

USANDO INFORMACIÓN DE SEGUNDA MANO EN UN SISTEMA DE RECOMENDACIÓN COLABORATIVO

L.M. de CAMPOS¹ J.M. FERNÁNDEZ-LUNA¹ J.F. HUETE¹ M.A. RUEDA-MORALES¹

¹ Dpto. de Ciencias de la Computación e I.A., E.T.S.I. Informática, Universidad de Granada, 18071 – España
{lci,jmfluna,jhg,mrueda}@decsai.ugr.es

Resumen

La construcción de Sistemas de Recomendación Colaborativos ha recibido una considerable atención en los últimos años. El problema de estos sistemas se presenta en aquellos productos para los que tenemos poca información, ya que las predicciones suelen ser erróneas. En este trabajo presentamos una idea novedosa con la que vamos a obtener información de calidad a partir de las Bases de Datos, y que nos va a permitir mejorar las predicciones del sistema. Para demostrar su valía, vamos a usar un sistema probabilístico que presentamos en este trabajo, y otro no probabilístico ajeno a nosotros.

Palabras Clave: Sistemas de Recomendación Colaborativos, Redes Bayesianas, vecindario, información de segunda mano.

- Basados en contenido: Se recomiendan productos similares a aquellos que el usuario activo ha votado positivamente en el pasado.
- Colaborativos: Se recomiendan productos que otros usuarios con gustos similares al usuario activo consideran buenos.
- Híbridos: La recomendación se realiza combinando las aproximaciones colaborativa y basada en contenido.

En este artículo nos centraremos en la variante colaborativa. Uno de los problemas de este tipo de sistemas surge cuando el usuario del sistema quiere información sobre un producto determinado y las personas con gustos similares a él no se la pueden ofrecer (no han visto la película, no han cenado en el restaurante...). En este caso el sistema ofrecerá una valoración del producto para el usuario que, seguramente, no va a ser de calidad. Una posible solución a este problema es utilizar información, que podríamos llamar de segunda mano, a la hora de recomendar. Por ejemplo, supongamos la siguiente situación: Yo pregunto a mis amigos por una determinada película pero ninguno, o pocos, de éstos la han visto. Sin embargo, ocurre que éstos tienen amigos, con gustos parecidos a ellos, que sí que la han visto. Parece obvio pensar que, si esos amigos le dicen (ya que tienen sus mismos gustos) si le va a gustar o no, él podría decirme a mí (ya que tenemos los gustos parecidos) si me va a gustar o no. Este es el objetivo de nuestro trabajo: obtener nueva información, y de calidad, para mejorar las recomendaciones del sistema.

El resto del artículo se divide de la siguiente forma: Después de una breve introducción a los SR Colaborativos, en la Sección 2, describiremos los modelos con los que vamos a experimentar en este trabajo. En la sección 3 expondremos cómo vamos a obtener la información de segunda mano. La sección 4 presenta los resultados obtenidos mediante la ejecución de distin-

1 INTRODUCCIÓN

Tomar decisiones sobre situaciones como ir a cenar a un restaurante, ver una película o planear unas vacaciones pueden ser tareas bastante complicadas por una razón principal: la gran cantidad de opciones de restaurantes, películas y destinos vacacionales que existen. Los Sistemas de Recomendación (SR) surgen para intentar paliar las dificultades de tratar con tantas opciones. En términos generales, los SR producen sugerencias (recomendaciones) sobre productos (o acciones) dentro de un determinado dominio en el cuál está interesado el usuario. En concreto, en este trabajo nos centraremos en películas como productos susceptibles de ser recomendados.

Hay muchos tipos de SR [11, 4] dependiendo de la información que se use para realizar la recomendación:

tos experimentos y, para finalizar, en la sección 5 se ofrecen algunas conclusiones y posible trabajo futuro.

2 SISTEMAS DE RECOMENDACIÓN COLABORATIVOS

Podemos encontrar distintos tipos de SR dependiendo de los recursos utilizados para determinar el vecindario, así como para determinar el voto propuesto por el sistema. En [8] se hace un análisis de distintas variantes de algoritmos colaborativos basados en vecindario tales como similitud vector-coseno, correlación de Pearson y métodos estadísticos Bayesianos. En [12, 10], se usan algoritmos de aprendizaje sobre Redes Bayesianas para aprender las distribuciones de probabilidad.

