
50	389	50078	8778	61,70
60	414	50835	12908	56,04
70	421	47359	15986	50,59
80	440	49311	20028	45,91
90	439	47079	21880	46,47

claro ganador, pero enfocándonos en el % de umbral, podemos encontrar que según se incrementa este valor, el error aumenta: nosotros lanzamos reglas con menor calidad pero, por otra parte, se incrementa el número de usuarios en los cuales se pueden obtener los mismos beneficios. Un equilibrio entre estos dos criterios puede lograrse de forma que se obtenga un número de usuarios satisfechos pero al mismo tiempo una alta mejora de los resultados.

En nuestra opinión un buen balance es considerar un umbral alrededor del 50-60%. Sin embargo pensamos que mejores resultados se obtendrán si fuésemos capaces de identificar el mejor umbral para cada usuario.

Evaluando con la métrica RMSE en lugar de con la MAE, las Tablas 5.16 y 5.17 muestran los valores obtenidos de dicha medida para los diferentes valores de umbrales y los modelos empleados. Similar a las tablas anteriores, para cada $\#I$ tenemos en la primera columna la RMSE que obtenemos con el modelo original del SR ($RMSE$), la RMSE obtenida empleando nuestra propuesta ($rRMSE$) y las mejoras (en términos de porcentaje, %G).

El desempeño que obtenemos es similar al obtenido con la MAE en el primero de los

Tabla 5.11: Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para #I=20, en el modelo BL.

%	Usuarios	Ítems	Reglas	G10
10	64	12652	777	64,06
20	142	27089	1718	70,42
30	207	36855	2909	62,80
40	297	45472	5283	59,60
50	365	48971	9029	54,79
60	387	48719	13940	46,51
70	401	48506	19401	46,88
80	425	49329	24692	38,59
90	412	46980	27740	41,02

Tabla 5.12: Número de reglas lanzadas en usuarios a los que se le aplica la propuesta para #I=25, en el modelo BL.

%	Usuarios	Ítems	Reglas	G10
10	53	10402	421	50,94
20	127	23244	1307	59,84
30	175	30257	2304	62,29
40	262	41266	4451	59,92
50	322	46921	8073	54,35
60	344	47012	13171	45,64
70	365	48579	19650	44,66
80	385	47905	24940	38,18
90	377	46613	29196	35,01

modelos, aunque las mejoras son menos consistentes. En nuestra opinión esto se debe al hecho de que la RMSE penaliza los grandes errores y éstos son más difíciles de descubrir por el algoritmo $M5'$ (Este método está cercanamente ligado a las características del conjunto de datos). En nuestro caso, no existe un alto porcentaje de grandes errores en el conjunto de datos empleado. Particularmente el número de predicciones con un error absoluto mayor o igual que 2 es igual a 0,5%. Por lo tanto, sería interesante utilizar un enfoque de aprendizaje diferente, el cual sea capaz de trabajar con conjuntos de datos no balanceados y usar la métrica RMSE, en lugar de la MAE, como función objetivo de los algoritmos.

5.4.1.3. Análisis de la importancia de las características de la interfaz de explicación

Finalmente, presentamos el análisis realizado con el objetivo de identificar aquellas variables que resultan más relevantes para las predicciones del error en las recomendaciones. En la experimentación en esta sección nos centramos en identificar aquellas características de la

Tabla 5.13: Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo BL para $\#I=15$.

%	Usuarios	MAE	rMAE	%G
10	74	0,381	0,308	19,1
20	174	0,488	0,357	26,8
30	255	0,551	0,418	24,2
40	335	0,597	0,477	20,1
50	389	0,631	0,527	16,4
60	414	0,638	0,553	13,4
70	421	0,640	0,565	11,8
80	440	0,643	0,578	10,1
90	439	0,647	0,583	9,9

Tabla 5.14: Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo BL para $\#I=20$.

%	Usuarios	MAE	rMAE	%G
10	64	0,317	0,239	24,7
20	142	0,435	0,322	26,1
30	207	0,491	0,375	23,7
40	297	0,557	0,445	20,1
50	365	0,595	0,495	16,7
60	387	0,626	0,543	13,3
70	401	0,623	0,553	11,2
80	425	0,629	0,569	9,5
90	412	0,639	0,581	9,0

interfaz de explicación sobre las que mayor atención debemos prestar.

Como mencionamos anteriormente, en los estudios experimentales con usuarios reales, nos percatamos del interés en determinadas regiones o partes de la explicación, lo que nos llevó a pensar que podría haber diferencias si en el aprendizaje no se empleaban todas las variables sino algunas de ellas. Así, en esta última experimentación pretendemos determinar cuáles son las variables que juegan un papel más importante en la calidad del proceso de aprendizaje.

Para analizar cada una de las características de la interfaz, creamos un conjunto de datos asociado a cada una de las 30 características (excepto el error) que consideramos de la explicación. La idea de estos conjuntos de datos es que cada uno contiene toda la información excepto la característica a la que está asociado, de esa forma y como veremos en lo adelante, verificamos cuánto influye la variable en los resultados generales al predecir el error.

Además de los conjuntos de datos asociados a cada característica individualmente, y con el objetivo de comprobar la relevancia de algunas características relacionadas, empleamos los

Tabla 5.15: Resultados de la MAE y $rMAE$ sobre las predicciones en usuarios a los que se le aplica la propuesta, modelo BL para $\#I=25$.

%	Usuarios	MAE	rMAE	%G
10	53	0,393	0,285	27,4
20	127	0,453	0,335	26,0
30	175	0,488	0,360	26,1
40	262	0,541	0,422	22,0
50	322	0,591	0,492	16,7
60	344	0,608	0,528	13,2
70	365	0,617	0,550	10,8
80	385	0,622	0,564	9,4
90	377	0,639	0,585	8,5

Tabla 5.16: Valores de RMSE sobre las predicciones en usuarios a los que se le aplica la propuesta, para el modelo LM.

	#I=15			#I=20			#I=25		
%	RMSE	rRMSE	%G	RMSE	rRMSE	%G	RMSE	rRMSE	%G
10	0,809	0,73	9,966	0,736	0,687	6,66	0,803	0,76	4,692
20	0,806	0,74	8,651	0,798	0,732	8,27	0,824	0,74	9,737
30	0,832	0,75	10,22	0,828	0,750	9,42	0,827	0,75	9,331
40	0,892	0,81	9,203	0,889	0,811	8,77	0,875	0,79	9,963
50	0,89	0,83	7,289	0,892	0,831	6,84	0,881	0,82	7,207
60	0,905	0,85	6,116	0,898	0,846	5,79	0,893	0,84	5,578
70	0,916	0,87	5,206	0,914	0,870	4,81	0,895	0,85	4,519
80	0,922	0,87	5,09	0,918	0,882	3,92	0,908	0,87	3,796
90	0,927	0,88	4,971	0,919	0,878	4,46	0,905	0,87	3,957

conjuntos de datos resultantes de:

- considerar todas las características excepto aquellas variables relacionadas con la MAE , es decir, aquellas variables computadas en las predicciones en cada una de las cuatro partes en las cuales dividimos la interfaz de explicación bidimensional, Figura 5.3 (identificadas como $_MAE$);
- considerar las mismas variables que en el punto anterior pero eliminando las variables relacionadas con $\eta(o)$ (identificadas como $_\eta$);
- considerar el voto recomendado $\hat{r}(a, o)$, el error de la predicción, $e(o)$, y la entropía $H(o)$ como únicas características. En este conjunto de datos no se incluye ninguna

Tabla 5.17: Valores de RMSE sobre las predicciones en usuarios a los que se le aplica la propuesta, para el modelo BL.

	#I=15			#I=20			#I=25		
%	RMSE	rRMSE	%G	RMSE	rRMSE	%G	RMSE	rRMSE	%G
10	0,736	0,687	6,66	0,539	0,503	6,74	0,603	0,534	11,5
20	0,798	0,732	8,27	0,671	0,593	11,6	0,698	0,614	12
30	0,828	0,750	9,42	0,706	0,631	10,7	0,710	0,619	12,8
40	0,889	0,811	8,77	0,753	0,677	10,1	0,739	0,655	11,4
50	0,892	0,831	6,84	0,779	0,713	8,46	0,776	0,714	7,99
60	0,898	0,846	5,79	0,806	0,750	6,95	0,787	0,735	6,56
70	0,914	0,870	4,81	0,801	0,752	6,15	0,795	0,750	5,63
80	0,918	0,882	3,92	0,806	0,763	5,34	0,797	0,757	5,06
90	0,919	0,878	4,46	0,817	0,771	5,61	0,819	0,779	4,93

información de nuestra propuesta de explicación bidimensional, e intentamos comprobar la importancia individual del histograma de votos del vecindario; y

- considerar todas las características e información disponible.

La idea al utilizar los dos últimos conjuntos de datos es, con el penúltimo conjunto comprobar la importancia a la vez de todas las características extraídas de la explicación, F_{Ex} , y con el último poder comparar los resultados que se obtienen con todos los conjuntos de datos, contra un resultado estándar donde están todas las características.

Finalmente, y considerando i) el modelo basado en las predicciones previas, LM, y ii) la configuración para un mínimo número de instancias $\#I = 20$ al lanzar las reglas, los resultados obtenidos con el objetivo de comprobar la importancia de cada una de las características o grupo de ellas, se pueden ver en la Tabla 5.18. En la primera columna se tienen las características asociadas a cada conjunto, mientras las restantes columnas representan el cómputo de los siguientes valores:

1. La segunda columna (Usuarios), el número de usuarios a los que se le intenta corregir el error y por tanto se le lanzan reglas.
2. La tercera columna (Reglas), el número total de reglas lanzadas y por tanto, número de ítems en los que se intenta corregir el error.
3. La cuarta columna (MAE), la MAE original con el modelo colaborativo empleado, pero solamente en los usuarios seleccionados en el primer punto.
4. La quinta columna ($rMAE$), la MAE resultante luego de lanzar las reglas con el modelo $M5'$, pero solamente en los usuarios seleccionados en el primer punto.

Tabla 5.18: Resultados generales al comprobar la importancia de las características al predecir el error de las predicciones.

Características	Usuarios	Reglas	MAE	$rMAE$	%G	%aG
r, e, H	232	3459	0,6844	0,5807	15,1536	1,2442
* \underline{MAE} y $\underline{\eta}$	287	6008	0,6965	0,5991	13,9806	1,8778
\underline{MAE}	271	6143	0,6929	0,5948	14,1568	2,0630
r	288	7975	0,6819	0,5991	12,1490	2,1364
H	304	8212	0,6793	0,5910	13,0041	2,2962
$\underline{\eta}$	294	7709	0,6858	0,5927	13,5749	2,3669
$rx \underline{\#S}$	291	8413	0,6834	0,5922	13,3441	2,4523
$lb \underline{MAE}$	297	8574	0,6896	0,5963	13,5262	2,4638
$xb \underline{\eta}$	295	8625	0,6949	0,6013	13,4681	2,4811
$lt \underline{\#S}$	288	8332	0,6898	0,5956	13,6470	2,4891
$lx \underline{\#S}$	288	8374	0,6914	0,5989	13,3820	2,4913
$xt \underline{MAE}$	292	8212	0,6879	0,5926	13,8493	2,4935
$rt \underline{\eta}$	293	8710	0,6794	0,5871	13,5898	2,5044
$lt \underline{\eta}$	290	8583	0,6914	0,5980	13,5068	2,5068
$lb \underline{\eta}$	288	8252	0,6813	0,5881	13,6825	2,5074
$rb \underline{MAE}$	290	8291	0,6938	0,5978	13,8423	2,5195
$rx \underline{MAE}$	286	8518	0,6846	0,5904	13,7539	2,5274
$lb \underline{\#S}$	296	8424	0,6920	0,5968	13,7475	2,5294
$rb \underline{\eta}$	290	8445	0,6915	0,5969	13,6792	2,5394
$rt \underline{\#S}$	295	8667	0,6880	0,5928	13,8369	2,5432
$xx \underline{MAE}$	285	8309	0,6936	0,5969	13,9434	2,5540
All features	289	8533	0,6880	0,5930	13,8081	2,5633
$rb \underline{\#S}$	293	8505	0,6939	0,5973	13,9173	2,5681
$lx \underline{\eta}$	293	8523	0,6845	0,5894	13,8862	2,5718
$xt \underline{\eta}$	286	8316	0,6856	0,5902	13,9102	2,5727
$lt \underline{MAE}$	288	8208	0,6832	0,5873	14,0407	2,5749
$lx \underline{MAE}$	293	8360	0,6926	0,5954	14,0449	2,5790
$rt \underline{MAE}$	297	8479	0,6898	0,5920	14,1702	2,5898
$xb \underline{\#S}$	280	8082	0,6880	0,5908	14,1205	2,5899
$xx \underline{\#S}$	299	8529	0,6932	0,5950	14,1758	2,5931
$xb \underline{MAE}$	292	8280	0,6988	0,5989	14,3005	2,6045
$xt \underline{\#S}$	289	8173	0,6882	0,5905	14,1990	2,6227
$xx \underline{\eta}$	299	8575	0,6913	0,5934	14,1646	2,6382
$rx \underline{\eta}$	289	8379	0,6895	0,5900	14,4314	2,6822

- La sexta columna ($\%G$), representa la ganancia de la MAE en los usuarios a los que se le intenta corregir el error en la predicción, es decir, la ganancia de $rMAE$ con respecto a la MAE para aquellos ítems donde las reglas son lanzadas.
- La séptima y última columna ($\%aG$), representa la ganancia absoluta de la MAE en

los usuarios a los que se le intenta corregir el error en la predicción, es decir, la ganancia de aplicar el modelo en general para todos los ítems de los usuarios seleccionados en el primer punto.

En la Tabla 5.18, ordenada crecientemente por la última columna, encontramos en la primera fila los resultados sobre el conjunto de datos en el que solo se tienen los atributos r , e , y H . El conjunto de datos donde se emplean todas las características, lo identificamos con “*All features*”, y el resto de conjuntos lo identificamos en la tabla por las características ausentes en ellos.

Con el objetivo de comentar los resultados obtenidos y no repetir las observaciones realizadas en experimentos previos, mostramos algunas gráficas que nos ayudarán a apreciar las principales conclusiones del estudio. Primeramente, y enfocándonos solo en el número de reglas lanzadas (tercera columna de la Tabla 5.18), en el gráfico de la Figura 5.8 se representa un histograma ordenado decrecientemente para cada conjunto de datos. Además, se puede observar en el eje Y, el nombre de las variables simples que han sido eliminadas del conjunto total de características (barras en color azul), los grupos de variables eliminadas (barras en color amarillo), la combinación del voto, el error y la entropía (barra en color verde), y el conjunto completo de características (barra en color naranja); mientras en el eje X se muestra el número de reglas lanzadas.

Desde una perspectiva general, podemos decir que cuando nosotros eliminemos una característica del conjunto original de características, cuanto menor sea el número de reglas que se lanzan, mayor será la importancia de dicho atributo. Observando el gráfico, si comparamos los resultados que se obtienen con el conjunto de datos donde están presentes todas las características, podemos decir que las características relacionadas con la MAE y con $\eta(o)$ son los atributos más relevantes, dado que es notable la reducción del número de reglas que se lanzan (en especial al eliminar todas las variables relacionadas con la MAE y(o) todas las relacionadas con la $\eta(o)$).

Enfocándonos en las características individuales, se puede observar que el voto, el número de ítems en las partes inferiores y superiores, así como la entropía, son las variables más importantes. Y los atributos como $rt_ \eta$ y $rt_ \#S$ son los menos relevantes (incluso se incrementa el número de reglas que se lanzan). Considerando estos elementos comprobamos la importancia y relevancia de la interfaz de la explicación. Los resultados nos confirman nuestras sospechas al observar a los usuarios en la evaluación de las interfaces de explicación. Por una parte, los usuarios prestan mucha atención a la distribución de los ítems en la explicación, mientras por otra parte, el histograma de votos se convierte en un rápido referente para decidir si aceptar o no la recomendación a pesar de no poseer información adicional como la que se obtiene en nuestra interfaz de explicación. El que aparezca el voto como un atributo relevante, nos indica que la opinión del usuario es esencial para la personalización en los SR.

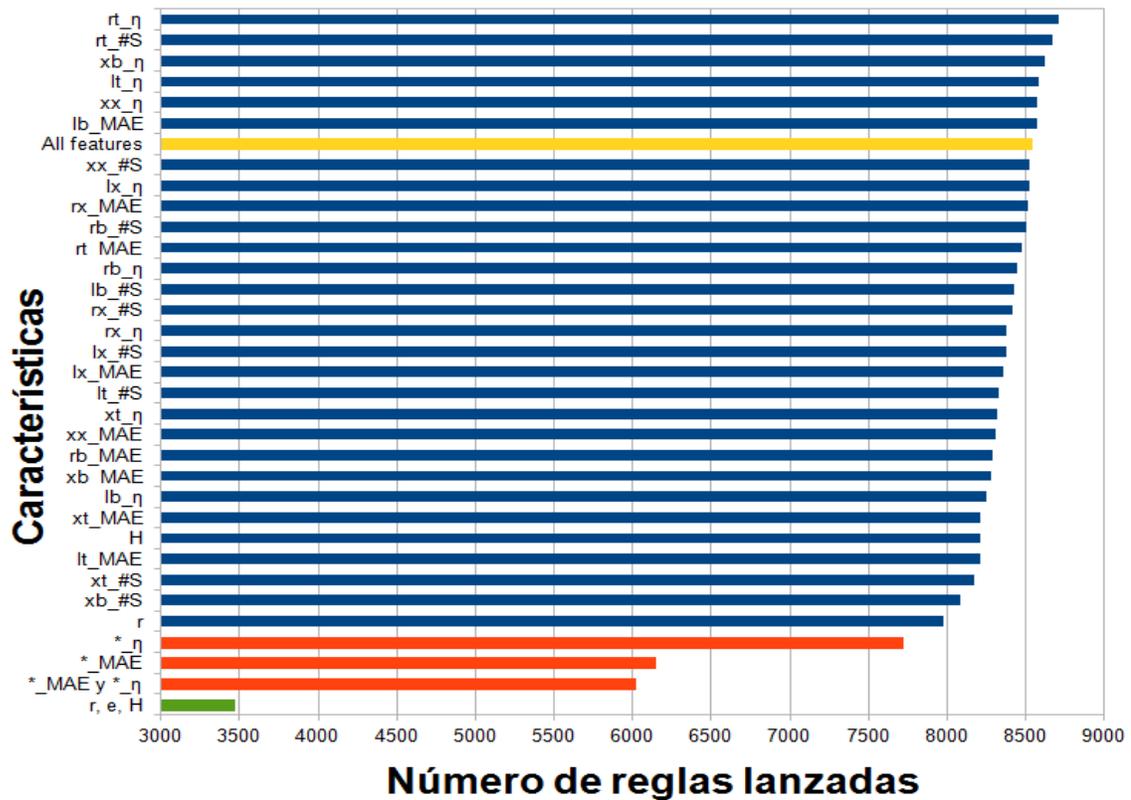


Figura 5.8: Número de reglas lanzadas para cada uno de los conjuntos de datos considerados.

Por otra parte, si nos centramos en la ganancia de la MAE y más específicamente en la MAE absoluta (columna $\%aG$ en la Tabla 5.18), encontramos en la Figura 5.9 ordenadas decrecientemente las ganancias correspondientes a cada uno de los conjuntos de datos. Si analizamos en detalles de esta medida sus valores en la Tabla 5.18, podemos intuir cierta relación de dependencia con el número de reglas lanzadas (Reglas) y la $\%G$. Con esta idea podemos considerar la ganancia absoluta, una medida global para evaluar la importancia de las características de la explicación. En este histograma, los valores indican que cuanto menor sea la ganancia, mayor será la pérdida cuando se elimine la variable del conjunto original, y por tanto, más importante es dicha variable.