En este trabajos nos centraremos en dos de ellos: El primero basado en vecindario y el segundo, un modelo probabilístico basado en Redes Bayesianas.

2.1 Promedio de desviaciones de la media del vecindario[2]

Partiendo del modelo propuesto por Grouplens [3], Promedio de desviaciones de la media del vecindario (Weighted Average of Deviations from the neighbor's mean) es un SR Colaborativo que usa un algoritmo basado en el vecindario:

$$voto_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) \cdot sim_{a,u}}{\sum_{u=1}^n sim_{a,u}}$$

donde $voto_{a,i}$ es la predicción de voto para el usuario activo a del item i , n es el número de usuarios similares (tamaño del vecindario) y $sim_{a,u}$ es la medida de similitud entre el usuario activo y el usuario similar u .

Para medir la similitud entre usuarios, usado también como la base de pesos en diferentes sistemas colaborativos, usa el Coeficiente de Correlación de Pearson (CCP): Sea U el conjunto de usuarios y U_a y U_b dos usuarios de dicho conjunto, el CCP para ellos se calcula de la forma:

$$CCP(U_a, U_b) = \frac{\sum_j (r_{a,j} - \bar{r}_a)(r_{b,j} - \bar{r}_b)}{\sqrt{\sum_j (r_{a,j} - \bar{r}_a)^2 \sum_j (r_{b,j} - \bar{r}_b)^2}} \quad (1)$$

donde las sumas sobre j se hacen sobre aquellos items para los cuales los usuarios U_a y U_b tienen votos, $I(U_a) \cap I(U_b)$ (donde $I(U)$ es el conjunto de items votados por el usuario U en el conjunto de datos). Si no hay items comunes en el historial de voto de U_a y U_b ,

entonces $CCP(U_a, U_b) = 0$ por defecto. Además, \bar{r}_a es el voto medio para el usuario U_a , esto es,

$$\bar{r}_a = \frac{1}{|I(U_a)|} \sum_{I_k \in I(U_a)} r_{a,k}$$

El valor final de similitud se obtendrá aplicando un factor de corrección que devalue aquellos valores de CCP que se hayan obtenido con menos de 50 películas en común, esto es:

$$sim(U_a, U_b) = CCP(U_a, U_b) \cdot FC$$

con

$$FC = \begin{cases} 1 & \text{si } n > 50 \\ \frac{n}{50} & \text{en otro caso} \end{cases} \quad (2)$$

siendo n el número de películas en común.

Para finalizar la exposición del modelo, decir que, para cada predicción de voto sobre un item, se utilizan solo los usuarios similares que han visto dicho item, esto es, para cada usuario, su vecindario varía dependiendo del item que se evalúe.

2.2 SR Colaborativo mediante Redes Bayesianas

En esta sección se presenta cómo se pueden modelar las relaciones entre usuarios (y su uso a la hora de predecir el voto) mediante Redes Bayesianas (RB): Sea U un conjunto de nodos en la RB representando al conjunto de usuarios, cada variable usuario U_a representa la distribución de probabilidad asociada a su patrón de voto. Si asumimos que los votos son números enteros en una escala del 1 hasta $\#r$, los estados de cada nodo corresponden con el conjunto $\{1, 2, \dots, \#r\}$.

Para facilitar la presencia de relaciones de dependencia entre individuos en el modelo (evitando una topología de red compleja), proponemos el uso de un nuevo conjunto de nodos V que identifiquen los votos colaborativos. Hay un nodo colaborativo por cada usuario del sistema, esto es, $V = V_1, V_2, \dots, V_n$. Estos nodos se usarán para estimar las distribuciones de probabilidad de los votos de usuario y, por lo tanto, tomarán sus valores del mismo dominio de voto que U .