El análisis del histograma de la Figura 5.9 nos muestra que dentro de las variables con menor ganancia y por tanto las más representativas e importantes son: $*_MAE$ y $*_η$; r ; H ; $rx_#s$, lb_MAE ; $xb_η$; $lt_#S$; $lx_#S$; xt_MAE . El histograma nos muestra que sin dudas nuestra propuesta de explicación es muy importante para intentar predecir el error de la predicción. Si eliminamos todas las características asociadas a la explicación

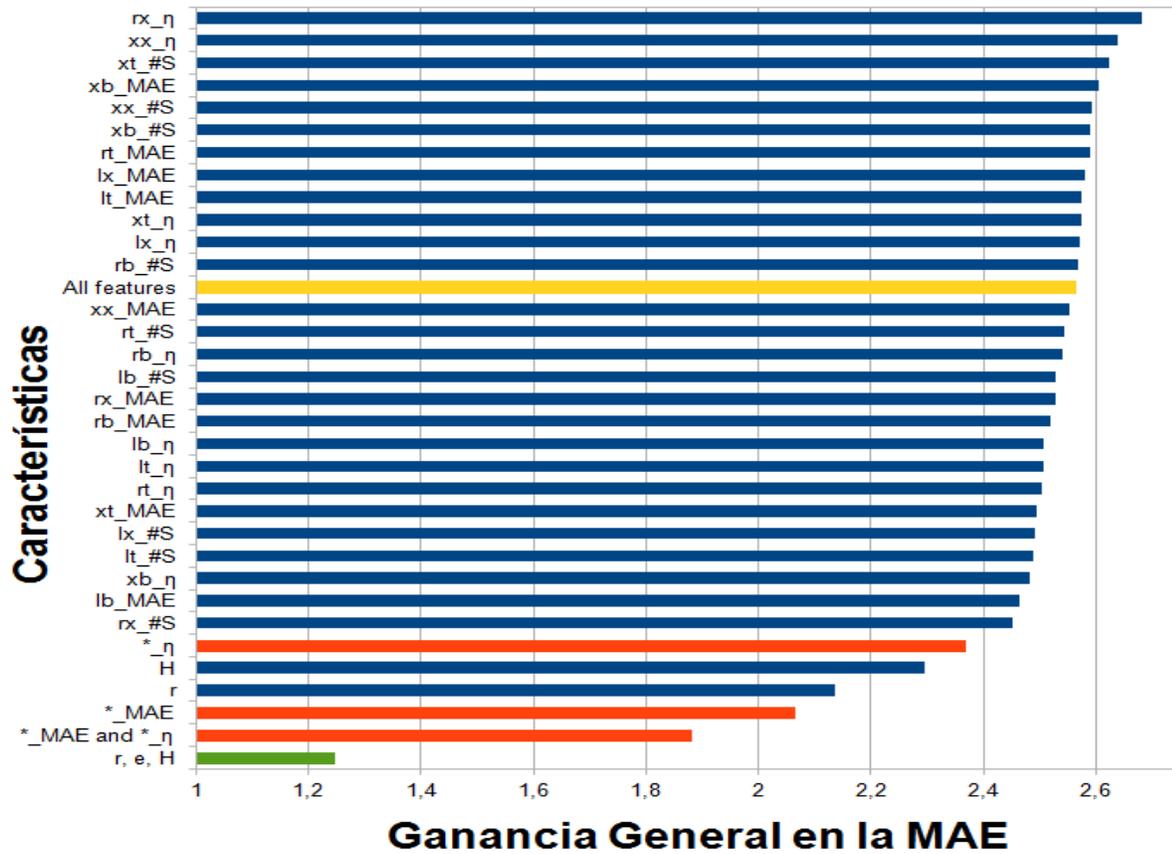


Figura 5.9: Ganancia absoluta de la MAE para cada conjunto de datos según las características consideradas.

(barra identificada con r, e, H), comprobamos como se reduce considerablemente la ganancia absoluta que se logra y mayor es la pérdida.

Si relacionamos los resultados tanto del orden obtenido con el número de reglas lanzadas, como de la ganancia absoluta de la MAE , dentro de los primeros atributos simples y por tanto más importantes, encontramos: r , H y $rx_{\#s}$. Mientras agrupados, los atributos relacionados con la MAE y η son los más importantes. Estos resultados nos permiten detectar atributos que son importantes al mismo tiempo tanto para lanzar reglas a los usuarios, como para que las reglas lanzadas sean efectivas.

Los resultados experimentales en esta sección nos confirman la importancia de la interfaz de explicación propuesta, y en especial de la gráfica bidimensional, donde mostramos el comportamiento del vecindario sobre las experiencias previas, ya que verificamos que la información que encontramos en ella, resulta ser muy relevantes para los propósitos de predicción.

5.5. Conclusiones

En el capítulo mostramos una propuesta para corregir el error en las predicciones, apoyándonos en la información que logramos obtener de las explicaciones. El proceso de aprendizaje del error, emplea una estrategia que permite inducir árboles de decisión para cada usuario. Un árbol generado contiene los atributos más relevantes de un usuario asociados a su comportamiento y la información de la explicación, lo que permite analizar las distintas ramas y hojas del árbol para determinar si es oportuno o no corregir la sugerencia inicial del SR.

Los resultados experimentales muestran que logramos corregir en un considerable porcentaje de usuarios (entre el 25 % y 35 %) las sugerencias iniciales del modelo de recomendación, lo que evidencia la validez de la propuesta. La experimentación muestra que los beneficios de emplear la explicación (especialmente, en términos de juzgar las recomendaciones) dependen altamente del usuario, que si para uno los resultados son buenos, no tiene porque serlo para todos. Al obtener mejoras solo en un grupo de usuarios, nos planteamos realizar futuros estudios para analizar la dependencia de los resultados, por ejemplo, por ítems, o tipos de ítems.

Otro importante resultado del capítulo, fueron las características que resultaron ser las más importantes dentro de todas las analizadas, donde además se corroboraron las conclusiones empíricas del estudio con usuarios reales sobre la información a la que se le prestaba mayor atención. Estos resultados servirán para futuros estudios en las tareas de explicación de las recomendaciones, al mostrar la información en las que debemos centrar especial atención.

Por último, se debe resaltar que aunque nuestro enfoque está diseñado para un SR basado en vecindario, este pudiera aplicarse a otras estrategias de recomendación, por ejemplo, para aprender la importancia de algunas de las características de un SR basados en contenido. En este sentido, este enfoque se puede utilizar para aprender de experiencias pasadas que tienen en cuenta los usuarios para tomar decisiones.

Parte II

Recomendación basada en contenido

En esta última parte abordamos otro tipo de SR, los basados en contenido, que junto a los de filtrado colaborativo son de los más utilizados. A diferencia de los sistemas analizados anteriormente, en estos recomendadores se establecen relaciones entre el contenido de los ítems.

En el primero de los capítulos diseñamos dos propuestas de modelos para predecir el interés noticioso de cada usuario. En la primera de las propuestas, a la que llamamos Dyadic, utilizamos el contenido de las noticias y los accesos realizados por los usuarios para predecir el interés hacía las noticias de un día determinado. Para la segunda de las propuestas, Triadic, además de la información utilizada por Dyadic, empleamos la clasificación dada para cada noticia.

Con los modelos propuestos, consideramos la tarea de obtener un *ranking* personalizado de noticias para cada usuario.

Finalmente realizamos una propuesta de explicación para este tipo de SR. Generalizando las ideas que logramos en el capítulo de explicación de la primera parte, diseñamos una interfaz para explicar la recomendación de noticias. Cada uno de los elementos de la propuesta se describe en detalles para facilitar su comprensión.

Capítulo 6

Desarrollo de modelos de recomendación basados en contenido

Las nuevas noticias a las que los usuarios se enfrentan al acceder a un periódico y su continuo crecimiento se convierte en la problemática que se trata en este capítulo. Un usuario al acceder a un sitio de noticias se encuentra con mucha información de la cual tiene que elegir cual leer en un tiempo determinado. Personalizar la forma de mostrar las noticias es la solución que se describe a lo largo del capítulo, donde se determinan cuáles son las noticias que interesan al usuario y el orden de lectura. La predicción anterior se sustenta en la información que se dispone de los accesos previos realizados por los usuarios, así como de la composición y clasificación de las noticias.

De forma general el objetivo en el que nos centramos es predecir cuáles son las noticias que le resultarán de interés a los usuarios que se conecten a un medio digital. Para lograr este objetivo se desarrollan los modelos Dyadic y Triadic, los cuales emplean la información disponible de las noticias y los accesos de los usuarios para estimar el interés de los usuarios por las nuevas noticias. Además se utilizan otros modelos descritos en el Capítulo 2 con la finalidad de validar las propuestas anteriores.

Se realizan varias comprobaciones con los diferentes modelos y se evidencia el buen desempeño de Dyadic y Triadic respecto a los restantes. En Triadic se emplea además de la información de los accesos a las noticias por parte de los usuarios (información empleada en Dyadic), la clasificación que se le da a dicha noticia, lo que reporta resultados superiores. Estos resultados reflejan la importancia que tiene para los usuarios el tener un patrón o forma de acceder a la información, por ejemplo, la organización que se da en los periódicos en temáticas de Deporte, Cultura, etc.

Los resultados alcanzados en las diferentes experimentaciones, siguiendo metodologías estándares nos permiten confirmar en general, i) la validez de las propuestas para la recomendación de un listado de noticias a los usuarios en un entorno digital, y ii) la calidad de las recomendaciones dado el orden ofrecido.

En el capítulo describimos inicialmente la problemática de la recomendación de noticias, destacando las principales respuestas académicas en el área. Seguidamente detallamos los modelos que proponemos y los datos empleados en la experimentación. Finalmente se realiza un análisis de los principales resultados experimentales que se obtienen.

6.1. La problemática de la recomendación en entornos noticiosos

Los usuarios al acceder a noticias en la Web, generalmente se encuentran con muchísima información de la que tienen que elegir cuál leer en el tiempo que disponen. La constante generación de noticias provoca que esta elección sea cada vez más compleja para quienes quieren estar bien informados. La posibilidad actual de disponer de dispositivos móviles, facilita que estos usuarios accedan cada vez con más frecuencia en busca de nuevas noticias, donde recomendar las noticias que deben ser leídas es una de las alternativas a este problema.

Cuando se menciona la temática de recomendación de noticias, como un problema clásico de clasificación dentro de los SR [1], se puede distinguir entre la aplicación de los sistemas de contenido y los de filtrado colaborativo. Los primeros proveen recomendaciones basados en el contenido de una noticia y la similitud de las noticias que el usuario consideró relevante en el pasado [85]. El filtrado colaborativo por su parte no considera el contenido de las noticias, pues emplea las opiniones del resto de los usuarios afines para generar las recomendaciones [107].

En los sistemas de filtrado colaborativo, se pueden distinguir dos diferentes situaciones. En la primera, el tiempo de vida útil de un artículo no expira, como en el caso de la recomendación de artículos científicos desde una biblioteca digital, tales como ResearchIndex [87]; en la segunda, el tiempo de vida de un artículo es corto, como en el caso de GroupLens (las publicaciones expiran después de una semana) [67, 95] y Google News (el tiempo de vida de las noticias puede ser de horas o un día) [32].

Nuestra investigación en este entorno, se enfoca en el último problema, dado el interés de los usuarios en acceder a las noticias que hayan sido publicadas recientemente. Esta situación es la más común en los usuarios que acceden a los diarios digitales. La principal diferencia entre los modelos propuestos en GroupLens y Google News son las formas que los usuarios expresan sus preferencias.

Primeramente, en GroupLens los usuarios evalúan las publicaciones explícitamente en una escala de 1 a 5, mientras en Google News se trata el intercambio del usuario con las noticias (clic) como una valoración positiva. Segundo, los modelos difieren en los algoritmos empleados para computar las recomendaciones. En el primero de los casos se sigue una propuesta basada en memoria, donde las recomendaciones para cada individuo son obtenidas al identificar la vecindad de usuarios similares y recomendar los artículos que este grupo

de usuarios encontró útiles. Esta filosofía fue la empleada para recomendar noticias en una versión digital de telegram.com, edición digital de noticias de Worcester, Massachusetts [28]. Sin embargo, en Google News se emplea una aproximación basada en modelos en la cual las recomendaciones se obtienen empleando una combinación de algoritmos de agrupamiento probabilísticos como minHashing y PLSI.

El problema con estos modelos es que un artículo debe ser evaluado muchas veces antes de poder ser recomendado, problema ya comentado y común en muchos SR (comienzo frío) [73]. En el campo de la recomendación de noticias este es un tema delicado, pues se necesita que desde el mismo momento que la noticia esté disponible, sea recomendada a los usuarios y no tener que esperar a que sea evaluada por otros usuarios [77]. Si este problema existiera en un sistema de recomendación noticioso, pudiera darse el caso de que una noticia perdiera vigencia mientras obtiene las suficientes valoraciones para ser recomendada.

De forma general, en el contexto que tratamos, el filtrado colaborativo por lo general suele ser combinado con las aproximaciones basadas en contenido. En aplicaciones que se tengan pocos usuarios, el tiempo en que las noticias logren ser valoradas por un gran número de usuarios puede ser considerable, por lo que aplicar un enfoque colaborativo simplemente, no es una opción adecuada.

Para las propuestas basadas en contenido, se puede distinguir entre los sistemas en busca de noticias relacionadas con el tema que un usuario está leyendo actualmente [119] y los que sólo tienen en cuenta el perfil del usuario (un modelo de los gustos generales de los usuarios o preferencias) para encontrar noticias relevantes. Bajo la primera alternativa se puede destacar el trabajo de Boger y Bosh [16], quienes vieron la recomendación de noticias como un tipo específico de tarea de recuperación de información (RI). Se emplean técnicas estándares de RI para medir la capacidad de un sistema de encontrar noticias relacionadas en una forma similar a como en la RI se trata de encontrar documentos relevantes a una consulta determinada. La misma idea fue empleada por Gabrilovich et al. [43] para un propósito contrario: ellos buscaban para nuevas historias, contextos donde los usuarios hubiesen accedido y resultarán interesantes.

Tsagkias y Blanco [119] proponen un modelo que utiliza los datos históricos de navegación de los usuarios para realizar predicciones de noticias en tiempo real. La idea es proponer nuevas noticias a los usuarios momentos después de haber leído una nueva noticia.

Para los modelos que emplean los perfiles de usuario, existen muchas alternativas que dependen de cómo se obtiene la expresión del interés del usuario y de la descripción del perfil. En algunas investigaciones se emplean los perfiles que fueron explícitamente declarados por los usuarios [12, 25, 28, 82, 88, 101], aunque en la mayoría de los casos los autores coinciden en que estas preferencias tienen que ser aprendidas implícitamente.

Kamba et al. [101] estimaron las preferencias de usuarios a partir del comportamiento que estos tuvieron en la navegación (desplazamientos, cambio de tamaño de ventanas, etc.); estas

acciones representan la evaluación que el usuario daría al artículo que estaba leyendo. Del mismo modo, Liang y Lai [74] consideraron que el interés del usuario dependía del tiempo invertido en leer una noticia. No obstante, la mayoría de los sistemas considera que cuando un usuario hace clic sobre una noticia para leerla, esto representa una valoración positiva para dicha noticia.

Un perfil de usuario como ya se valoró, describe el interés del usuario en el contenido de un artículo, y las propuestas que se suelen encontrar que aborden la temática varían desde considerar sólo un conjunto de categorías, particularmente la categoría en la que se clasifica una noticia [77], tales como deportes, internacional o salud, a considerar una subconjunto de palabras relevantes (por ejemplo, los términos relacionados con quién, qué, dónde, cuándo y por qué [74] o bien las 150 palabras más informativas [12]) [28, 82], otra opción es considerar las etiquetas [25] o simplemente usando todas las palabras (términos) en los artículos de entrada como las características, representado como un vector de palabras y sus correspondientes pesos [101].

Ahora bien, debe recordarse que a los sitios Web de noticias continuamente acceden personas ávidas de información, de las que se pueden distinguir aquellos usuarios frecuentes que presentan un comportamiento de visitas muy similar en cada una de las sesiones de navegación. Puede observarse además un grupo de personas cuyo comportamiento varía de sesión en sesión, o simplemente aquellos que realizan una visita ocasional, motivados por un evento deportivo, cultural o algún suceso que en ese instante de tiempo acapara la atención.

De manera general, todos los usuarios esperan encontrar aquellas noticias que le permitan estar satisfechos al concluir la visita, pero muchas veces la estructuración de la aplicación Web, o los resúmenes que pudieran aparecer de los artículos o simplemente el no percatarse de alguna noticia, pudieran ser las razones para que el estado de satisfacción deseado no sea alcanzado.

Al problema anterior pudieran ser aplicados diversos mecanismos con el fin de incrementar la satisfacción de los usuarios, algunos de ellos conciben el registrar al usuario para conocer sus intereses, por ejemplo mediante un perfil de usuario [44] como ya se ha mencionado previamente, ya sea explícitamente (por medio de un cuestionario) o implícitamente (analizando su patrón de navegación). Luego, ante cada visita y una vez que el usuario se identifique, se le muestran (recomiendan) aquellas nuevas noticias que pudieran resultarle interesantes. En [28] se sigue esta idea, donde los usuarios van indicando su opinión de los artículos en un diario online de noticias. Esta información entonces es utilizada por un recomendador híbrido.

Pero lograr en este tipo de sistemas que un usuario se logre registrar es complejo, pues para un alto porcentaje revelar algunos datos personales va en contra de la privacidad o es un gasto de tiempo innecesario, resultando más práctico acceder de forma anónima. Para atenuar esta situación, se emplean otras técnicas, las cuales intentan reconocer a los usuarios

por la dirección IP desde donde acceden, siendo un problema en este caso el que personas con preferencias variadas pudieran acceder desde la misma localización.

Se pudiera resumir que todas estas técnicas intentan reconocer las preferencias de los usuarios para poder realizarles recomendaciones de noticias acorde a sus intereses. En sentido general funcionan bien, pero el número de visitas que se va registrando, trae aparejado un crecimiento en el volumen de datos y es cuando se tiende a invertir un tiempo considerablemente mayor al que un usuario pudiera esperar para recibir una recomendación.

En lo adelante consideraremos la dirección IP como identificador de usuario, éste es un número asignado a cada dispositivo conectado a la red pública de Internet y serán éstas las direcciones desde donde accedan usuarios al periódico digital. Obviamente, este identificador es posible sólo cuando el IP es estático, ya que en los casos de direccionamiento dinámico, se vuelve más complejo atacar este tipo de problema.

6.2. Modelos propuestos

El principal objetivo de esta investigación es predecir el conjunto de noticias que a un usuario le interesaría leer cuando se conecte a un medio digital. Partimos para ello de la hipótesis de que los usuarios pueden ser agrupados en función de sus preferencias, y que éstas si no son declaradas explícitamente, pudieran obtenerse del comportamiento previo realizado por los usuarios.

La utilización de las técnicas de agrupamiento está enfocada en encontrar conjuntos de direcciones IP que muestren comportamientos de visitas muy similares en sus sesiones de navegación. Las distintas técnicas de *clustering* analizadas en el capítulo de introducción, se emplean en esta investigación para formar grupos de usuarios con intereses comunes que le faciliten a un usuario el acceso a las noticias. La idea es identificar el grupo al que un usuario pertenece, reduciendo con ello el conjunto de datos sobre el que los algoritmos de recomendación intentarían predecir las noticias que le pueden resultar de interés.

La principal razón de la utilización de este enfoque, como se podrá comprobar más adelante, está dada por el número elevado de usuarios que se tienen en el sistema, y con ello la dificultad en la etapa de búsqueda de vecinos en los algoritmos basados en memoria cada vez que un usuario comienza a interactuar con el portal de noticias.

Los algoritmos K-Medias y Aspect Model, descritos en el Capítulo 2, fueron utilizados en una investigación inicial [31]. En dicha propuesta se agruparon los usuarios de forma tal que los grupos formados contenían usuarios con comportamientos de lectura similares. En el primero de los algoritmos cada usuario pertenecía a un solo grupo, mientras con la segunda de las técnicas, la pertenencia de un usuario era a más de uno de los grupos que se formaban en un grado determinado.

La idea de dicha investigación fue encontrar aquellos IPs que pudieran estar interesados en una determinada categoría de noticia. Particularmente, dada una categoría de noticia, nuestro objetivo fue ordenar los IPs en orden decreciente según el interés que mostraban hacia una categoría. Este problema lo podemos encontrar comúnmente en muchas aplicaciones de comercio electrónico.

Los resultados obtenidos en general reflejaron el buen desempeño de las propuestas, siendo mejor en aquellas categorías de mayor acceso. El empleo de información colaborativa evidentemente ayudó a predecir el interés de los IPs, y por tanto de los usuarios.

Dado que esta investigación fue realizada empleando el conjunto de datos con los accesos de los usuarios a las categorías de noticias, y no el contenido de las noticias, decidimos realizar otros estudios para comprobar la utilidad de las noticias, así como su composición y clasificación.

Considerando los elementos anteriores, proponemos dos modelos, Dyadic y Triadic, que se basan en Aspect Model y sus extensiones analizadas en el capítulo introductorio. Se considera las valoraciones implícitas, es decir, se considera que cuando un usuario accede a una noticia está expresando un voto positivo.

Respecto al perfil del usuario, se tienen en cuenta tanto los términos de las noticias como la categoría a la que la misma pertenece. Las recomendaciones como se observará, se obtienen al considerar un propuesta probabilística que representa diferentes tipologías de usuarios.

6.2.1. Modelo Dyadic

La idea de Dyadic queda reflejada en la Figura 6.1. Este modelo cuenta de dos partes: usuarios y términos (palabras) de las noticias. Se asume que los usuarios son independientes de los términos. Una interpretación de este modelo, de acuerdo a Yoshii et al. [124] es que el usuario u estocásticamente selecciona un genero conceptual z de acuerdo a sus preferencias $P(z|u)$, y que z estocásticamente genera términos t de acuerdo con la probabilidad $P(t|z)$. Los géneros conceptuales no son suministrados por adelantado, sino que son automáticamente determinados según el modelo que provee la mejor explicación para las noticias accedidas.

Para entrenar Dyadic, seguimos el protocolo empleado por Hofmann [55] y que describimos en el Capítulo 2 en la Sección 2.3.1.2. En la Figura 6.2 se puede observar cómo los m usuarios que se tienen, van a tener un cierto grado de pertenencia a cada uno de los k grupos que se decidan formar, lo mismo ocurrirá con cada una de los n términos de las noticias.

Una vez se ha entrenado el modelo, es posible emplearlo para realizar el proceso de inferencia. Por ejemplo, podemos computar la probabilidad de que el usuario acceda a una noticia N_i , considerando todos los términos contenidos en la noticia:

$$P(N_i, U) = \sum_{t \in N_i} P(t, U) = \sum_{t \in N_i} \sum_{z \in Z} P(z)P(t|z)P(u|z). \quad (6.1)$$

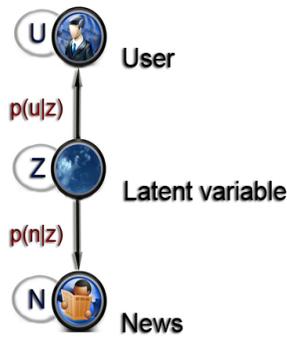


Figura 6.1: Representación gráfica del modelo Dyadic.

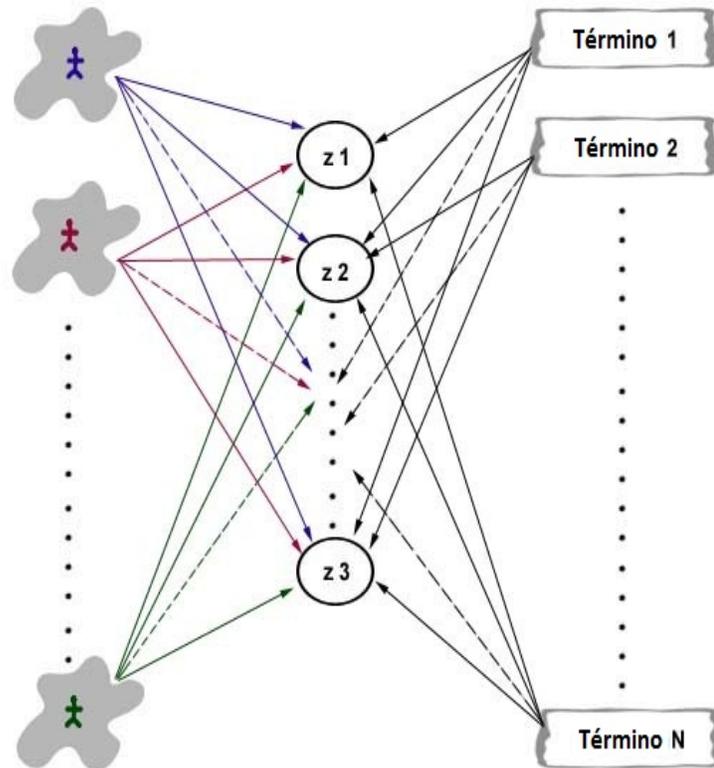


Figura 6.2: Representación del modelo para usuarios que acceden a noticias.

Evidentemente dado el enfoque planteado, es posible filtrar del total de noticias, el conjunto que sea más relevante a un usuario. Para ello bastaría con computar la Ecuación 6.1 para todas las noticias a recomendar y luego de ordenarlas seleccionar las “ k ” mejores. De igual forma sería posible determinar aquellos usuarios que más interesados estarían en una noticia. Para esto último bastaría con computar la Ecuación 6.1 para todos los usuarios que

se desearan analizar.

6.2.2. Modelo Triadic

En el caso de Triadic, encontramos en la Figura 6.3 una representación gráfica, que incluye además de los elementos considerados en Dyadic (usuarios y términos), las categorías en las que son clasificadas las noticias.

El empleo de un nuevo elemento en este modelo, es motivado por la intuición de lo útil que resulta para los usuarios el encontrar la información clasificada en un medio de información. Al emplear las categorías asociadas a la información que tenemos de los accesos realizados por los usuarios a las noticias, estamos considerando que el proceso de predicción será más eficiente.

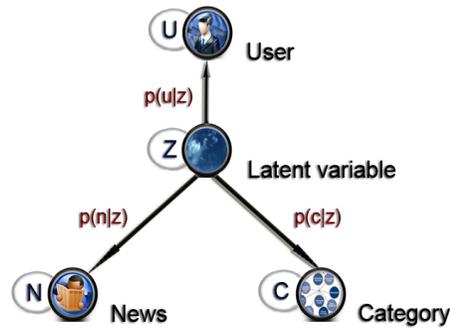


Figura 6.3: Representación gráfica del modelo Triadic.

En este caso, para entrenar el modelo Triadic, consideramos una observación como una tripleta (U, N, C) correspondiente al acceso realizado por un usuario U a una noticia N (y el conjunto de términos) de la categoría C . Conceptualmente, podemos distinguir diferentes tipologías de usuarios $z \in Z$, las cuales a su vez generan tanto las noticias como las categorías a la que pertenecen las noticias. Asumiendo que los usuarios, noticias y categorías, son independientes dado las tipologías, estimamos las diferentes distribuciones de probabilidades.

Sea $n(U, t, C)$ el número de veces que el usuario U lee el término t en las noticias de la categoría C . Disponiendo de los datos de esta forma, el logaritmo (\log) de la probabilidad, L , es computado según la ecuación:

$$L = \sum_{U,t,C} n(U, t, C) \log P(U, t, C). \quad (6.2)$$

En el modelo las probabilidades de distribución son estimadas empleando el algoritmo EM (Ecuaciones 6.3 y 6.4).

Paso E:

$$P(z|U, t, C) = \frac{P(z)P(u|z)P(t|z)P(c|z)}{\sum_{z'} P(z')P(u|z')P(t|z')P(c|z')}. \quad (6.3)$$

Paso M:

$$\begin{aligned} P(u|z) &\propto \sum_{n,c} n(U, t, C)P(z|U, t, C), \\ P(t|z) &\propto \sum_{u,c} n(U, t, C)P(z|U, t, C), \\ P(c|z) &\propto \sum_{u,n} n(U, t, C)P(z|U, t, C), \\ P(z) &\propto \sum_{u,n,c} n(U, t, C)P(z|U, t, C). \end{aligned} \quad (6.4)$$

Una vez que el modelo ha sido entrenado, la probabilidad conjunta de distribución para que el usuario U acceda a la noticia N_i perteneciente a la categoría C , puede ser fácilmente computada al considerar todos los términos en la noticia y marginar la clase latente Z :

$$P(N_i, U, C) = \sum_{t \in N_i} P(t, U, C) = \sum_{t \in N_i} \sum_{z \in Z} P(z)P(t|z)P(u|z)P(C|z). \quad (6.5)$$

En la literatura se pueden encontrar enfoques cercanos a la propuesta que se muestra [77, 87]. Popescul et al. [87] presentan una propuesta que recomienda artículos combinando enfoques colaborativos y basados en contenidos. Una de las principales diferencias con dicha propuesta es el tiempo de vida de un artículo, es por ello que es posible considerar el enfoque colaborativo. Ellos asumen una variable de clase latente (representando los tópicos de interés de los usuarios) que relaciona usuarios, documentos y palabras o términos. La información colaborativa es obtenida al medir el número de veces que un usuario con los mismos intereses ha accedido a un determinado documento. Para aplicar el enfoque basado en contenido, obtienen esta información computando el número de veces que una determinada palabra aparece en un documento de un determinado tópico. Con toda esta información construyen un modelo probabilístico.

Aunque nuestra propuesta y el modelo de Popescul et al. [87] emplean el mismo modelo base (*three-way aspect model*), la codificación de la información es diferente: nuestro modelo está basado en contenido, mientras el de Popescul et al. [87] es una propuesta híbrida. Además, para determinar cómo de probable es que un usuario determinado acceda a una noticia, también es diferente la información empleada en el proceso de recomendación. Nosotros

recomendamos un artículo al considerar el contenido, N , y su categoría, C . Popescul et al. [87] recomiendan un artículo si los usuarios con tipología similar han leído el artículo en el pasado.

Liu et al. [77] emplean una propuesta híbrida para personalizar la recomendación de noticias en Google News. Se enfocan en una propuesta basada en contenido (cercana a nuestra propuesta) donde analizan la distribución de accesos por categorías (argumentan que esto representa la distribución de interés del usuario). Nótese que ellos no emplean el contenido de las noticias (términos), sólo la noticia como un todo y la categoría en la que ha sido clasificada. El objetivo es predecir el interés de un usuario en una categoría determinada c_i (la probabilidad de que el usuario accede a un artículo de la categoría c_i). Para determinar esta probabilidad emplean tanto la distribución de accesos realizada por el usuario como la información de la tendencia actual de las noticias (obtenida de la distribución de accesos de todos los usuarios en un determinado periodo de tiempo, por ejemplo, la última hora). Finalmente, esta información se combina con información colaborativa obtenida de los usuarios similares. Se recomienda un artículo cuando usuarios similares han accedido al artículo anteriormente.

Nuestra propuesta es diferente a la de Google News dado que esta última no considera el contenido de las noticias para la recomendación. En la predicción del interés en una categoría, no se emplea información de usuarios similares o relacionados, mientras que nuestro enfoque tiene en cuenta la tipología del usuario.

En resumen, Dyadic es capaz de encontrar las relaciones entre usuarios y noticias, donde usuarios con accesos a noticias muy similares por su contenido son utilizados para recomendarle al usuario activo aquellas noticias que debe de leer. Por su parte Triadic es más específico, al considerar como dato adicional la categoría, este elemento es incorporado al modelo permitiendo que las recomendaciones que reciba el usuario estén más adaptadas a sus intereses de noticias. Así si un usuario accede a noticias de Cultura y Fútbol, sus colaboradores serán aquellos muy similares en hábitos de lectura, donde además las categorías de noticias se convierten en una condición a cumplir.

6.3. Experimentación

Una vez analizados los modelos propuestos, antes de describir la experimentación y los resultados obtenidos, creemos oportuno destacar las principales características del conjunto de datos que se empleó.

6.3.1. Colección de prueba

Los datos que se emplearon en esta parte de la investigación, tienen su origen en datos reales procedentes del portal Web de noticias del diario ¡Ahora! de Holguín, Cuba ¹; diario digital provincial que muestra noticias diarias principalmente del acontecer de la provincia de Holguín, así como del resto del país y del mundo.

Originalmente se tuvo la información de un periodo de tiempo de un año, comenzando en el mes de marzo del 2008, etapa que incluía un total de 2.262.163 peticiones realizadas de una fuente total de 3.153 páginas. Los accesos se efectuaron desde 137.314 direcciones IP diferentes, para un total de 540.503 sesiones de usuarios. De forma general, no se concibe colaboración explícita por parte de los usuarios, pues aunque se encuentra implementado el registro y posterior identificación de éstos en el portal, son escasos los que hacen uso de dicho registro, dado que generalmente los usuarios Web se muestran esquivos de las aplicaciones que solicitan información personal.

Los datos que conforman el conjunto de datos resultantes son extraídos de las distintas tablas donde se almacena toda la información referente a los accesos realizados por los usuarios a las distintas páginas del diario digital. Dentro de las tablas que se manejan se encuentran, las que almacenan las direcciones IP desde donde se realizan los accesos, las páginas de contenido (noticias), las categorías de noticias y los accesos realizados en un instante de tiempo por los usuarios a las noticias que se muestran en el portal.

El análisis y procesamiento de toda esta información permite identificar las sesiones de usuarios, las que reflejan el acceso realizado en un intervalo de tiempo por los usuarios, conociéndose en ellas la dirección IP desde donde se accede y las noticias accedidas. De cada noticia se conoce entre otros datos la categoría en la que ha sido clasificada o a la que pertenece.

Por ejemplo:

Sesión_0 : $ip_1 + (N_A, Cat_10), (N_B, Cat_10), (N_C, Cat_12), (N_D, Cat_18)....$

Sesión_1 : $ip_2 + (N_B, Cat_10), (N_H, Cat_50), (N_E, Cat_36)....$

Sesión_2 : $ip_1 + (N_A, Cat_10), (N_H, Cat_50), (N_M, Cat_18)....$

Sesión_3 : $ip_2 + (N_A, Cat_10), (N_G, Cat_36), (N_I, Cat_36)....$

...

Sesión_x : $ip_m + (N_A, Cat_10), (N_F, Cat_12), (N_D, Cat_18)....$

donde se indica que se accedió desde el ip_1 a la noticia A que pertenece a la categoría 10, noticia B que pertenece a la categoría 10, noticia C que pertenece a la categoría 12, noticia D que pertenece a la categoría 18, etc. Nótese que se obtienen por IP, cada uno de los accesos realizados en diferentes intervalos de tiempo, es decir, pudieran obtenerse una o varias sesiones por cada IP.

¹(<http://www.ahora.cu/>)

Con estos datos se forma entonces una matriz como se muestra en la Tabla 6.1, donde las filas serán los números $IP = ip_1, ip_2, \dots, ip_m$ desde donde accedieron los usuarios y las columnas las categorías $C = Cat_1, Cat_2, \dots, Cat_n$ a las que pertenecen las noticias, la intersección de una fila y columna ha de representar el número total de noticias accedidas de una categoría correspondiente por un usuario desde una dirección IP.

Tabla 6.1: Ejemplo de una matriz de accesos realizados por los usuarios a las categorías de noticias

	...	Cat_10	Cat_12	Cat_18	Cat_36	Cat_50	...	Cat_n
ip_1		12	4	19				
ip_2		1			12	7		
...								
ip_m		20	6	36				19

Los datos obtenidos fueron analizados como primera fase en el proceso, lo cual permitió eliminar peticiones de gestión de información interna u otras que no representaban accesos a las categorías definidas. En este momento todos los accesos de administradores y desarrolladores se eliminan del conjunto de datos. También se eliminan aquellas direcciones IP que tienen solo un acceso al portal de noticias, dado que no reportan suficiente información.

Las categorías de noticias son analizadas, eliminándose las que luego de los pasos previos, no tienen registrados accesos de usuarios en el periodo de tiempo señalado.

Al concluir esta fase se obtuvo el conjunto de datos final sobre el que se hace el resto de las etapas del proceso, cuyo total de diferentes direcciones IP resultó ser de 102.631, las que accedieron a un número de 2.036.449 noticias de 49 categorías. En la Tabla 6.2 encontramos todas las categorías y sus respectivos identificadores.

En las Figuras 6.4 y 6.5 se resumen algunas propiedades básicas del conjunto de datos resultantes luego de las operaciones de filtrado que se realizaron, descritas anteriormente.

La Figura 6.4 muestra por categorías, los accesos totales realizados a éstas, gráfico que permite comprobar el poco número de categorías que son muy visitadas por los usuarios. Resaltan dentro de las categorías a las que los usuarios accedieron 100.000 o más veces las siguientes:

- Holguín
- Nacional
- Internacional
- Cultura
- Deporte

Tabla 6.2: Categorías y sus identificadores (Id).

Id	Categoría	Id	Categoría
1	Holguín	26	VI Festival Internacional de Cine Pobre
2	Nacional	27	Desfile Primero de Mayo
3	Internacional	28	Romerías de Mayo 2008
4	Cultura	29	Los Cinco
5	Deporte	30	Ambito
6	Salud	31	Serranía
7	Especiales	32	Asamblea Provincial UPEC
8	Opinión	33	José Martí
9	Semana de la Cultura Holguinera	34	Terroristas-Mercenarios-EE.UU.
10	Feria del Libro	35	VIII Congreso UPEC
11	Romerías de Mayo	36	Asamblea Nacional (VII Legislatura)
12	Festival de Cine Pobre	37	Aniversario 32 de Gesta del Moncada
13	Fiesta de la Cultura Iberoamericana	38	Cuba en Beijing 2008
14	Feria Comercial Expo Holguín 2007	39	Datos Beijing 2008
15	Esquina Joven	40	Huracán Ike
16	Detrás de la Palabra	41	Bloqueo de EE.UU. Contra Cuba
17	Página 8	42	IV Coloquio Internacional en Holguín
18	De lo Humano	43	Aniversario 27 Revolución Cubana
19	Actualidad	44	Asamblea Nacional del Poder Popular
20	Reflexiones de Fidel	45	Semana de la Cultura Holguinera 2008
21	Hemeroteca	46	Desfile Primero de Mayo 2009
22	Homeopatía	47	Romerías de Mayo 2009
23	Oye, amiga	48	Día de la Infancia
24	Noticias	49	Portada
25	VII Congreso UNEAC		

- Huracán Ike

En el gráfico de la Figura 6.5 se puede observar el número de direcciones IP que visitan noticias de las distintas categorías, igual que para los totales de accesos, se tienen un alto número de categorías de noticias que reciben poca atención de los usuarios.