2.2.1 Aprendizaje

El conjunto de padres de una variable V_a en el grafo, $Pa(V_a)$, se aprenderá de la base de datos de votos y contendrá aquellas variables usuario, $U_b \in U$, donde U_a y U_b deben tener una gran similitud entre sus gustos. Así, dado un valor de similitud, el conjunto

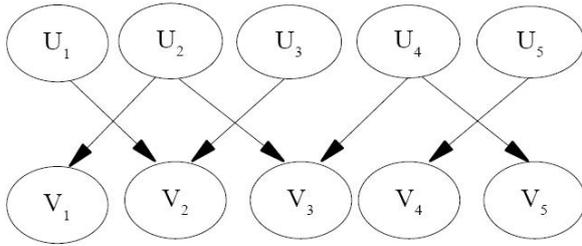


Figura 1: Topología del Sistema de Recomendación

$Pa(V_a)$ puede ser obtenido usando un umbral o considerando solo las primeras p variables en el ranking. Para medir la similitud entre usuarios, proponemos una combinación de dos criterios diferentes pero complementarios: correlación de voto entre items comunes (ver ecuación 1) y grado de solapamiento, esto es:

$$sim(U_a, U_b) = abs(CCP(U_a, U_b)) \cdot D(U_a, U_b)$$

Notar que con el uso del valor absoluto para CCP estamos diciendo que tanto los usuarios similares como los usuarios con gustos opuestos (complementarios) pueden ayudar en la predicción del voto final para un usuario activo. Esta es una de las diferencias con respecto al primer modelo, ya que allí se obtienen los usuarios similares a partir de aquellos que tienen una correlación positiva ($CCP > 0$).

El segundo criterio trata de penalizar aquellos vecinos altamente correlados que están basados en muy pocos items en común. Por consiguiente, tenemos que tener en cuenta el número de items que tanto U_a como U_b han votado en común, esto es, su grado de solapamiento. En esta situación, consideramos que cuanto más grande sea la probabilidad de que un usuario U_a vote un item que ha sido votado por U_b , mayor será la calidad de U_b como padre de U_a ¹. Este criterio se define mediante la siguiente expresión:

$$D(U_a, U_b) = \frac{|I(U_a) \cap I(U_b)|}{|I(U_b)|}$$

2.2.2 Estimando las distribuciones de probabilidad condicionadas

Para completar la especificación del modelo, tenemos que fijar los valores numéricos para las probabilidades condicionadas. Previamente, vamos a introducir alguna notación. Así, dada una variable X_i , usamos letras minúsculas para denotar la realización de la variable, esto es, $x_{i,j}$ significa que la variable X_i toma

¹Notar que hemos usado la penalización propuesta en la Ecuación 2, obteniendo peores resultados que usando esta propuesta.

el j^{th} valor. Escribimos $Pr(x_{i,j}|pa(X_i))$ para $P(X_i = x_{i,j}|pa(X_i))$, siendo $pa(X_i)$ una configuración del conjunto de padres de X_i , $Pa(X_i)$, o, a veces, $Pr(X)$ para denotar la distribución de probabilidad.

Debemos distinguir entre nodos de usuario, U , y nodos colaborativos, V . Los primeros, al ser nodos raíz en el grafo, almacenan distribuciones de probabilidad marginadas mientras que los segundos, almacenan un conjunto de distribuciones de probabilidad condicionadas. Se definen de la forma:

- Para cada nodo usuario U_k necesitamos fijar la distribución de probabilidad a priori sobre su patrón de voto. En este trabajo proponemos éstas sean equiprobables, esto es, que $Pr(U_{k,s} = 0.2)$ si $s \in \{1, 2, \dots, 5\}$.
- Para un nodo dado V_i debemos definir para cada configuración $pa(V_i)$ la distribución de probabilidad condicionada $Pr(V_{i,j}|pa(V_i))$. En estos casos, puede ser bastante complicado fijar y almacenar las tablas de probabilidades condicionadas (con un tamaño exponencial al número de padres). Por ello, proponemos el uso de un modelo canónico similar al presentado en [1], para representar las probabilidades condicionadas, que nos va a permitir diseñar un procedimiento de inferencia muy eficiente. Así, para un nodo X_i , definimos esas probabilidades de la forma:

$$Pr(x_{i,j}|pa(X_i)) = \sum_{Y_k \in Pa(X_i)} w(y_{k,l}, x_{i,j}) \quad (3)$$

donde $y_{k,l}$ es el valor que la variable Y_k toma en la configuración $pa(X_i)$ y $w(y_{k,l}, x_{i,j})$ son pesos que miden cómo el l^{th} valor de la variable Y_k describe el j^{th} estado del nodo X_i , con $w(y_{k,\bullet}, x_{i,j}) \geq 0$ y $\sum_{Y_k \in Pa(X_i)} w(y_{k,\bullet}, x_{i,j}) \leq 1$.

El problema ahora viene en estimar los pesos dados por aquellos usuarios U_b con gustos similares, esto es, $U_b \in Pa(V_a)$. Al no conocer esos pesos, tenemos que estimarlos de la base de datos de entrenamiento. Siguiendo las ideas de [1], consideramos que $w(u_{b,t}, v_{a,s})$, esto es, el efecto que el voto t dado por el usuario U_b tiene en la predicción del voto de V_a se calcula por medio de:

$$w(u_{b,t}, v_{a,s}) = \frac{N(u_{b,t}, v_{a,s}) \cdot sim(U_a, U_b)}{N(u_{b,t}) \cdot \sum_{U_k \in Pa(V_a)} sim(U_a, U_k)}$$

con $1 \leq t, s \leq r$, siendo $N(u_{b,t}, v_{a,s})$ el número de items del conjunto $I(U_a) \cap I(U_b)$ que han sido votados con el valor t por el usuario U_b y que también han sido votados con el valor s por el usuario U_a , $N(u_{b,t})$

el número de items del conjunto $I(U_a) \cap I(U_b)$ que han sido votados con el valor t por el usuario U_b y $sim(U_a, U_b)$ el valor de similitud entre el usuario U_a y U_b .

2.2.3 Inferencia

Dado un usuario activo U_a y una película I_k en el sistema (votada por otros usuarios), el problema está en predecir el voto que el usuario U_a dará a la película. Desde un punto de vista probabilístico, tenemos que calcular² $Pr(v_{a,s}|I_k)$, $1 \leq s \leq \#r$. En primer lugar vamos a ver cómo incluir la evidencia, I_k , en la RB. Dada la película, tendremos que cambiar nuestra certeza en las distribuciones de voto para cualquier usuario que haya votado previamente a la película (nodos U en nuestro modelo), esto es,

$$Pr(U_i = s|I_k) = \begin{cases} 1 & \text{si } U_i \text{ votó } s, \\ 0 & \text{si } U_i \text{ votó } k \neq s. \end{cases}$$

Cuando el usuario no ha visto la película, la distribución de probabilidad a posteriori se corresponde con la a priori.

Una vez la evidencia ha sido incluida en el sistema, el problema es predecir el voto del usuario para dicho item I_k . Esos valores de probabilidad se calculan propagando la evidencia a través de la red.

Usando las ventajas de los modelos canónicos usados para expresar las distribuciones de probabilidad condicionada (ecuación 3), estas probabilidades finales pueden calcularse eficientemente [1] de la forma:

$$Pr(v_{a,s}|ev) = \sum_{j=1}^{m_{V_a}} \sum_{k=1}^{l_{U_j}} w(u_{j,k}, v_{a,s}) \cdot Pr(u_{j,k}|ev).$$

siendo m_{V_a} el número de padres de V_a , U_j un nodo en $Pa(V_a)$ y l_{U_j} el número de estados que toma U_j .