Al considerar el contenido de las noticias y de forma similar a como se creó el conjunto de datos anterior, obtuvimos un conjunto de datos que reflejaba por un usuario los términos que mejor representaban sus intereses noticiosos. Partiendo de que se tiene los datos de la interacción de los usuarios agrupados en sesiones de navegación, \mathcal{S} , donde cada sesión representa los accesos realizados por un usuario en un periodo de tiempo:

$$[s_id, u_id, ts, \{(N_i, C_{N_i}), i = 1, \dots, k\}],$$

donde s_id y u_id representan la sesión y el ID de usuario. Mientras ts es el momento de tiempo y $\{(N_i, C_{N_i}), i = 1, \dots, k\}$ es el listado de accesos de noticias realizados a la

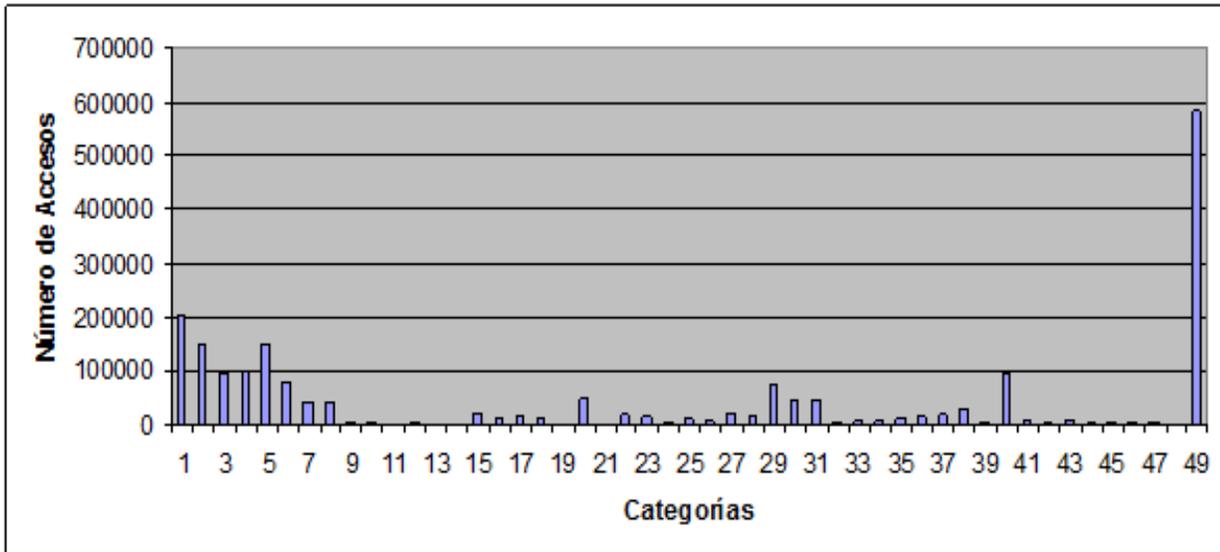


Figura 6.4: Número de accesos por categorías

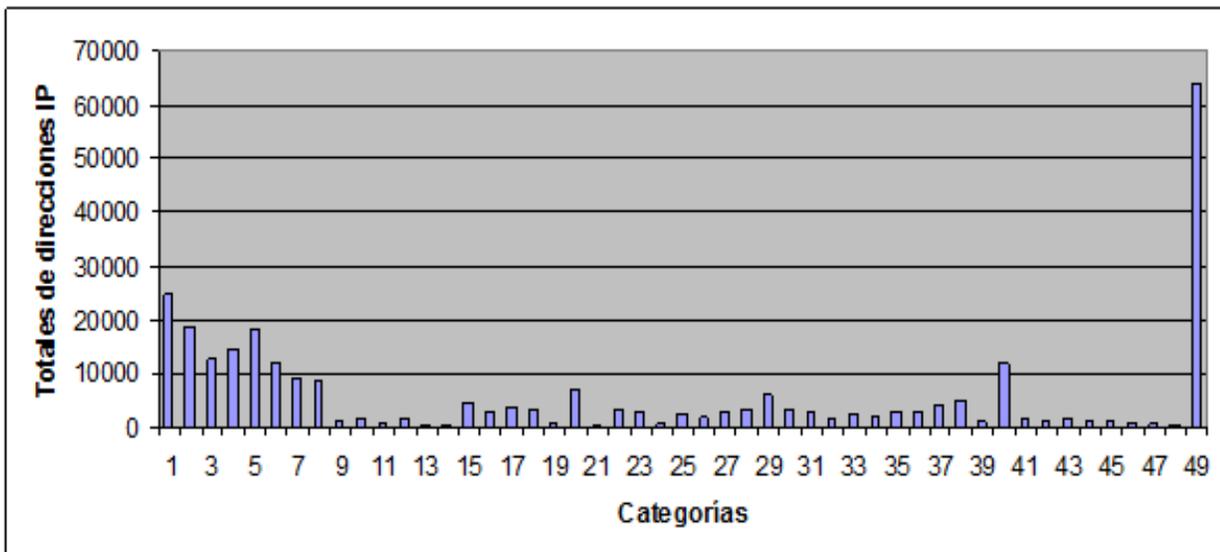


Figura 6.5: Totales de direcciones IP que acceden por categorías

categoría correspondiente.

Se emplea un segundo conjunto de datos para el contenido de las noticias \mathcal{C} , representado como una bolsa de términos. A dicho conjunto de datos se le realiza un preprocesado siguiendo los pasos del procesamiento documental clásico, el primer paso fue realizar el análisis léxico (*tokenizing* en inglés) de las noticias, obteniendo con ello las palabras o términos que la conforman.

Luego se realizó el proceso de selección de términos de las noticias, donde se eliminan

palabras vacías de significado que no nos sirven en el proceso, como pueden ser artículos, preposiciones, conjunciones, etc. (Proceso conocido como eliminación de palabras vacías *stop words* en inglés).

Posterior a este proceso se redujo algo más el vocabulario existente al disponer de un número considerable de términos, lo que requeriría una elevada potencia de computo para los modelos empleados. Para ello se pasó a realizar la segmentación, con el fin de obtener la raíz léxica de las palabras (*stemming* en inglés). Por último y analizando todas las noticias accedidas por el usuario se pasó a la fase de ponderación de términos, donde a cada uno de los términos que finalmente se encontraban en el vocabulario disponible, se le dio un peso en correspondencia a los pasos anteriormente descritos. El peso que se dio a cada término dependía del número de veces que aparecía en las noticias accedidas por el usuario.

Después de realizar todas las tareas descritas, se obtuvieron un total de 8.605 términos, por lo que cada artículo N_i es representado como un vector de tamaño igual a 8.605.

6.3.2. Metodología de experimentación

Un elemento importante en el empleo de las técnicas de *clustering* es el referente a la validación y verificación de los resultados obtenidos. Como característica común a la mayoría de estos algoritmos, se supone conocido el número de clusters; es decir, que los algoritmos buscan optimizar una determinada estructura de *clusters* para un número determinado de ellos. Dar por conocido el número de *clusters* puede ser algo razonable en ciertas ocasiones, pero no tiene una clara justificación si se está realizando el agrupamiento de un conjunto de datos cuyas propiedades se desconocen. Por tanto, un problema habitual en este tipo de análisis es, simplemente, decidir el número apropiado de grupos.

Cuando se utilizan técnicas de *clustering* basadas en la optimización de una determinada función, un método común para encontrar el número idóneo de grupos es llevar a cabo el agrupamiento aumentando o reduciendo el número de grupos (m), observando cómo la función cambia con dicho número. Generalmente se comprueba cómo el agrupamiento obtenido se comporta ante determinados comportamientos. En esta sección se comprueba el comportamiento de Dyadic y Triadic al variar el número de grupos.

En la experimentación que se describe en la sección, se consideran los siguientes objetivos y parámetros de configuración:

- **Objetivo de la experimentación:** El objetivo de esta experimentación es medir la capacidad del sistema de predecir los intereses de un usuario en las noticias publicadas en un periódico digital. En particular, tratamos de predecir un conjunto de noticias que pueden ser visitadas por un usuario en un día determinado y el orden de acceso de las mismas.

Las predicciones que se le hagan a los usuarios pueden ser empleadas para personalizar la aplicación a la que este accede o para enviar digitalmente al usuario resúmenes de las noticias más relevantes. Las predicciones se realizarán antes de que cualquier usuario acceda a las noticias, por lo que se debe resaltar que se desconoce en todo momento si la noticia se ha considerado relevante o no por otros usuarios.

La idea es encontrar un *ranking* personalizado, $\mathcal{R} = \{R_1, \dots, R_d\}$, del conjunto de noticias publicadas en un día determinado. El orden de las noticias representará la relevancia que tendrá cada noticia para el usuario.

- **Conjuntos de entrenamiento y prueba:** Para evaluar las diferentes alternativas, se emplea una metodología común en muchos problemas de minería de datos, donde se divide el conjunto de datos en conjuntos de entrenamiento, validación y prueba.

Se debe resaltar que nuestra metodología experimental es diferente a otros estudios de la literatura en los que se intenta encontrar noticias similares a las leídas por el usuario [16], o bien se predice un voto para la noticia [101]. Otros estudios sobre predicción de interés de los usuarios en noticias no leídas se han realizado [12, 43, 74, 77].

En particular, en esta experimentación se sigue la siguiente metodología para poder comparar el desempeño de los diferentes modelos bajo las mismas condiciones experimentales.

Los diferentes conjuntos de datos son obtenidos de forma aleatoria, seleccionando primero un día determinado, luego se extraen todas las noticias que han sido publicadas ese día y los accesos realizados por los usuarios a dichas noticias. La distribución de los conjuntos de entrenamiento, validación y prueba es de 70 %, 10 % y 20 % respectivamente del total de días analizados.

Siguiendo esta distribución podemos notar que todas las noticias y accesos de un día determinado se encontrarán en un solo conjunto de datos, ya sea de entrenamiento, validación o prueba. De esta forma se asegura que en el momento de realizar la predicción para un día en particular, ninguno de los usuarios ha leído la noticia.

Con el objetivo de reducir la variabilidad de los resultados, se ha diseñado una validación cruzada (*five fold cross validation*), obteniendo 5 particiones de los diferentes conjuntos de datos. Los resultados que se presentarán muestran los valores promedios obtenidos en las ejecuciones de las particiones.

- **Medidas de evaluación:** Dado el conjunto $\mathcal{N} = \{N_1, \dots, N_d\}$ de d noticias publicadas en un día determinado y un usuario dado U , nuestro objetivo es encontrar

un *ranking* $\mathcal{R} = \{R_1, \dots, R_d\}$ de estas noticias de acuerdo a la relevancia predicha para el U , donde $R_i \prec R_j$ significa que la noticia R_i es más relevante que R_j .

Se pueden encontrar muchas métricas comúnmente usadas para juzgar el *ranking* propuesto por un modelo M con respecto al orden de las noticias que han sido accedidas por el usuario, $Acc(U)$ [12, 32, 54, 69]. En esta experimentación se considerarán dos que fueron descritas en términos generales en la introducción:

- **Métrica basada en precisión:** En el contexto que se analiza se evaluará el porcentaje de noticias para las cuales la predicción obtenida es correcta en un conjunto ordenado de predicciones y por tanto, una alternativa es fijar una posición y evaluar esta métrica solo para las k predicciones superiores. Enfocándonos en la precisión, obtenemos la métrica de precisión a k (P@k).

Dado que se considera un grupo de usuarios, los resultados que se muestran representan la generalización de esta métrica al computar el promedio obtenido con los usuarios.

En el conjunto de datos se encuentra una gran variabilidad en el número de noticias accedidas por los usuarios, podemos encontrar usuarios que han accedido en un día a tres noticias, mientras otro accedió a ocho noticias. Para considerar esta variabilidad se ha considerado una métrica especial, de forma que se evalúe la predicción para cada usuario empleando el número de noticias accedidas por el usuario en el día que se analiza, $ac_u = |Acc(U)|$. De esta forma computamos $P@ac_u$ para todos los usuarios que accedieron ese día. Resulta interesante notar que al considerar lo anterior, estamos considerando similares la precisión y el recall, lo cual parece ser un apropiado criterio para medir el desempeño del sistema. Similar a como se mencionó anteriormente, se muestra el computo del promedio para los diferentes usuarios.

- **Ganancia acumulada de descuento normalizada (nDCG):** Esta métrica tiene en cuenta las posiciones en el *ranking* de las noticias accedidas por el usuario U , $Acc(U)$. Con la métrica penalizamos cuando una noticia relevante aparece en posiciones inferiores en el *ranking* \mathcal{R} .

Dado que el conjunto de noticias relevantes depende del número de veces que el usuario accedió durante el día que se analiza, el valor DCG no es comparable entre usuarios. Para facilitar la comparación como mencionamos en la introducción al analizar la métrica, se utiliza la medida nDCG. Como en los casos anteriores, se muestran los valores promedios para los diferentes usuarios.

Para ilustrar el desempeño de los modelos Dyadic y Triadic, se considerarán otros tres diferentes modelos de referencia:

- **Predicción personalizada basada en el coseno** (CosPP): En este modelo se realizan las recomendaciones considerando las noticias, N_1, \dots, N_d , en un orden de acuerdo a la similaridad de cada una de éstas respecto al perfil del usuario, empleando el coseno como medida de similitud, $\cos(N_i, u_j)$, según fue descrita en la sección de sistemas de recomendación basados en contenido del capítulo introductorio (Sección 2.3).
- **Predicción global basada en coseno** (CosGP): Este modelo no considera la información individual para realizar las predicciones. En particular, se considera que todas las noticias publicadas pueden ser representadas según la línea editorial adoptada por la publicación. De forma similar a como ha sido creado el perfil del usuario, es posible crear un perfil que responda a los intereses editoriales, mp. Este perfil es un simple vector que almacena la frecuencia de cada término durante el periodo que se analice. Entonces, todas las noticias publicadas en una día determinado, N_1, \dots, N_d , son ordenadas de acuerdo a la similitud con la línea editorial, empleando $\cos(N_i, mp)$.
- **Predicción global a posteriori** (PosGP): Este modelo ordena las noticias N_1, \dots, N_d de acuerdo al número global de accesos que tengan las mismas en un día de publicación de la editorial. Empleando este criterio, las noticias más populares (más visitadas) serán ubicadas en las primeras posiciones del *ranking* de noticias que se recomienda. Asumiendo este criterio el modelo simula el comportamiento real de los usuarios en un día determinado, por lo que representa una aproximación a posteriori. Debemos notar que no se emplea esta información cuando se realizan las predicciones en el resto de los modelos presentados, por lo tanto, el desempeño de este modelo puede ser considerado como un modelo ideal superior, al cual es deseable alcanzarlo en el proceso de predicción.

6.3.3. Resultados y análisis

Nosotros además consideramos emplear un peso proporcional a la frecuencia del término i -ésimo en la noticia j -ésima, e inversamente proporcional al número de noticias de la colección completa en las que aparece ese término (tf-idf), para medir la calidad de los diferentes términos en los modelos basados en la medida coseno, pero los resultados fueron mucho peores; por ejemplo, los valores $P@ac$ estuvieron en el rango de 0,12 a 0,15 (Tabla 6.3)².

Una vez construido el modelo, éste puede ser empleado para dos diferentes propósitos. Primeramente, por un lado, se puede predecir como un usuario puede interactuar con el sistema en un día determinado. En otras palabras, se pueden predecir aquellas nuevas noticias

²La idea era darle mayor importancia según crecía la frecuencia del término en la noticia y restarle importancia con respecto al número de noticias que la contenían.

Tabla 6.3: Predicción de un *ranking* global para todas las noticias publicadas en un día determinado, \mathcal{N} .

Models	P@1	P@3	P@ac	nDCG
BL-CosPP	0,354107118	0,32766772	0,342964774	0,11146047
BL-PostGP	0,563781265	0,465675828	0,548517887	0,20203343
BL-CosGP	0,308926672	0,296070125	0,302382166	0,08801274
Dyadic-Z2	0,356900929	0,345799828	0,358405485	0,09543524
Dyadic-Z3	0,358127255	0,346854619	0,36035569	0,09702731
Dyadic-Z5	0,360238749	0,34739972	0,362236544	0,09878843
Dyadic-Z10	0,365546683	0,348772129	0,365849497	0,10139783
Dyadic-Z15	0,365171716	0,34921851	0,365679994	0,10112429
Dyadic-Z20	0,365310024	0,348590786	0,366231325	0,10159453
Triadic-Z2	0,357325072	0,34388983	0,356859076	0,09589319
Triadic-Z3	0,363379866	0,345059435	0,361253063	0,09925867
Triadic-Z5	0,363007973	0,347629349	0,364388794	0,10139783
Triadic-Z10	0,363164721	0,346413603	0,363036714	0,10074932
Triadic-Z15	0,378458456	0,352206906	0,376225587	0,11088572
Triadic-Z20	0,38033329	0,351459003	0,37608993	0,11063677

publicadas que serán accedidas por el usuario. Esto está relacionado a la predicción de un *ranking* global \mathcal{R} para todas las noticias en \mathcal{N} . Por otro lado, podemos seleccionar las primeras noticias que le serán mostradas al usuario por cada categoría. En este segundo caso, nos enfocamos en seleccionar una categoría, C_k , y posteriormente predecir un *ranking* con todas las noticias publicadas en dicha categoría, $\mathcal{R}(C_k)$. En lo adelante mostraremos el desempeño de los modelos propuestos para ambos propósitos.

6.3.3.1. Predicción de *ranking* global

Los tres modelos adicionales descritos anteriormente fueron ejecutados empleando los conjuntos de datos de entrenamiento descritos. Los resultados obtenidos quedan reflejados en la Tabla 6.3, donde se muestran las medidas P@1, P@3, P@ac y nDCG (se destacan en negrita los mejores resultados). En la Tabla 6.3 se evidencia que es mejor emplear una aproximación personalizada (CosPP supera CosGP) en todas las medidas que se analizan, por lo que se ha decidido en lo adelante sólo mostrar los resultados obtenidos con el modelo CosPP como referente. Además, como esperábamos, el modelo donde se obtienen mejores resultados es el PostGP, el cual utiliza información real de cómo los usuarios han accedido a las \mathcal{N} noticias cuando se computa el *ranking*. Desde que esta información no está disponible para el resto de los modelos, estos resultados pueden ser considerados como los ideales a alcanzar para nuestras expectativas.

Ahora nos enfocaremos en los resultados obtenidos empleando los modelos Dyadic y Triadic. Para un usuario determinado U , el modelo Dyadic ordena las noticias N_1, \dots, N_d en orden decreciente según la probabilidad a posteriori $P(N_i, U)$ computada empleando la

Ecuación 6.1. Similarmente, el modelo Triadic emplea la información de las categorías de noticias C y por lo tanto, las noticias se ordenan en orden decreciente de $P(N_i, U, C)$, empleando la Ecuación 6.5.

La Tabla 6.3 muestra la precisión de los diferentes modelos cuando la variable latente Z toma los valores de 2, 3, 5, 10, 15 y 20. Al analizar los valores obtenidos, podemos concluir que todos los modelos que consideran una variable latente, se comportan mejor que los modelos basados en el computo de la medida coseno cuando se considera la precisión. Al analizar el valor de la clase latente Z , los mejores resultados son obtenidos con los valores de 15 y 20 tanto en el modelo Dyadic como en Triadic.

Finalmente, podemos observar como al incluir la categoría en el proceso de predicción, los resultados alcanzados resultan superiores y por tanto beneficioso el proceso de recomendación. Los resultados obtenidos con el modelo Triadic consistentemente tienen un rendimiento superior que los obtenidos con el modelo Dyadic bajo las mismas condiciones.

Estos resultados resultan peores que los esperados, pero como veremos en lo adelante, cuando nos enfocamos en una categoría en específico, la precisión mejora considerablemente.

6.3.3.2. Predicción de *ranking* basado en las categorías de noticias

En lo adelante se mostrarán las experimentaciones enfocadas en el segundo de los objetivos marcados, determinar las noticias más relevantes de una categoría de noticias para un usuario determinado. Para ello se seguirán dos enfoques: el primero siguiendo una propuesta general, donde se mezclan las noticias de todas las categorías, y el segundo, donde se trabaja con cada modelo de forma independiente.

- **Propuesta General:** Esta propuesta consiste en obtener un *ranking* global \mathcal{R} para todas las noticias como se han obtenido hasta ahora. Luego de obtener dicho *ranking* se filtran todas las noticias de una categoría C_k manteniendo la posición relativa en \mathcal{R} , obteniendo así un *ranking* para dicha categoría, $\mathcal{R}(C_k)$. Se debe resaltar que en esta propuesta se tiene en cuenta información del resto de categorías cuando se genera el modelo de recomendación. Por ejemplo, si un término es muy relevante para un usuario dada las noticias que este ha visitado, digamos “fútbol”, entonces la aparición de este término en noticias de otra categoría, por ejemplo, Cultura, hará que dichas noticias sean relevantes para el usuario, incluso si el usuario no ha mostrado preferencia por dicha categoría.

Para la experimentación empleamos el mismo conjunto de datos empleado para la predicción del *ranking* global. La Tabla 6.4 muestra los resultados obtenidos cuando consideramos tanto los macro y micro promedios empleando los dos métodos convencionales de cálculo del desempeño del sistema.