2.2.4 Obtención del voto

Una vez calculada la distribución de probabilidad a posteriori $Pr(V_{a,s}|ev)$, debemos decidir el voto que el sistema recomienda al usuario. En este trabajo estudiamos dos alternativas para la obtención del voto:

MP : Seleccionamos el voto con mayor probabilidad a posteriori:

²Con la misma filosofía, el sistema podría ser usado para recomendar a un usuario U_a aquellas películas que tienen una gran probabilidad de que le gusten. Esto podría hacerse instanciando el conjunto de películas no vistas por el usuario U_a .

$$\text{voto} = \arg_s \max\{Pr(V_a = s|ev)\}$$

DP : La idea es medir sólo la porción de evidencia que cada voto posible recibe gracias a la instanciación del item. Esta porción de evidencia puede ser definida fácilmente considerando la diferencia entre los valores de probabilidad a priori (sin evidencia) y a posteriori, y devolviendo el voto con el que obtenemos la máxima diferencia:

$$\text{voto} = \arg_s \max\{Pr(V_a = s|ev) - Pr(V_a = s)\}$$

3 OBTENCIÓN DE INFORMACIÓN DE SEGUNDA MANO

Los SR Colaborativos identifican grupos de personas con gustos similares al usuario activo y le recomienda aquellos productos que les hayan gustado. El problema surge cuando el usuario del sistema quiere información sobre un producto determinado y las personas con gustos similares a él no se la pueden ofrecer. En este caso el sistema ofrecerá una valoración del producto para el usuario que, seguramente, no va a ser de calidad. Ahora bien, ¿qué pasaría si la información de sus vecinos podemos obtenerla a partir de sus propios usuarios similares? O dicho de otra forma, ¿mejoraría la predicción de voto de un usuario a una película si hacemos uso de información de segunda mano?

Para probar nuestra idea modificamos la forma en la que los modelos obtienen el voto:

- Para el modelo promedio (sección 2.1):

$$\text{voto}_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (\hat{r}_{u,i} - \bar{r}_u) \cdot \text{sim}_{a,u}}{\sum_{u=1}^n \text{sim}_{a,u}}$$

donde $\hat{r}_{u,i}$ es el voto del usuario si ha visto la película o , en caso contrario, el voto predicho por el modelo promedio si no la ha visto, esto es $\hat{r}_{u,i} = \text{voto}_{u,i}$.

- Para el modelo probabilístico (sección 2.2) modificamos la distribución de probabilidad a posteriori de los posibles votos de la forma:

$$Pr(v_{a,s}|ev) = \sum_{j=1}^{m_{V_a}} \sum_{k=1}^{l_{U_j}} w(u_{j,k}, v_{a,s}) \cdot \hat{Pr}(u_{j,k}|ev).$$

siendo

$$\hat{Pr}(U_i = s|I_k) = \begin{cases} 1 & \text{si } U_i \leftarrow s, \\ 0 & \text{si } U_i \leftarrow k \neq s. \end{cases}$$

donde $U_i \leftarrow s$ representa que, o bien el usuario U_i votó a I_k con el valor s , o bien, que el sistema probabilístico recomendó el voto s para el usuario utilizando los criterios de la sección 2.2.4.

4 ANÁLISIS EMPÍRICO

Esta sección presenta los resultados experimentales obtenidos mediante la prueba del sistema. Los conjuntos de datos utilizados en nuestros experimentos se han obtenido de la Base de Datos Movielens³, que contiene 1682 películas y 943 usuarios que aportan su voto a las películas que han visto, dando lugar a 100000 votos en una escala del 1 al 5. Para las ejecuciones hemos usado los conjuntos de datos U1.base y U1.test hasta U5.base y U5.test que dividen la colección en 80% para entrenamiento y 20% para test respectivamente.

Para medir la capacidad de recomendación usamos el error medio absoluto (MAE) [9] que mide la desviación media absoluta entre el voto real y el voto predicho. Así mismo, hemos establecido el tamaño del vecindario en 30, esto es, para cada usuario hemos calculado los 30 usuarios más cercanos a sus gustos con los que realizar las predicciones.

La experimentación se ha realizado evaluando los dos modelos presentados en este trabajo: *modelo probabilístico*⁴ (P) (sección 2.2) y *promedio de desviaciones de la media del vecindario* (A) (sección 2.1).

Los resultados obtenidos lanzando los sistemas con los conjuntos de datos originales obtenidos por ambos modelos se pueden ver en la tabla 1. Las ejecuciones con estos datos podemos considerarlas como *Baseline*(B). Podemos observar cómo el modelo promedio (BA) obtiene los mejores resultados y, en el modelo probabilístico (BP), los mejores resultados los obtenemos usando el criterio DP.