Tabla 6.4: Macro y micro precisión: filtrado por categorías de noticias del modelo general.

Macro promedio				
Modelos	P@1	P@3	P@ac	nDCG
BL-CosPP	0,4659447	0,3798107	0,4575006	0,49922
BL-PostGP	0,7950342	0,5408736	0,7902097	0,5477356
Dyadic-Z2	0,5686149	0,4278608	0,5699118	0,487442
Dyadic-Z3	0,5689888	0,4286874	0,5704128	0,4878337
Dyadic-Z5	0,5692002	0,4279905	0,5704751	0,4881231
Dyadic-Z10	0,5695761	0,4290746	0,5709735	0,4884489
Dyadic-Z15	0,5688535	0,4281071	0,5699352	0,4875908
Dyadic-Z20	0,5696533	0,4294967	0,5707063	0,4882353
Triadic-Z2	0,5920298	0,429654	0,5946552	0,4877365
Triadic-Z3	0,5921359	0,4303322	0,5955958	0,4882751
Triadic-Z5	0,5926644	0,4297969	0,5957799	0,4885539
Triadic-Z10	0,5926376	0,4297541	0,5953825	0,4883072
Triadic-Z15	0,59266	0,4309301	0,5958309	0,488627
Triadic-Z20	0,5925927	0,430671	0,5959047	0,488531
Micro promedio				
BL-CosPP	0,4957716	0,3411112	0,4828946	0,5668142
BL-PostGP	0,8356161	0,5146428	0,8306649	0,6194009
Dyadic-Z2	0,6263095	0,3977281	0,6229234	0,5575646
Dyadic-Z3	0,6264303	0,3984745	0,6227707	0,5575893
Dyadic-Z5	0,6266099	0,3983816	0,6230815	0,5578185
Dyadic-Z10	0,627096	0,3993683	0,6236488	0,5582551
Dyadic-Z15	0,6268669	0,3991056	0,6232787	0,5577968
Dyadic-Z20	0,6270248	0,3994003	0,623466	0,5580631
Triadic-Z2	0,6422354	0,4017484	0,6420186	0,5578278
Triadic-Z3	0,642096	0,4022257	0,6422073	0,5579331
Triadic-Z5	0,6429166	0,4022065	0,6428757	0,5585957
Triadic-Z10	0,6431303	0,401899	0,6428406	0,5584626
Triadic-Z15	0,6429693	0,4028504	0,6429541	0,5586267
Triadic-Z20	0,6430436	0,4033181	0,6429558	0,5584347

El macro promedio se calcula al computar primeramente la métrica para cada categoría de noticia y luego obtener el promedio de éstas. En este caso a todas las categorías se le da igual peso de importancia. Sin embargo, para calcular el micro promedio, cada predicción de forma individual tiene el mismo peso y se obtiene una media general. Como consecuencia, en el computo del micro promedio, las categorías más frecuentes juegan un rol importante en el valor de la métrica, al tener un mayor número de predicciones del total e incidir en los resultados.

Como esperábamos los resultados son mejores cuando consideramos un *ranking* global para ambas medidas (macro y micro). Una explicación de estos resultados es que la predicción de un *ranking* general es más difícil que la predicción de un *ranking* para una categoría específica. Esto se debe a la tendencia de las categorías más frecuentes de posicionar sus noticias en mejores lugares en el *ranking*. Por ejemplo, supongamos

que un usuario suele leer noticias deportivas y rara vez lee noticias de cultura. Sea N_g una noticia de cultura que pueda ser de interés para el usuario. Usando su perfil, es muy probable que todas las noticias de cultura sean ubicadas en posiciones inferiores que las noticias de deporte en el *ranking* que se le recomienda al usuario.

Nuevamente, el modelo Triadic muestra una mayor capacidad predictiva que el resto de los modelos empleados. Sorprendentemente, el número de valores para la clase latente no resulta ser muy relevante. Esperábamos notar que con la variación del número de clases latentes, se definieran mejor grupos de usuarios con intereses de lectura bien marcados.

- **Propuesta basada en categorías:** En este caso, se empleará un modelo específico para cada categoría. Para entrenar cada modelo, hemos dividido el conjunto de datos original en 49 diferentes conjuntos de datos, uno por cada una de las categorías presentes. Luego, siguiendo la misma metodología que antes se ha descrito y hemos empleado en el resto de las experimentaciones, cada conjunto de datos se particiona en respectivos conjuntos de entrenamiento, validación y prueba en la relación 70 %, 10 % y 20 %, respectivamente, de los días en los cuales una noticia en dicha categoría fue accedida. Una vez los conjuntos de datos fueron obtenidos se construyeron los diferentes modelos. Se debe destacar que en este caso preciso, no es posible construir el modelo Triadic, al no disponer de noticias de diferentes categorías.

La idea es obtener 49 diferentes perfiles para cada usuario. Cada perfil contiene los términos (y su frecuencia) que aparecen en una determinada categoría dada las noticias que accedió. Estos vectores son entonces empleados en los diferentes modelos de referencia (BL-XXX-Cat en la Tabla 6.5). Con respecto al modelo Dyadic, se entrenaron 49 diferentes modelos, uno por cada categoría (denominados como Dyadic.Cat-Zx en la Tabla 6.5).

Enfocándonos en las diferentes métricas en este experimento se puede decir que las medidas de precisión son inferiores respecto a la propuesta general, básicamente porque hay información valiosa que no se emplea. En otras palabras, existen relaciones entre noticias de diferentes categorías que no se obtienen en estos modelos independientes. Sorprendentemente, nDCG da mejores valores en este último enfoque que se analiza.

De forma general se puede concluir que teniendo las categorías y modelos independientes para ellas, un menor número de noticias relevantes se pueden recomendar, pero en mejores posiciones en el *ranking* específico para dicha categoría. Siguiendo el mismo procedimiento que en la propuesta general, el valor de la clase latente no es relevante y el modelo Dyadic básicamente muestra el mismo comportamiento general con respecto a los modelos de referencia empleados.

Tabla 6.5: Macro y micro precisión: modelos independientes basados en categorías de noticias.

Macro Medida				
Models	P@1	P@3	P@ac	nDCG
BL-CosPP-Cat	0,315471	0,2817824	0,2825813	0,5285388
BL-PostGP-Cat	0,8124582	0,5598107	0,8100626	0,5702044
Dyadic-Cat-Z2	0,3027948	0,2803774	0,2946284	0,5148334
Dyadic-Cat-Z3	0,3184932	0,2818067	0,3084476	0,528657
Dyadic-Cat-Z5	0,3185508	0,2817539	0,3081962	0,5284667
Dyadic-Cat-Z10	0,31197	0,2820352	0,3032364	0,5234387
Dyadic-Cat-Z15	0,3105944	0,2802007	0,3017878	0,5221819
Dyadic-Cat-Z20	0,3177032	0,2812885	0,3078489	0,5280714
Micro Medida				
BL-CosPP-Cat	0,3746567	0,2383982	0,3465239	0,5946918
BL-PostGP-Cat	0,8364596	0,5145217	0,8314996	0,6432068
Dyadic-Cat-Z2	0,5277519	0,3263839	0,5163446	0,586569
Dyadic-Cat-Z3	0,5359533	0,3275605	0,5235446	0,593913
Dyadic-Cat-Z5	0,5362163	0,3281212	0,5236287	0,594077
Dyadic-Cat-Z10	0,5330157	0,3290294	0,5210205	0,5913409
Dyadic-Cat-Z15	0,5317211	0,3261114	0,5201635	0,5906133
Dyadic-Cat-Z20	0,5357586	0,3278843	0,5234428	0,5937149

6.3.3.3. Análisis comparativo entre las categorías

Al analizar los resultados de los diferentes modelos analizados hasta acá, decidimos hacer un análisis en mayor detalle con las categorías de noticias. Para ello se consideraron las 16 categorías más frecuentes: 1, Holguín; 2, Deporte; 3, Huracán Ike; 4, Nacional; 5, Cultura; 6, Internacional; 7, Salud; 8, Reflexiones de Fidel; 9, Opinión; 10, Especiales; 11, Los Cinco; 12, Cuba en Beijing 2008; 13, Homeopatía; 14, Esquina Joven; 15, Aniversario 32 de la Gesta del Moncada; y 16, Página 8.

La Figura 6.6 muestra los resultados de la P@ac para las categorías seleccionadas empleando $Z = 20$, así como (A) una propuesta general y (B) una propuesta basada en las categorías independientes. Para los modelos de referencia se muestran los resultados obtenidos con la medida coseno, empleando una predicción personalizada para ambas propuestas.

Luego de ver el comportamiento de las categorías más frecuentes, se puede llegar a algunas interesantes conclusiones. En primer lugar, los resultados obtenidos con los modelos que utilizan una variable latente superan constantemente los modelos de referencia empleados. En segundo lugar, el empleo del modelo general y la información de todas las categorías de noticias parece ser mejor que emplear un modelo específico por categoría. Consideramos que este enfoque construye mejores perfiles de usuario al disponer de información de las distintas categorías. En este sentido el modelo Triadic se comporta mejor de forma general.

Para confirmar la conclusión previa, se investigó si los resultados dependían del número de noticias en una categoría en el conjunto de entrenamiento. En este estudio se consideraron todas las categorías del conjunto de datos. Por un lado, encontramos que cuando se entrena

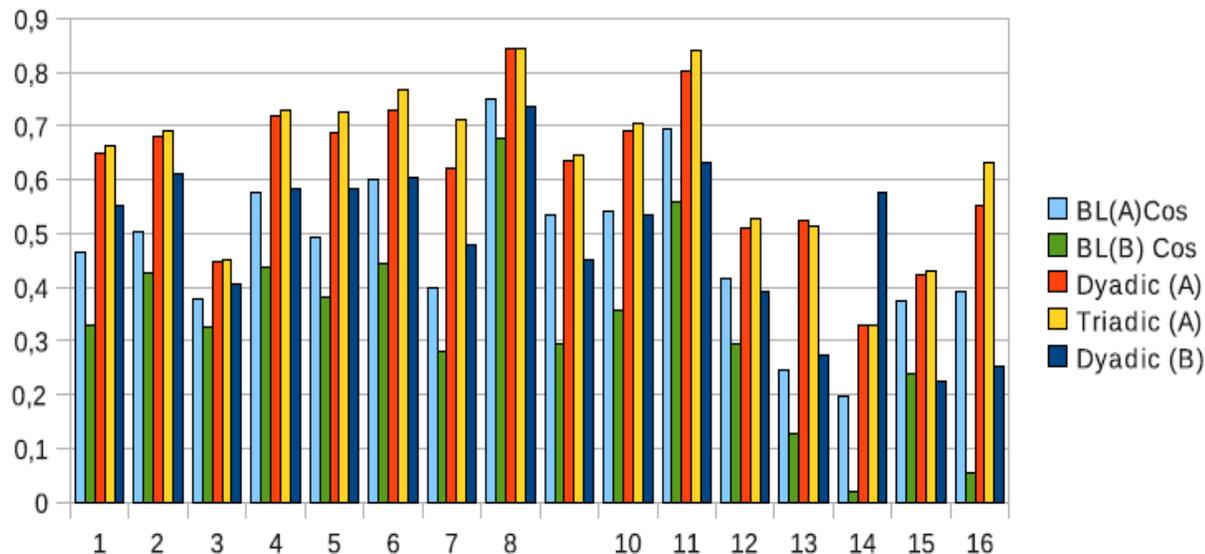


Figura 6.6: P@ac para las 16 categorías más frecuentes.

con modelos independientes por cada categoría de noticia, hay una alta correlación entre el número de noticias del conjunto de entrenamiento y la calidad de la predicción (Correlación de Pearson se encuentra entre 0,55 y 0,60 para todos los modelos), lo cual implica que la calidad de la predicción es peor cuando se tienen pocos datos. Por otro lado, cuando empleamos el modelo general, la frecuencia y la calidad de la predicción no están fuertemente correlacionadas (Correlación de Pearson de 0,15 a 0,20). Esto puede implicar que en muchos casos, la información de categorías relacionadas ayuda en la predicción, reduciendo la dependencia del número de noticias en una categoría.

Finalmente, para la categoría “Esquina Joven” (categoría número 14 en la Figura 6.6), es mejor considerar el modelo basado en dicha categoría independiente, en lugar del modelo general. Sin embargo, este hecho también es relevante ya que esta es una sección popular en la que un columnista emite su opinión sobre diversos temas de interés para los jóvenes, como el sexo, el alcohol y las fiestas, que no están relacionadas con el resto de la información del periódico. Por lo tanto, con la finalidad de obtener las mejores recomendaciones puede ser conveniente seleccionar el mejor modelo para cada categoría de noticia.

6.3.3.4. Análisis de comportamiento de preferencias en el tiempo

Hasta aquí hemos comprobado que nuestra propuesta es capaz de capturar a largo plazo las preferencias de los usuarios. En esta sección exploraremos el desempeño de nuestro enfoque cuando se trata de intereses a corto y largo plazo. Los intereses a corto plazo resulta un suceso común en ciertos usuarios al acceder a los periódicos buscando cierto tipo de información en un momento determinado.

Con el objetivo de explorar estos tipos de intereses, consideraremos los siguientes escenarios:

- **Preferencias a corto plazo:** Por medio de estas preferencias se trata de medir los efectos de los accesos cercanos a un determinado tema en un periodo de tiempo, en nuestro caso, un mes básicamente. Estas preferencias se tendrán en cuenta al dividir el conjunto de datos en 12 subconjuntos, donde cada uno contiene la información asociada a un mes.

Empleando la metodología que hasta ahora hemos utilizado, las noticias accedidas en el último día de cada mes se emplea como conjunto de prueba y el resto de días, para construir el modelo.

- **Preferencias a largo plazo:** Estas preferencias pueden estar fuertemente relacionadas con los accesos continuos y(o) repetitivos de los usuarios sobre un tema determinado. Con el objetivo de capturar estas preferencias dividiremos el conjunto de datos de forma cronológica, por ejemplo, consideraremos los primeros 70 % de días para el conjunto de entrenamiento, los siguientes días que representen el 10 % para el conjunto de validación y el resto para el conjunto de prueba³.

Nosotros estamos interesados en obtener un análisis basado en el comportamiento por categorías, para ello realizaremos un experimento analizando la dinámica temporal y cómo se comporta nuestro modelo en el momento de predecir las noticias relevantes por cada categoría. En los escenarios descritos anteriormente, es posible que algunas categorías tengan pocos o ningún acceso en algún periodo de tiempo. Por lo tanto, y para hacer que los resultados sean comparables, solo serán analizadas las 11 secciones estables analizando el rango de frecuencia. Nótese que para el resto de las categorías, dado que no hay suficiente número de accesos en el conjunto de entrenamiento, los resultados son considerablemente peores. Aunque resulta obvio, esta es una conclusión importante: necesitamos un suficiente número de datos en el conjunto de entrenamiento para obtener un modelo predictivo confiable.

La Tabla 6.6 muestra la macro y micro P@ac obtenida en el análisis de las categorías seleccionadas en el experimento cuando se emplearon las preferencias del usuario a corto (ST) y largo plazo (LT), y también se muestran los valores obtenidos cuando se centra en las categorías seleccionadas en una propuesta general (GB), es decir, no teniendo en cuenta los aspectos relacionados en el tiempo. Se puede observar que el resto de las métricas muestran un comportamiento similar.

³Nótese que en este caso no usaremos una validación cruzada dado que se no dispone de suficiente información.

Tabla 6.6: $P@ac$ cuando empleamos perfiles para corto (ST) y largo (LT) plazo.

	Perfiles	BL-CosPP	Diadic	Triadic	BL-PostGP
Macro	ST	0,4294	0,5974	0,6863	0,7344
	LT	0,3830	0,5077	0,5663	0,8627
	GB	0,5398	0,6829	0,7074	0,8808
	1-ST vs 2-ST	6,23 %	4,81 %	3,40 %	-
Micro	ST	0,4067	0,5810	0,6534	0,8619
	LT	0,3707	0,5153	0,5305	0,8656
	GB	0,5429	0,6834	0,7129	0,8706
	1-ST vs 2-ST	10,32 %	8,47 %	4,81 %	-

Como referencia, se muestra además en la última de las columnas, el valor obtenido en el enfoque a posteriori. En todos los casos se muestra solamente el resultado obtenido siguiendo un enfoque general para $Z = 20$.

Aunque los resultados que hemos obtenido no son directamente comparables (tenemos tres experimentos diferentes, cada uno proporcionando resultados distintos), algunas conclusiones interesantes pueden extraerse. Podemos decir que el desempeño relativo entre los modelos de referencia, Dyadic y Triadic, no parece verse afectado por los criterios analizados del factor tiempo.

Enfocándonos en el uso de perfiles basados en el tiempo, podemos obtener las siguientes conclusiones: primero, parecer ser que mediante un enfoque general (GB) podemos aprender apropiadamente los perfiles de los usuarios, siendo capaces de determinar si un usuario está interesado o no en un tema determinado. Además, si comparamos los resultados con los obtenidos usando solamente las preferencias a largo plazo (LT), podemos inferir que estas preferencias son dinámicas, cambiantes con el tiempo. Por lo tanto, teniendo en cuenta si un lector sigue un asunto determinado o no durante los días previos, parece ser muy relevante predecir los intereses actuales de los usuarios. Esta conclusión está soportada por los resultados obtenidos en la experimentación con las preferencias a corto plazo (ST), que considera esta información junto con las preferencias a largo plazo que prevalecen en el último mes.

Parece que los dos tipos de perfiles basados en el tiempo se refuerzan mutuamente cuando se unen en un único modelo. Con el objetivo de comprobar esta hipótesis, hemos realizado una nueva experimentación. Particularmente, hemos seguido la misma aproximación que la realizada para las preferencias a corto plazo, pero comparando los resultados obtenidos cuando se utiliza solo el último mes (1-ST) con respecto a los obtenidos utilizando los dos meses anteriores (2-ST)⁴.

⁴En el caso de 1-ST, los datos utilizados corresponden a los meses 2^e , 3^e , ..., 12^e y cuando consideramos 2-ST cada conjunto de datos está compuesto de los meses $1^e - 2^e$, $2^e - 3^e$, ..., $11^e - 12^e$.

Nótese que en este último caso estamos aumentando los datos que se utilizan para aprender las preferencias en un plazo de tiempo mayor. Los resultados, en términos de porcentaje de cambio, se muestran en la última fila de la macro y micro medidas de la Tabla 6.6.

Por otra parte, al considerar las diferencias entre las macro y micro medidas, parece que los mejores resultados se obtienen con las categorías más comunes, que, de cierta forma, están relacionadas con las preferencias a largo plazo. En esta línea, los resultados de esta experimentación corroboran los obtenidos con la experimentación general (GB).

Finalmente queremos señalar que para un número suficiente de accesos, los perfiles a corto plazo muestran un buen desempeño. Por lo tanto, la utilización de este tipo de modelos puede ser adecuado para abordar temáticas de corto plazo en las tendencias, como, por ejemplo, los Juegos Olímpicos, huracanes, etc. Para el resto de las categorías, es mejor tener en cuenta los perfiles generales de los usuarios.

De forma general consideramos que con los resultados obtenidos demostramos la capacidad de la propuesta para ordenar un conjunto de noticias en un día determinado, según las preferencias de los usuarios.

6.4. Conclusiones

En este capítulo se han descrito las experimentaciones realizadas dentro de los SR basados en contenido, específicamente en un entorno de noticias. Hemos propuesto dos modelos que permiten mostrar de forma ordenada las noticias a los usuarios según sus intereses, considerando para este propósito las visitas previas de los usuarios a noticias, los términos que contienen los artículos de noticias y las categorías en las que son clasificadas.

Los modelos probabilísticos desarrollados, Dyadic y Triadic, son capaces de identificar relaciones semánticas en los accesos realizados por los usuarios a las noticias clasificadas, información que resulta útil para la recomendación. De forma general un usuario está interesado en determinadas temáticas al acceder a un sitio de noticias y puede que leer el título de una noticia no le represente interés alguno, sin embargo, los modelos analizados son capaces de relacionar el contenido de la noticias accedidas previamente y determinar el interés que un usuario tendrá sobre determinados tópicos noticiosos que se generan en un periódico.

Los resultados obtenidos muestran que al considerar la clasificación de las noticias (categorías) se mejora en la predicción de los intereses del usuario.

Consideramos que dado el volumen de nueva información que tienen los sitios de noticias constantemente, y con el objetivo de ayudar a los usuarios a encontrar noticias de interés, nuestras propuestas son interesantes y competitivas para la recomendación de noticias.

Hemos presentado una detallada evaluación experimental de los algoritmos propuestos donde se ha analizado el desempeño de la propuesta y los diferentes parámetros considerados. Los resultados muestran que en el modelo Triadic, donde se combinan los accesos a las

noticias, los términos y las categorías, se producen recomendaciones de mayor calidad que con el resto de los modelos utilizados.

Capítulo 7

Interfaz de explicación para sistemas de recomendación basados en contenido

En este capítulo se describe una propuesta de explicación para los sistemas de recomendación basados en contenido. La propuesta tiene su origen en el estudio que fue descrito en el Capítulo 4 de esta investigación, por lo que comparten características. Aunque la idea puede ser generalizada en otros SR que manejen contenido, nos centramos en mostrar la utilidad de la misma en la recomendación de noticias.

En los SR de noticias se emplean mecanismos para explicar las recomendaciones apoyándose en la similitud con las noticias accedidas previamente o en la opinión de los usuarios más similares. Como se verá en el capítulo, estos elementos y otros muy importantes son fusionados con la idea de brindarle los argumentos adecuados al usuario para que decida si acepta o no la recomendación recibida.

Partimos de la idea de lo importante que resulta para los usuarios en la vida diaria y en el momento de tomar una decisión, el comparar nuevas oportunidades con la experiencia previa que se tiene. Específicamente en el entorno noticioso, los usuarios suelen acceder a noticias que le resultan de gran interés, pero de forma general llegan a tener intereses de lecturas similares en periodos de tiempo, provocados por intereses personales o sucesos que hagan darle seguimiento a una temática dada.

La interfaz de explicación contiene una nube de etiquetas que representa el perfil del usuario y al mismo tiempo se resalta la relevancia en dicho perfil de la noticia que se recomienda. Además muestra el comportamiento previo de lectura del usuario y la similitud de dichas noticias con la noticia recomendada. El usuario tiene la posibilidad adicional de consultar y(o) seleccionar aquellas noticias que desea aparezcan visualizadas.

De todos los elementos considerados se muestran ejemplos de funcionamiento a lo largo del capítulo que de forma general permiten comprender la propuesta. Además se describen los principales resultados que se obtuvieron al mostrarle la interfaz a un grupo de expertos.

7.1. Propuesta de explicación para sistemas de recomendación basados en contenido

La idea que se presenta en el capítulo, se basa en un modelo de recomendación de contenido y siguiendo la taxonomía propuesta en [84], podemos clasificarla en una explicación con estilo híbrido, sustentado en los ítems visitados previamente y su contenido. Como veremos más adelante, se establece la similitud de la noticia recomendada con las accedidas previamente, y se emplean elementos de dichas noticias para enriquecer la propuesta y ser más útil a los usuarios en el momento de tomar alguna decisión.

Aunque el apoyarse en las noticias previamente visitadas para mostrar una interfaz de explicación no es nuevo, encontraremos una propuesta visual donde la presentación de dichas noticias reporta a los usuarios otros elementos para juzgar la recomendación: Les brinda a los usuarios la posibilidad de visualizar noticias leídas en un espacio bidimensional, encontrando incluso relación con las nuevas noticias. La posición en el gráfico da elementos al usuario para aceptar o no la sugerencia ofrecida por el sistema.

Los usuarios al ver la interfaz de explicación, podrían rápidamente tener una idea de cómo de bien/mal se relaciona la noticia que se les recomienda con sus intereses noticiosos anteriores. El tener una idea de estos elementos le permitirá rápidamente tomar la decisión de leer o no la noticia sugerida.

Consideramos que los ejemplos analizados en la introducción de esta investigación relacionados con las interfaces de explicación brindan información rápida a los usuarios, pero no les permite juzgar situaciones con suficientes elementos. Apoyándonos principalmente en los criterios, ítems accedidos previamente y similitud de estos con la noticia que se recomienda, proponemos una interfaz de explicación que describiremos en lo adelante.

Como ya hemos visto, los SR de noticias, suelen ofrecer sugerencias basándose en la similaridad del contenido accedido en las visitas previas o en aquellas noticias visitadas por usuarios similares. El usuario recibe la recomendación de un conjunto de noticias, ordenadas según el interés precedente. Para lograr este *ranking* se analiza el contenido de todas las noticias accedidas, obteniéndose los k términos que mejor caracterizan las preferencias del usuario [30, 33] y su frecuencia de aparición, elementos que forman el perfil del usuario.

Cuando un usuario accede a noticias sobre una cierta temática, en su perfil se puede comprobar que los términos relacionados con las noticias accedidas incrementan su importancia. Nosotros consideramos el perfil del usuario como un elemento muy importante y lo incluimos en la propuesta. Incorporamos otros elementos a los cuales los usuarios dan gran importancia como son:

- la clasificación de la noticia en categorías,
- el seguimiento a eventos determinados,

- y la fecha de publicación de las noticias.

La hipótesis que se maneja en lo adelante es que un usuario confiará/leerá las noticias sugeridas/recomendadas si recibe argumentos que le permitan comparar la sugerencia con sus intereses de información. Se considera además que, el momento en el tiempo que cada noticia surgió y fue consultada son elementos que resultarían de gran interés. Para usuarios que le interesa el fútbol, por ejemplo, acceder a noticias de fútbol, en días próximos a una nueva jornada de liga es más significativo que en otros momentos de la semana.

Seguidamente pasamos a describir cómo se utiliza toda la información anterior en la propuesta de explicación. Recordemos que la explicación en los SR, soporta la valoración/puntuación de una noticia, además de ayudar a decidir al usuario si finalmente lee la noticia o no [11], por lo que se debe ser cuidadoso en los elementos que se brindan de soporte.

Las propuestas que justifican las recomendaciones ofrecidas, apoyándose en el perfil de términos del usuario, muestran el conjunto de noticias y términos que mejor caracterizan la sugerencia. Otras propuestas soportan su explicación en el interés mostrado por usuarios muy similares que ya han tenido la oportunidad de leer la noticia o bien por el interés despertado por dichas noticias en la comunidad en general.

Aunque en la propuesta empleamos el perfil general del usuario que se obtiene del acceso a todas las noticias sin importar la categoría, valoramos además emplear otros tipos de perfiles más específicos, dada la importancia que obtuvimos del empleo de las categorías en los experimentos del capítulo anterior. Mientras en el perfil general se tienen los principales términos de la totalidad de noticias accedidas y de cada término la cantidad de veces (frecuencia) que se repite, en los perfiles por categoría, de forma similar al perfil general, se tienen solamente los términos de las noticias accedidas de la categoría que se analiza.

El momento de utilizar uno u otro perfil será otro de los temas que se analizarán en futuros estudios. Cuando un usuario lee nuevas noticias, su perfil se actualiza, a la vez que se incrementa y pasa a ser más general, por lo que en determinados momentos la utilización de un perfil más refinado puede ser una opción. Igualmente se debe tener cuidado con el perfil por categoría, pues cuando un usuario lee muy pocas noticias de una categoría, el perfil que se tiene es poco informativo.

Nuestra idea al considerar el perfil, es mostrarle al usuario su comportamiento precedente y al mismo tiempo destacar la importancia de la noticia recomendada. En la Figura 7.1 reflejamos lo anterior. El gráfico muestra la relación que se establece entre el perfil general del usuario y la nueva noticia que se recomienda. Esta nube de términos en su totalidad representa el conjunto de términos que conforman el perfil (el tamaño está en correspondencia a la aparición del término en las noticias leídas previamente), quedando resaltados aquellos términos que forman parte de la noticia sugerida.

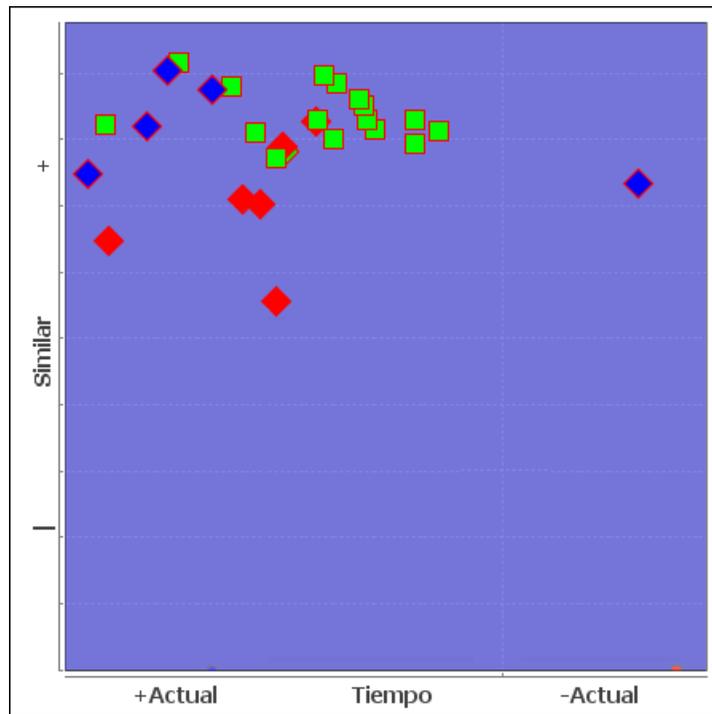


Figura 7.4: Noticias que soportan la explicación.

Este gráfico permite al usuario obtener nichos de noticias sobre temáticas similares. En la Figura 7.5 se destaca un área donde noticias clasificadas en la categoría Cultura, que son bastante similares a la que se recomienda, han sido accedidas por el usuario en un periodo de tiempo corto, lo que pudiera significar el interés mostrado por algún evento cultural en ese lapso.

Las noticias en el gráfico que se analiza están representadas por figuras geométricas, específicamente por cuadrados (□) y diamantes (◇). Un diamante indica al usuario que el modelo de recomendación le hubiese recomendado leer la noticia noticia, mientras un cuadrado indicaría lo contrario. Este elemento es una información adicional que el usuario tiene acerca del comportamiento del sistema hacía sus valoraciones previas.

Los colores de relleno en cada figura indican la categoría a la que pertenece cada noticia, categorías que aparecen en la Figura 7.3. En la Figura 7.4 tenemos más a la izquierda un rombo relleno con el color azul, lo que nos indica que la noticia pertenece a la categoría Economía, mientras la noticia ubicada en la posición más inferior que el resto pertenece a la categoría Deporte.

Al ubicar el cursor sobre cualquiera de las noticias (figura geométrica) en la Figura 7.4, el usuario obtiene un resumen del contenido de la misma (similar a la Figura 7.6), facilitándole rápidamente inspeccionar cualquier región de interés (noticias) que encuentre en el gráfico.

Esta propuesta daría la idea al usuario, de cuan similar es la noticia sugerida respecto al

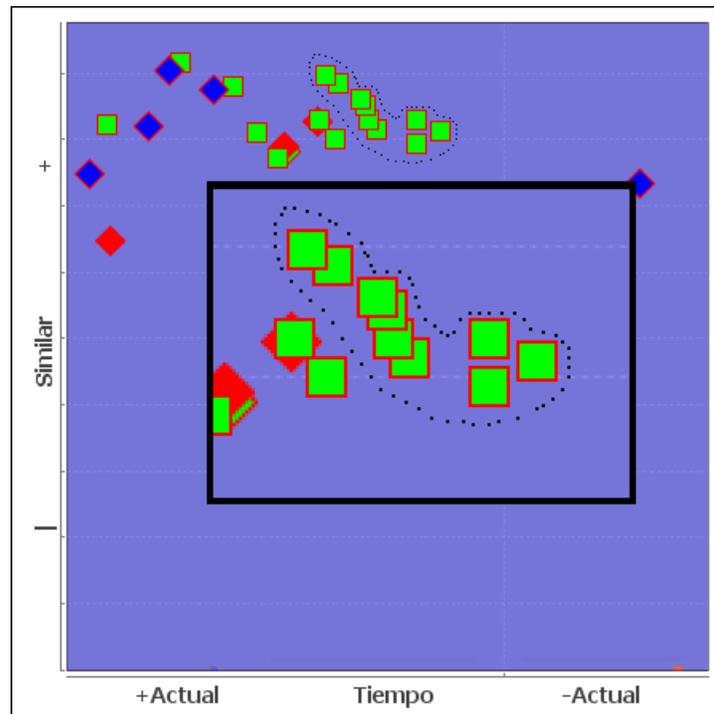


Figura 7.5: Ejemplo de nicho de noticias similares que soportan la explicación.

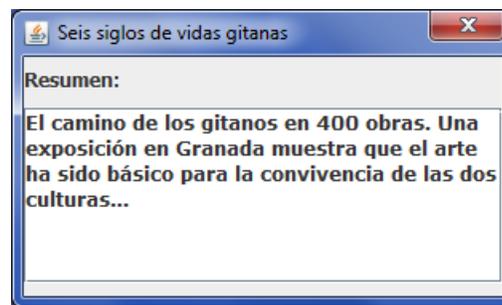


Figura 7.6: Resumen emergente de noticia seleccionada.

conjunto de noticias accedidas previamente e incluso las temáticas que le son más cercanas. Todos los elementos anteriores, quedan distinguidos por el momento en el tiempo en que fue accedida cada noticia. Los usuarios pueden crearse rápidamente una visión espacio - tiempo de sus intereses noticiosos y determinar si finalmente leen o no la noticia.

7.1.1. Ejemplos de uso de la interfaz de explicación

Ya descritos todos los elementos considerados en la propuesta, mostraremos ejemplos donde se refleja la integración de los mismos y simultáneamente describiremos las interpretaciones que se obtienen. Se parte de que al usuario se le mostrará la interfaz de explicación,

por ejemplo, la Figura 7.7, una vez haya recibido un listado de recomendaciones de noticias y decida obtener la explicación del “porqué” se le ha dado una recomendación específica. Por lo tanto, no será una explicación explícita inicialmente, sino que estará condicionada a la solicitud por parte del usuario.



Figura 7.7: Interfaz de explicación para sistemas de recomendación basados en contenido.

El primer ejemplo que mostramos (Figura 7.7) muestra una explicación para una noticia de la categoría Cultura. Esto queda reflejado en la nube de etiquetas donde se resaltan los términos de la noticia recomendada con el color que representa la categoría en la que fue clasificada. Los términos resaltados del perfil que a la vez forman parte de los términos de la noticia sugerida son: lectura, administración, elecciones, publicas, nuevos y millones.

Al presentarse la interfaz, en el espacio inferior aparecen marcadas todas las categorías de noticias, por lo que se mostrarán noticias de todas las categorías. En este ejemplo encontramos noticias que el usuario visitó previamente de las categorías Deporte (color rojo), Cultura (color verde) y Economía (color azul). El usuario puede, entre las opciones disponibles en este espacio, seleccionar aquellas categorías de noticias de las cuales quiere definitivamente ver noticias en la interfaz, personalizándola a sus intereses.

La noticia leída más reciente por el usuario en la Figura 7.7, que pertenece a la categoría Economía (color azul), sería de las recomendadas por el SR (◇).

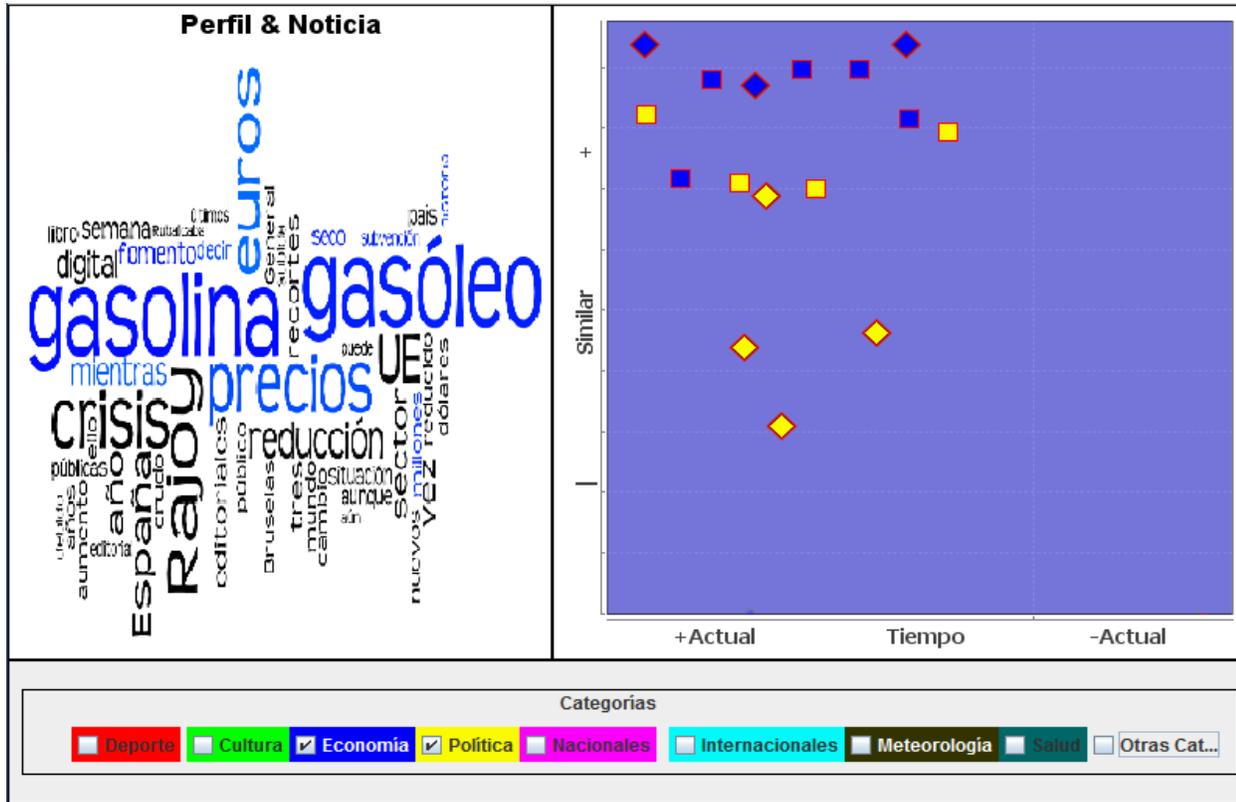


Figura 7.9: Ejemplo de interfaz de explicación propuesta, para una noticia de la categoría Economía. Selección de noticias de las categorías Economía y Política.

Refinando aún más la visualización anterior, la Figura 7.10 muestra solamente noticias de la misma categoría de la noticia recomendada, Economía. De esta forma el usuario puede reducir el número y tipo de noticias que se visualicen en el gráfico.

La interfaz de explicación que se ha propuesto en esta sección, pudiera emplear los resultados de cualquier modelo de recomendación basado en contenido y mostrar las relaciones existentes entre los intereses de información de cada usuario en el tiempo. Evidentemente se deben considerar las peculiaridades del sistema y los datos que se utilizan. En general con la propuesta intentamos brindar al lector argumentos para aceptar las recomendaciones ofrecidas e incrementar la eficiencia de los sistemas de recomendación.

7.2. Evaluación de la propuesta

Para evaluar cualitativamente el impacto de la interfaz, se seleccionó un grupo de expertos con el objetivo de recopilar opiniones y sugerencias para futuras evaluaciones con usuarios reales.

Antes de la evaluación se les facilitó a los expertos un prototipo que contenía la interfaz de explicación. Se les mostró un listado de noticias a recomendar y de cada una de ellas

elementos que se brindan resultan de utilidad para analizar la recomendación. Conclusiones definitivas en este tema serán comprobadas al realizar un estudio con usuarios reales y en el que se empleen otras interfaces.

Finalmente señalan que este tipo de propuestas resultan muy útiles en situaciones donde aceptar una recomendación implica gastos de recursos y se deben analizar varios elementos antes de tomar una decisión. Tener una explicación como la que proponemos resultaría muy útil.

- La interfaz diseñada puede ser generalizada en otros dominios.

Aunque se debe ser cuidadoso con su generalización, el empleo por un lado de la nube de términos para representar el perfil y la forma de resaltar la noticia recomendada, y por otro lado la representación bidimensional para las noticias accedidas previamente, constituyen representaciones interesantes, por ejemplo, en negocios de comercio electrónico que establecen relaciones cruzadas entre sus ítems.

En un mundo digital tan dinámico y rico en elementos como imágenes, textos e íconos, las explicaciones lineales son rechazadas, por lo que esta propuesta puede servir en la explicación de muchos SR.

- La propuesta puede incrementar la confianza de los usuarios en el sistema y su posterior reutilización.

En general los usuarios confían en un sistema luego de obtener beneficios y utilizarlo en un número determinado de veces. Los expertos opinan que la propuesta inicialmente será compleja de entender para los usuarios, y que sólo la utilización frecuente facilitará su uso. Los que pasen a ser usuarios frecuentes, encontrarán en la interfaz argumentos para aceptar o no las recomendaciones, incluso llegando a utilizarla para evitar leerse innecesariamente documentos muy largos.

- Los elementos visuales que se emplean (colores, figuras) resultan de agrado para los usuarios.

La utilización de colores para diferenciar las categorías en las que pueden ser clasificadas las noticias se convierte en un recurso visual importante para ubicar y transmitirle información al usuario. Su empleo tanto en la nube de términos para resaltar la noticia recomendada, como en el componente bidimensional permite al usuario distinguir entre las noticias que se le muestran.

Las figuras empleadas brindan la información al usuario de la capacidad de predicción del modelo de recomendación empleado, y por tanto, la coincidencia de este con su interés noticioso previo. Con esta información el usuario tiene otro elemento para juzgar la recomendación.

- Se logró diseñar una interfaz intuitiva con adecuados recursos gráficos.

La disposición y relación de los componentes en la interfaz, permite a los usuarios inspeccionar toda la información que dispone de manera coherente. El usuario encuentra componentes que le permiten personalizar la interfaz a sus intereses.

Pudieran incluirse otros componentes de control para filtrar la información que se muestra en el componente bidimensional, por ejemplo, el intervalo de tiempo para mostrar las noticias o un umbral de similitud entre la noticia que se recomienda y las accedidas.

- Las noticias consultadas previamente, su descripción de acceso temporal y la nube de términos empleada para describir el perfil, son elementos que enriquecen la opinión del usuario para tomar una decisión.

Toda la información que se muestra pudiera encontrarse en anteriores propuestas de forma separada. Pero en esta propuesta se logra fusionar mucha información de manera oportuna.

El usuario recibe información de elementos que le son importantes y que normalmente emplea para tomar decisiones. La posibilidad de consultar rápidamente las noticias accedidas previamente y al mismo tiempo visualizar su perfil de interés, son características que resaltan en la propuesta.

Otros elementos como la probabilidad de aceptar la recomendación ofrecida después de consultar la interfaz de explicación, la satisfacción del usuario con la explicación, así como la probabilidad de realizar nuevas consultas, son datos que se obtendrán en estudios futuros que se desarrollarán con usuarios reales. Dadas las características de la experimentación, es necesario disponer de un periodo de tiempo razonable para que el estudio sea válido. Otro elemento importante es lograr varias iteraciones de intercambio por parte de los usuarios.

Los estudios futuros incluirán interfaces que han sido utilizadas en otras investigaciones y sirven de base para investigaciones sobre la temática, como por ejemplo, la empleada en Amazon donde se muestran los ítems similares por contenido como argumento de la recomendación. De esta forma y similar a como se realizó el estudio experimental del Capítulo 4, los usuarios evaluarán las explicaciones recibidas y en general nuestra propuesta.

7.3. Conclusiones

Conociendo la desconfianza que suele existir en el proceso predictivo de los SR por parte de los usuarios, se logra una propuesta que emplea elementos que resultan útiles a los usuarios en su vida diaria y sirven para explicar las recomendaciones, lo que provoca la aceptación de las sugerencias. En el capítulo se hace una propuesta de explicación para los SR basados en contenido y se utiliza el entorno noticioso para describir los diferentes componentes.

En general, la propuesta de explicación se basa, por un lado, en la importancia que normalmente tiene para los usuarios la actualidad de las noticias, y por otro, el seguimiento que se suele hacer en el tiempo sobre determinadas temáticas o sucesos noticiosos. Se logra visualizar a través de una nube de etiquetas el perfil del usuario y se destaca la relevancia de la noticia que se recomienda. Por otro lado, se observan las noticias accedidas previamente y la categoría en las que están clasificadas. En esta última visualización la posición de cada noticia representa, por una lado, cuan reciente en el tiempo fue la misma, y por otro, cuan similar es con la noticia que se recomienda.

Aunque no hemos concluido el estudio con usuarios reales, la opinión inicial de un grupo de expertos nos confirma la utilidad, generalización y aceptación que puede tener la propuesta.

Consideramos que se les brinda a los usuarios elementos importantes para valorar adecuadamente la recomendación, y decidir finalmente si aceptar o rechazar la sugerencia. Además se les proporciona una herramienta de consulta y apoyo para posteriores accesos al sistema.

Capítulo 8

Conclusiones y trabajos futuros

8.1. Conclusiones

Inicialmente en esta investigación nos propusimos como objetivo diseñar modelos de recomendación que incrementaran la eficiencia de la recomendación en los SR. En la memoria hemos mostrado toda la investigación desarrollada con los SR, tanto en los basados en contenido como en los de filtrado colaborativo. Nuestras propuestas a lo largo de la memoria se enfocan en las principales tareas asociadas a los SR, la *predicción de voto* y el *encontrar buenos ítems*.

En la primera de las partes, relacionada completamente con el filtrado colaborativo, se expusieron detalladamente todas las características de las propuestas y se comentaron los resultados experimentales. En el primero de los capítulos de esta parte se describió un nuevo método de selección del vecindario que permite incrementar la precisión de las recomendaciones. Luego mostramos en el siguiente capítulo una propuesta de explicación para estos tipos de sistemas y el estudio experimental realizado con usuarios reales. Finalmente, describimos en el tercer capítulo una propuesta que nos permite aprender de las explicaciones para mejorar la precisión de las recomendaciones.

En el primero de los capítulos de la segunda parte, relacionada con los SR basados en contenido, describimos los modelos propuestos empleados para predecir a los usuarios las noticias de interés, y comentamos los resultados alcanzados en toda la experimentación. En el siguiente capítulo y último, se describe una propuesta de explicación para este tipo de sistemas.

En lo adelante exponemos las principales conclusiones que sacamos como resultado de las tareas desarrolladas y el análisis de los diferentes estudios experimentales. En relación con los modelos y propuestas de explicación los SR de filtrado colaborativo tenemos:

- La propuesta de selección de los mejores vecinos para realizar la recomendación, resulta un método competitivo. La idea de estimar la capacidad predictiva como criterio para

seleccionar los vecinos se convierte en una adecuada alternativa y se refleja en los resultados de precisión obtenidos.

- La calidad y tamaño del vecindario están directamente relacionados con la mejora de precisión de la recomendación en los métodos basados en vecindarios. La calidad de cada vecino depende de su capacidad predictiva y del número de ítems en común con el usuario al que se le recomienda. Los resultados experimentales reflejan, por un lado, que con el aumento del vecindario se mejora la precisión, mientras por otro lado, que considerar un mínimo de ítems en común con el usuario activo, permite seleccionar mejores vecinos.
- La evaluación del impacto en la recomendación de determinadas características del conjunto de datos o del usuario, reflejan dependencias que otras métricas clásicas como la MAE o RMSE, no la muestran. Al computar una de las métricas propuestas, wMAE, podemos observar:
 - El error que se obtiene en las predicciones es altamente dependiente del patrón de votos de los usuarios, y menos dependiente de la popularidad del ítem. Los usuarios están influenciados por su forma de votar y no tanto por la popularidad que tenga un determinado ítem.
 - Cuando se estiman las preferencias de votos más frecuentes, se predice de manera más efectiva que cuando se intenta predecir votos menos frecuentes. Al evaluar el beneficio de la wMAE respecto a la MAE, se comprueba esta dependencia entre el voto y el error.
 - Al computar la métrica propuesta wMAPE, relacionada con el error relativo, se comprueba que el método que utiliza la propuesta de selección de vecinos, se ajusta mejor a las preferencias comunes del usuario. Además se verifica que los modelos *Slope One* y SVD están más sesgados hacia la distribución de votos en general del sistema.
- La propuesta de explicación, que se soporta en el criterio del vecindario sobre las experiencias previas, resulta eficaz, y con su empleo, se evidencian resultados superiores que con otras explicaciones. Nuestra explicación muestra la relación entre los ítems accedidos previamente y el ítem para el que se ofrece la recomendación, ofreciéndole al usuario argumentos para comprender las razones de la recomendación y finalmente persuadirlo a aceptarla o no.
- Los resultados experimentales con usuarios reales, muestran la utilidad de la propuesta de explicación en un SR de películas, mejorando la MAE que se obtiene originalmente con el modelo de recomendación.

- El aprendizaje basado en la información que se obtiene de las explicaciones, permite corregir el error cometido en las predicciones. La estrategia de inducir árboles con la información recopilada, y el posterior análisis del árbol para determinar si lanzar o no la regla, resulta adecuada para propósitos predictivos. La experimentación muestra que entre el 25 % y 35 % de los usuarios se benefician con la corrección del error al utilizar la estrategia anterior.
- Los resultados que se obtienen en la experimentación anterior, confirman la influencia entre otras variables, del voto recomendado y el histograma del vecindario en la decisión de aceptar o no la recomendación. Otra característica importante de la propuesta de explicación resulta ser el número de elementos en cada región de la propuesta, a la que los usuarios le prestan una especial atención. La importancia de estas características se logra comprobar con la metodología experimental empleada, donde se computa el rendimiento con la incorporación o no de cada una de las variables.

Las conclusiones relacionadas con los SR basados en contenido son:

- Los modelos propuestos, Dyadic y Triadic, identifican las relaciones semánticas en los accesos realizados por los usuarios a las noticias clasificadas, información que se utiliza para determinar el interés del usuario y recomendarle nuevas noticias. Con ambos modelos se logran desempeños superiores que con el resto de los algoritmos utilizados, resultando propuestas válidas para la recomendación de noticias.
- La evaluación experimental muestra que el modelo Triadic, donde se combinan la información de los accesos a las noticias de los usuarios, además de la composición (términos) y clasificación de las noticias, produce recomendaciones de mayor calidad que con el resto de los modelos utilizados.
- La evaluación con perfiles basados en el tiempo corrobora que la utilización de perfiles a corto plazo, puede ser adecuada para darle seguimiento a temáticas de corta duración en las tendencias de noticias, como por ejemplo, Juegos Olímpicos y huracanes. Para otras categorías, donde las temáticas y noticias tienden a ser recurrentes y se relacionan, los perfiles generales son los más adecuados.
- La propuesta de explicación brinda argumentos importantes para valorar la recomendación. Por un lado, muestra la relevancia de la noticia recomendada en el perfil del usuario, y por otro, la relación de esta noticia con las accedidas previamente, tanto temporalmente como en contenido.
- Aunque para describir la propuesta de explicación se utiliza el entorno noticioso, pudiera emplearse en otros entornos que manejen contenido.

8.2. Trabajos futuros

Las tareas futuras que consideramos para continuar con la investigación que se mostró en esta memoria, las exponemos a continuación:

- Generalizar la métrica propuesta para evaluar la importancia de determinadas características en la recomendación.

La métrica propuesta permite comprobar la importancia de características específicas de los conjuntos de datos, razón por la que se deben realizar experimentaciones para comprobar la importancia de otras características del conjunto de datos seleccionado. Igualmente se debe comprobar con otros conjuntos de datos y algoritmos para verificar las tendencias y resultados obtenidos.

- Generalizar las propuestas de explicación para SR basados en filtrado colaborativo.

La aceptación y opinión de los usuarios al intercambiar con la propuesta, nos confirma la idea de la posibilidad de generalización en otros entornos, donde las recomendaciones deben estar adecuadamente fundamentadas para que los usuarios las acepten. Por ejemplo, recomendaciones asociadas a productos con un costo determinado.

- Determinar las características más importantes de las explicaciones.

Los estudios experimentales realizados en la investigación para detectar el impacto de las características de las explicaciones en las recomendaciones, servirán de guía para desarrollar otros estudios que permitan detectar las características más importantes a considerar en las explicaciones de las recomendaciones.

- Estudiar la mejor configuración para la extracción de información en la propuesta de explicación.

La tarea de determinar cuáles son los mejores ítems que explican la recomendación a los usuarios en la propuesta de explicación, es muy interesante. Los resultados de esta tarea permitirán obtener un parámetro para configurar la explicación, incluso independiente para cada usuario. Su correcta determinación permitirá mejorar los resultados obtenidos y personalizar aún más las explicaciones.

- Estudio experimental para comprobar el aprendizaje basado en la explicación con diferentes métodos de aprendizaje y parámetros experimentales.

Dado que el algoritmo utilizado en el aprendizaje basado en la explicación para la mejora de la recomendación, por una lado, no logra detectar los grandes errores y además está cercanamente ligado a las características del conjunto de datos, sería interesante utilizar un enfoque de aprendizaje diferente, el cual sea capaz de trabajar con conjuntos de datos

no balanceados y usar la métrica RMSE, en lugar de la MAE, como función objetivo de los algoritmos.

- Estudio de la dependencia de la mejora de la MAE en el aprendizaje basado en la explicación, con el contenido de los ítems.

Considerando los resultados experimentales en todos los usuarios, sería oportuno realizar estudios teniendo en cuenta la información disponible de los ítems. Por ejemplo, comprobar cuál es la tendencia al analizar cada uno de los tipos de ítems.

- Evaluación experimental de la propuesta de explicación para SR basados en contenido.

La experimentación con usuarios reales permitirá obtener datos para: evaluar la probabilidad de aceptar la recomendación ofrecida después de consultar la interfaz de explicación, la satisfacción del usuario con la explicación, así como la probabilidad de realizar nuevas consultas. Es necesario realizar varias iteraciones experimentales, por lo que se necesita de un periodo de tiempo razonable para que el estudio sea válido. Los estudios incluirán interfaces utilizadas en otras investigaciones y que sirven de base para investigaciones sobre la temática, como por ejemplo, la empleada en Amazon, donde se muestran los ítems similares por contenido como argumento de la recomendación. Los usuarios deben determinar la utilidad de cada una de las interfaces que se les muestran.

- Generalizar las propuestas de explicación para SR basados en contenido.

Aún sin realizar el estudio anterior, la opinión inicial de los usuarios que intercambiaron con la aplicación, nos hace pensar en la idea de aplicación y generalización de la propuesta.

- Experimentación con otras colecciones de datos.

La necesaria evaluación de la precisión de los modelos resultantes de esta investigación con otras colecciones de datos es una de las tareas futuras. Lo primero y relacionado con la escalabilidad, es la posibilidad de medir el rendimiento en colecciones de mayor tamaño. Por otro lado, es importante comprobar la no dependencia de las características de los conjuntos de datos utilizados en esta investigación.

Apéndice A

Cuestionario a usuarios

Estimado usuario:

Los sistemas de recomendación juegan un rol importante en la actualidad, donde el comercio electrónico forma parte de nuestras vidas. Aplicaciones como Amazon, Netflix, Jester, Facebook y otras más del mundo digital, realizan un gran número de recomendaciones a los usuarios para ayudarlos ante la creciente cantidad de información a procesar. Estas recomendaciones suelen estar relacionadas con las experiencias previas de los usuarios o con la opinión de un grupo de usuarios (vecinos / colaboradores) con gustos o preferencias muy similares.

Las preguntas que forman parte de esta encuesta, están enfocadas en evaluar la calidad de las explicaciones que reciben los usuarios en los sistemas de recomendación respecto a un ítem en cuestión. Usted ha percibido enfoques de explicación diferentes y nuestro interés es conocer que importancia les concede a un grupo de criterios (experiencias previas, predicciones de los vecinos, complejidad, entre otros) presentes y evaluables en dichas propuestas.

Todas las preguntas del cuestionario están enfocadas en evaluar la satisfacción y capacidad de persuadir que se logra con una interfaz de explicación. Le pedimos que al llenar el cuestionario se enfoque en las propuestas en que le fueron mostradas las explicaciones y el efecto causado en usted.

No le solicitamos ningún dato personal, el cuestionario es completamente anónimo. Cada una de las preguntas debe ser evaluada en una escala de 1 a 10, donde el 1 es considerado por usted el estado menos significativo y el 10 indica que usted califica dicha característica como muy significativa. Le pedimos que señale siempre su elección correspondiente.

De antemano muchas gracias por todo.

Valore de uno (1 = Falso) a diez (10=Cierto) los siguientes ítems:

▪ La transparencia dentro de los sistemas de recomendación es muy importante.
Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto

▪ Un sistema de recomendación debe mostrar la explicación de la recomendación ofrecida.
Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto

- Las explicaciones en los sistemas de recomendación ayudan a que dichos sistemas sean más transparentes.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- Una interfaz de explicación debe visualizar cómo los usuarios similares votaron al producto recomendado.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- Visualizar las experiencias previas del usuario activo ayuda a la explicación.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- Se ofrece una mejor explicación si se utilizan tanto las experiencias previas como los votos de usuarios similares.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- Para volver a utilizar el sistema de recomendación, necesita recibir adecuadas explicaciones.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- Si el sistema dispone de explicaciones, aceptaría una recomendación sin consultar las explicaciones dadas por el sistema.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- La interfaz de explicación debe ser fácil de interpretar, funcional y operativa.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- La interfaz de explicación “Basada en Vecindario” ha sido de ayuda para incrementar la confianza en el vecindario.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- La interfaz de explicación “Basada en Historial” ha sido de ayuda para incrementar la confianza en el vecindario.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- La interfaz de explicación “Basada en Vecindario+Historial” ha sido de ayuda para incrementar la confianza en el vecindario.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto
- La información recibida en la interfaz “Basada en Vecindario”, proporciona suficiente ayuda para tomar una decisión.

Falso 1 __ 2 __ 3 __ 4 __ 5 __ 6 __ 7 __ 8 __ 9 __ 10 __ Cierto

- La interfaz de explicación “Basada en Vecindario”, contiene información relevante y útil para tomar decisiones.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- La información recibida en la interfaz “Basada en Historial”, proporciona suficiente ayuda para tomar una decisión.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- La interfaz de explicación “Basada en Historial”, contiene información relevante y útil para tomar decisiones.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- La información recibida en la interfaz “Basada en Vecindario+Historial”, proporciona suficiente ayuda para tomar una decisión.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- La interfaz de explicación “Basada en Vecindario+Historial”, contiene información relevante y útil para tomar decisiones.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- En un sistema de recomendación destinado a películas, prefiero la información dada en la interfaz “Basada en Vecindario”.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- En un sistema de recomendación destinado a asignaturas, prefiero la información dada en la interfaz “Basada en Vecindario”.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- En un sistema de recomendación destinado a restaurantes, prefiero la información dada en la interfaz “Basada en Vecindario”.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- En un sistema de recomendación destinado a prendas de vestir, prefiero la información dada en la interfaz “Basada en Vecindario”.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto
- En un sistema de recomendación destinado a libros, prefiero la información dada en la interfaz “Basada en Vecindario”.
Falso 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ 6 ___ 7 ___ 8 ___ 9 ___ 10 ___ Cierto

Bibliografía

- [1] Gediminas Adomavicius y Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
- [2] Ahmad M. Ahmad Wasfi. Collecting user access patterns for building user profiles and collaborative filtering. In *Proceedings of the 4th international conference on Intelligent user interfaces*, IUI '99, pages 57–64, 1999.
- [3] Chris Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, 2006.
- [4] Robin Anil, Sean Owen, Ted Dunning, y Ellen Friedman. *Mahout in Action*. Manning Publications Co., 2010.
- [5] Yossi Azar, Aviv Nisgav, y Boaz Patt-Shamir. Recommender systems with non-binary grades. In *Proceedings of the 23rd Association for Computing Machinery (ACM) symposium on Parallelism in algorithms and architectures*, SPAA '11, pages 245–252, New York, NY, USA, 2011. Association for Computing Machinery.
- [6] Marko Balabanovic y Yoav Shoham. Content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72, 1997.
- [7] Justin Basilico y Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 65–72, New York, NY, USA, 2004. Association for Computing Machinery.
- [8] Benjamin B. Bederson, Ben Shneiderman, y Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, October 2002.

- [9] Robert M. Bell y Yehuda Koren. Lessons from the netflix prize challenge. *Special Interest Group on Knowledge Discovery and Data Mining, SIGKDD Explorations Newsletter*, 9:75–79, December 2007.
- [10] Mustafa Bilgic y Raymond J. Mooney. Explaining recommendations: Satisfaction vs. promotion. In *Proceedings of Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research at the 2005 International Conference on Intelligent User Interfaces*, pages 13–18, San Diego, CA, January 2005.
- [11] Daniel Billsus y Michael J. Pazzani. A personal news agent that talks, learns and explains. In *Proceedings of the third annual conference on Autonomous Agents*, AGENTS '99, pages 268–275, New York, NY, USA, 1999. Association for Computing Machinery.
- [12] Daniel Billsus y Michael J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10:147–180, February 2000.
- [13] Christopher Bishop y Christopher K. I. Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21:203–224, 1998.
- [14] J. Bobadilla, F. Serradilla, y J. Bernal. A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, 23(6):520–528, August 2010.
- [15] J. Bobadilla, F. Serradilla, y A. Hernando. Collaborative filtering adapted to recommender systems of e-learning. *Knowledge-Based Systems*, 22(4):261–265, May 2009.
- [16] Toine Bogers y Antal van den Bosch. Comparing and evaluating information retrieval algorithms for news recommendation. In *Proceedings of the 2007 ACM Conference on Recommender systems*, RecSys '07, pages 141–144, 2007.
- [17] Leppik D. Konstan J. Borchers, A. y J. Riedl. Partitioning in recommender systems. Technical report, University of Minnesota - Computer Science and Engineering, 1998.
- [18] Matthew Brand. Fast online svd revisions for lightweight recommender systems. In *In Society for Industrial and Applied Mathematics, SIAM International Conference on Data Mining (SDM)*, 2003.
- [19] John S. Breese, David Heckerman, y Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

- [20] R. Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Science*, 69(32), 2000.
- [21] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, November 2002.
- [22] Robin Burke. The adaptive web. chapter Hybrid web recommender systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007.
- [23] Emma Byrne. A logical framework for identifying and explaining unexpected news. *Computers and Artificial Intelligence*, 25(2-3):127–152, 2006.
- [24] Luis M. Campos, Juan M. Fernández-Luna, Juan F. Huete, y Miguel A Rueda-Morales. Managing uncertainty in group recommending processes. *User Modeling and User-Adapted Interaction*, 19(3):207–242, August 2009.
- [25] Iván Cantador, Alejandro Bellogín, y Pablo Castells. News@hand: A semantic web approach to recommending news. In Wolfgang Nejdl, Judy Kay, Pearl Pu, y Eelco Herder, editors, *AH*, volume 5149 of *Lecture Notes in Computer Science*, pages 279–283. Springer, 2008.
- [26] Iván Cantador, Alejandro Bellogín, y Pablo Castells. Ontology-based personalised and context-aware recommendations of news items. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, WI-IAT '08, pages 562–565, Washington, DC, USA, December 2008. IEEE Computer Society.
- [27] Li Chen y Pearl Pu. Trust building in recommender agents. In *in 1st International Workshop on Web Personalization, Recommender Systems and Intelligent User Interfaces (WPRSIUI05)*, pages 135–145, 2005.
- [28] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, y Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, California, 1999. Association for Computing Machinery.
- [29] Sergio Cleger-Tamayo, Juan M. Fernández-Luna, y Juan F. Huete. A new criteria for selecting neighborhood in memory-based recommender systems. In *Conferencia de la Asociación Española para la Inteligencia Artificial, CAEPIA*, pages 423–432, 2011.

- [30] Sergio Cleger-Tamayo, Juan M. Fernández-Luna, y Juan F. Huete. Top-n news recommendations in digital newspapers. *Knowledge-Based Systems*, 27:180–189, March 2012.
- [31] Sergio Cleger-Tamayo, Juan M. Fernández-Luna, Juan F. Huete, Ramiro Pérez-Vázquez, y Julio C. Rodríguez Cano. A proposal for news recommendation based on clustering techniques. In *Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE (3)*, pages 478–487, 2010.
- [32] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, y Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, pages 271–280, 2007.
- [33] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, y Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, pages 271–280, 2007.
- [34] Luis M. de Campos, Juan M. Fernández-Luna, y Juan F. Huete. e-bay.net: Helping users to buy in e-commerce applications. In Eduard Montseny y Pilar Sobrevilla, editors, *European Society for Fuzzy Logic and Technology, EUSFLAT Conference*, pages 565–570. Universidad Polytechnica de Catalunya, 2005.
- [35] Luis M. de Campos, Juan M. Fernández-Luna, y Juan F. Huete. A bayesian network approach to hybrid recommending systems. In Eduard Montseny y Pilar Sobrevilla, editors, *In: Eleventh International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 2158–2165, 2006.
- [36] A. P. Dempster, N. M. Laird, y D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [37] Mukund Deshpande y George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.
- [38] Christian Desrosiers y George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 107–144. 2011.

- [39] Dónal Doyle, Alexey Tsymbal, y Pádraig Cunningham. A review of explanation and explanation in case-based reasoning. Number TCD-CS-2003-41, 2003.
- [40] Martin Ester, Hans peter Kriegel, Jörg S, y Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. Association for the Advancement of Artificial Intelligence, AAAI Press, 1996.
- [41] Tom Fawcett y Foster Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, 1999.
- [42] Gerhard Friedrich y Markus Zanker. A taxonomy for generating explanations in recommender systems. *Association for the Advancement of Artificial Intelligence, AI Magazine*, 32(3):90–98, 2011.
- [43] Evgeniy Gabrilovich, Susan Dumais, y Eric Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 482–490, New York, NY, USA, 2004. Association for Computing Machinery Press.
- [44] Susan Gauch, Mirco Speretta, Aravind Chandramouli, y Alessandro Micarelli. The adaptive web. chapter User profiles for personalized information access, pages 54–89. Springer-Verlag, Berlin, Heidelberg, 2007.
- [45] David Goldberg, David A. Nichols, Brian M. Oki, y Douglas B. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the Association for Computing Machinery*, 35(12):61–70, 1992.
- [46] Ken Goldberg, Theresa Roeder, Dhruv Gupta, y Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, July 2001.
- [47] Asela Gunawardana y Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *Machine Learning Research*, 10:2935–2962, 2009.
- [48] D. Gupta, M. Digiovanni, H.Ñarita, y K. Goldberg. Jester 2.0: A new linear-time collaborative filtering algorithm applied to jokes. In *ACM-SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.
- [49] Jeffrey Heer y Ed H. Chi. Separating the swarm: Categorization methods for user sessions on the web. In *ACM Human Factors in Computing Systems (CHI)*, pages 243–250, 2002.

- [50] Jon Herlocker, Joseph A. Konstan, y John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5:287–310, October 2002.
- [51] Jonathan L. Herlocker. Position statement - explanations in recommender systems. In *Conference on Human Factors in Computing Systems, CHI'99 Workshop, Interacting with Recommender Systems*, 1999.
- [52] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, y John Riedl. An algorithmic framework for performing collaborative filtering. In *Special Interest Group on Information Retrieval, SIGIR*, pages 230–237, 1999.
- [53] Jonathan L. Herlocker, Joseph A. Konstan, y John Riedl. Explaining collaborative filtering recommendations. In *CSCW '00: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, pages 241–250, New York, NY, USA, 2000. Association for Computing Machinery.
- [54] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, y John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53, January 2004.
- [55] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI99*, pages 289–296, 1999.
- [56] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.
- [57] Adele E. Howe y Ryan D. Forbes. Re-considering neighborhood-based collaborative filtering parameters in the context of new data. In *Proceeding of the 17th ACM conference on Information and knowledge management, CIKM '08*, pages 1481–1482, New York, NY, USA, 2008. Association for Computing Machinery.
- [58] Juan F. Huete, Juan M. Fernández-Luna, Luis M. de Campos, y Miguel A. Rueda-Morales. Using past-prediction accuracy in recommender systems. *Information Science*, 199:78–92, 2012.
- [59] Jamil A.S. Itmazi. *Sistema flexible de Gestión del eLearning para soportar el aprendizaje en las Universidades tradicionales y abiertas*. Tesis doctoral, Universidad de Granada, 2005.
- [60] Anil K. Jain, M.Ñarasimha Murty, y Patrick J. Flynn. Data clustering: A review. *ACM Computing Survey*, 31(3):264–323, 1999.

- [61] Anthony Jameson y Barry Smyth. The adaptive web. chapter Recommendation to groups, pages 596–627. Springer-Verlag, Berlin, Heidelberg, 2007.
- [62] M. Kagie, M.C. van Wezel, y P.J.F. Groenen. An empirical comparison of dissimilarity measures for recommender systems. Research Paper ERS-2009-023-MKT, Erasmus Research Institute of Management (ERIM), ERIM is the joint research institute of the Rotterdam School of Management, Erasmus University and the Erasmus School of Economics (ESE) at Erasmus Uni, May 2009.
- [63] L. Kaufman y P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley and Sons, New York, 1990.
- [64] Jae Kyeong Kim y Yoon Ho Cho. Using web usage mining and svd to improve e-commerce recommendation quality. In Jaeho Lee y Mike Barley, editors, *Intelligent Agents and Multi-Agent Systems, 6th Pacific Rim International Workshop on Multi-Agents, PRIMA 2003, Seoul, Korea, November 7-8, 2003, Proceedings*, volume 2891 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2003.
- [65] Youngji Kim, Sooho Ok, y Yongtae Woo. A case-based recommender system using implicit rating techniques. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, AH '02, pages 522–526, London, UK, UK, 2002. Springer-Verlag.
- [66] T. Kohonen, M. R. Schroeder, y T. S. Huang, editors. *Self-Organizing Maps*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2nd edition, 1997.
- [67] Joseph A. Konstan, BradleyÑ. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, John Riedl, y High Volume. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40:77–87, 1997.
- [68] Carmen Lacave y Francisco J. Diez. A review of explanation methods for heuristic expert systems. *The Knowledge Engineering Review*, 19(02):133–146, 2004.
- [69] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the twelfth International conference on machine learning (ICML-95)*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.
- [70] Steffen L. Lauritzen. The em algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19(2):191–201, February 1995.

- [71] Daniel Lemire y Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of Society for Industrial and Applied Mathematics, SIAM Data Mining (SDM'05)*, 2005.
- [72] Daniel Lemire y Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. *Computing Research Repository, CoRR*, abs/cs/0702144, 2007.
- [73] Cane Wing-ki Leung, Stephen Chi-fai Chan, y Fu-lai Chung. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, 21(7):515–529, October 2008.
- [74] T.-P. Liang y H.-J. Lai. Discovering user interests from web browsing behavior: An application to internet news services. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7*, HICSS '02, page 203, 2002.
- [75] G. Linden, B. Smith, y J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.
- [76] Harold A. Linstone y Murray Turoff, editors. *The Delphi Method: Techniques and Applications*. 2002.
- [77] Jiahui Liu, Peter Dolan, y Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proceeding of the 14th international conference on Intelligent user interfaces*, IUI '10, pages 31–40, 2010.
- [78] Linyuan Lü, Matus Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, y Tao Zhou. Recommender systems. *Computing Research Repository, CoRR*, 1202.1112, 2012.
- [79] Kevin Mccarthy, James Reilly, Lorraine Mccinty, y Barry Smyth. Thinking positively - explanatory feedback for conversational recommender systems. In *In Proceedings of the European Conferences on Case-Based Reasoning, ECCBR 2004 Workshops*, 2004.
- [80] J. B. McQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam y J. Neyman, editors, *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [81] David McSherry. Explanation in recommender systems. *Artificial Intelligence Review*, 24(2):179–197, October 2005.

- [82] Kenrick J. Mock y V. Rao Vemuri. Information filtering via hill climbing, wordnet and index patterns. *Information Processing and Management*, 33:633–644, September 1997.
- [83] MovieLens. Data sets <http://www.grouplens.org/>.
- [84] Alexis Papadimitriou, Panagiotis Symeonidis, y Yannis Manolopoulos. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery*, pages 1–29, 2011.
- [85] Michael J. Pazzani y Daniel Billsus. Content-based recommendation systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 10, pages 325–341. 2007.
- [86] David M. Pennock, Eric Horvitz, Steve Lawrence, y C. Lee Giles. Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, UAI'00, pages 473–480, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [87] Alexandrin Popescul, Lyle H. Ungar, David M. Pennock, y Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, pages 437–444, 2001.
- [88] C. Porcel y E. Herrera-Viedma. Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. *Knowledge-Based Systems*, 23(1):32–39, February 2010.
- [89] Pearl Pu y Li Chen. Trust building with explanation interfaces. In *Proceeding of the 11th international conference on Intelligent User Interfaces*, IUI '06, pages 93–100. Association for Computing Machinery Press, 2006.
- [90] Pearl Pu y Li Chen. Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, 20(6):542–556, 2007.
- [91] J. R. Quinlan. Induction of decision trees. *Machine Learning Research*, 1(1):81–106, March 1986.
- [92] J.R. Quinlan. Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348, Singapore, 1992. World Scientific.

- [93] M. Elena Renda y Umberto Straccia. A personalized collaborative digital library environment: a model and an application. *Information Processing and Management*, 41(1):5–21, January 2005.
- [94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, y John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. Association for Computing Machinery.
- [95] Paul Resnick, Neophytos Iacovou, Mitesh Sushak, Peter Bergstrom, y John Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *1994 ACM Conference on Computer Supported Collaborative Work Conference*, pages 175–186, Chapel Hill, NC, 10 1994. Association of Computing Machinery, Association of Computing Machinery.
- [96] Paul Resnick y Hal R. Varian. Recommender systems. *Communications of the ACM*, 40:56–58, March 1997.
- [97] Francesco Ricci, Lior Rokach, Bracha Shapira, y Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
- [98] Walter E. Riggs. The delphi technique: an experimental evaluation. *Technological Forecasting and Social Change*, pages 89–94, 1983.
- [99] Ekkawut Rojsattarat y Nuanwan Soonthornphisaj. Hybrid recommendation: Combining content-based prediction and collaborative filtering. In Jiming Liu, Yiu ming Cheung, y Hujun Yin, editors, *Intelligent Data Engineering and Automated Learning, 4th International Conference, IDEAL 2003, Hong Kong, China, March 21-23, 2003, Revised Papers*, volume 2690 of *Lecture Notes in Computer Science*, pages 337–344. Springer, 2003.
- [100] Alexandra Roshchina, John Cardiff, y Paolo Rosso. Evaluating the similarity estimator component of the twin personality-based recommender system. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ug(ur Dog(an, Bente Maegaard, Joseph Mariani, Jan Odiijk, y Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).

- [101] Hidekazu Sakagami y Tomonari Kamba. Learning personal preferences on online newspaper articles from user behaviors. In *Selected papers from the sixth international conference on World Wide Web*, WWW'97, pages 1447–1455, 1997.
- [102] B. M. Sarwar, G. Karypis, J. Konstan, y J. Riedl. Incremental svd-based algorithms for highly scaleable recommender systems. In *Fifth International Conference on Computer and Information Technology*, ICCIT, 2002.
- [103] Badrul Sarwar, George Karypis, Joseph Konstan, y John Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. Association for Computing Machinery.
- [104] Badrul M. Sarwar, George Karypis, Joseph Konstan, y John Reidl. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT)*, 2002.
- [105] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, y John T. Riedl. Application of dimensionality reduction in recommender system – a case study. In *In Association for Computing Machinery WEBKDD Workshop*, 2000.
- [106] Badrul M. Sarwar, Joseph A. Konstan, Al Borchers, Jon Herlocker, Brad Miller, y John Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 345–354. ACM Association for Computing Machinery Press, 1998.
- [107] J. Ben Schafer, Dan Frankowski, Jon Herlocker, y Shilad Sen. Collaborative filtering recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter 9, pages 291–324. 2007.
- [108] J.B. Schafer. The application of data-mining to recommender systems. *J. Wang (Ed.), Encyclopedia of data warehousing and mining (pp. 44-48). Hershey, PA: Idea Group Reference, 1:44–48, 2005.*
- [109] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, y David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, Tampere, Finland, 2002.

- [110] Upendra Shardanand y Pattie Maes. Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '95, pages 210–217, New York, NY, USA, 1995. Association for Computing Machinery Press/Addison-Wesley Publishing Co.
- [111] Barry Smyth. Case-based recommendation. In *The Adaptive Web*, pages 342–376, 2007.
- [112] Panagiotis Symeonidis, Alexandros Nanopoulos, y Yannis Manolopoulos. Movieexplain: a recommender system with explanations. In *Recommender Systems, RecSys*, pages 317–320, 2009.
- [113] Sergios Theodoridis y Konstantinos Koutroumbas. *Pattern recognition*. Academic Press, 1999.
- [114] Sergios Theodoridis y Konstantinos Koutroumbas. *Pattern Recognition, Third Edition*. Academic Press, Inc., Orlando, FL, USA, 2006.
- [115] Nava Tintarev. Explanations of recommendations. In *Recommender Systems, RecSys*, pages 203–206, 2007.
- [116] Nava Tintarev y Judith Masthoff. A survey of explanations in recommender systems. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 801–810, Washington, DC, USA, 2007. IEEE Computer Society.
- [117] Nava Tintarev y Judith Masthoff. Designing and evaluating explanations for recommender systems. In *Recommender Systems Handbook*, pages 479–510. 2011.
- [118] Nava Tintarev y Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems - methodological issues and empirical studies on the impact of personalization. *User Modeling and User-Adapted Interaction*, 22(4-5):399–439, 2012.
- [119] Manos Tsagkias y Roi Blanco. Language intent models for inferring user browsing behavior. In *Special Interest Group on Information Retrieval, SIGIR*, pages 335–344, 2012.
- [120] L. Ungar y D. Foster. Clustering methods for collaborative filtering. In *Proceedings of the Workshop on Recommendation Systems*. Association for the Advancement of Artificial Intelligence, AAAI Press, Menlo Park California, 1998.

- [121] Y. Wang y I. H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning*. Springer, 1997.
- [122] C.J. Willmott, S.G. Ackleson, R.E. Davis, J.J. Feddema, K.M. Klink, D.R. Legates, J. O'Donnell, y C.M. Rowe. Statistics for the evaluation and comparison of models. *Journal of geophysical Research*, 90(C5):8995–9005, 1985.
- [123] Rui Xu y II Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.
- [124] Kazuyoshi Yoshii, Student Member, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno, y Senior Member. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *Audio, Speech, and Language Processing, IEEE*, 16:435–447, February 2008.
- [125] Cong Yu, Laks V. S. Lakshmanan, y Sihem Amer-Yahia. Recommendation diversification using explanations. In *International Conference on Data Engineering, ICDE*, pages 1299–1302, 2009.
- [126] Markus Zanker y Daniel Ninaus. Knowledgeable explanations for recommender systems. In Jimmy Xiangji Huang, Irwin King, Vijay V. Raghavan, y Stefan Rueger, editors, *Web Intelligence*, pages 657–660. IEEE, 2010.
- [127] DeJia Zhang. An item-based collaborative filtering recommendation algorithm using slope one scheme smoothing. In *Proceedings of the 2009 Second International Symposium on Electronic Commerce and Security - Volume 02*, ISECS '09, pages 215–217, Washington, DC, USA, 2009. IEEE Computer Society.