Tabla 1: Resultados Baseline.

	U_1	U_2	U_3	U_4	U_5
BP+MP	0.845	0.816	0.812	0.814	0.822
BP+DP	0.817	0.792	0.797	0.794	0.812
BA	0.745	0.737	0.731	0.723	0.734

La idea de este trabajo es la de usar información de segunda mano para mejorar la predicción de los sistemas colaborativos. Sin embargo, no todos los votos que predice el sistema tienen el mismo grado de utilidad. Por ejemplo, el valor que puede tener un voto obtenido a partir de 15 vecinos no puede ser el mismo que el obtenido a partir de uno solo. En este último caso, probablemente estamos introduciendo ruido en las predicciones. Es por esto que pretendemos introducir sólo información de calidad. Una alternativa

³<http://www.movielens.org>

⁴Notar que para el modelo probabilístico, hemos añadido MP o DP a la identificación del experimento dependiendo de cómo se obtenga el voto (ver sección 2.2.4).

para definir la calidad, es el estudiar el acierto del sistema calculando sólo los votos en los que el número de padres con el que se obtiene sea mayor que un cierto umbral. A la vista de los resultados de la Tabla 2, hemos considerado votos de calidad aquellos que se pueden obtener usando la información proporcionada por al menos el 40% de su vecindario ya que obtienen unos buenos resultados de MAE junto con una cobertura razonable (siendo la cobertura el porcentaje de datos que evaluamos con ese umbral).

Tabla 2: Calidad para U1.

% Padres	16	26	33	40
MAE_P	0.796	0.766	0.750	0.734
MAE_A	0.699	0.685	0.676	0.670
$Cobertura_P(\%)$	83.99	70.56	60.88	51.17
$Cobertura_A(\%)$	79.07	63.52	53.42	44.22

Para poder determinar el comportamiento del sistema y poder medir la ganancia que obtenemos mediante la inclusión de los nuevos votos, hemos reducido el conjunto de votos reales que se utilizan al recomendar. Así, para calcular el voto del usuario activo sobre una película, hemos eliminado aleatoriamente el 50% de los votos de calidad de sus padres. Las ejecuciones con estos datos podemos considerarlas como *Reducida* (R). En la Tabla 3 podemos ver el número de votos eliminados para cada subconjunto de datos ($E_{P,A}$). Hemos de tener en cuenta que cada vez que quitamos un voto, éste deja de utilizarse en todas las predicciones en las que entraba a formar parte. En la Tabla 4 podemos ver cómo varían los tamaños de los conjuntos de padres con los que realizamos las predicciones una vez eliminados los votos. Los resultados que proporciona el sistema los podemos ver en la Tabla 5 y como es de esperar, los valores de acierto empeoran con respecto a baseline para todas las ejecuciones de ambos modelos.

Tabla 3: Votos eliminados e Insertados.

	U_1	U_2	U_3	U_4	U_5
$E_{P,A}$	24114	23954	24001	24218	24347
I_P	45967	42523	44072	45688	47890
I_A	33487	34695	35716	36690	36819

Tabla 4: Evolución Vecindario en U1.

Pa	≤ 5	6 - 12	13 - 18	19 - 24	≥ 25
BP	4022	6570	5130	3512	766
RP	9618	9107	1254	21	0
BA	5203	6810	4750	2762	475
RA	10796	8214	984	6	0

Tabla 5: Resultados Reducido.

	U_1	U_2	U_3	U_4	U_5
RP+MP	0.872	0.838	0.834	0.834	0.842
RP+DP	0.866	0.863	0.844	0.835	0.856
RA	0.784	0.812	0.775	0.771	0.777

Finalmente, la Tabla 6 presenta los resultados obtenidos cuando usamos información de segunda mano (ver Sección 3) (C), en la cuál introducimos todos los votos de calidad que ambos modelos predicen. Podemos ver en número de votos de calidad que introducimos en cada conjunto de datos en la Tabla 3 (I_P y I_A). Como podemos observar, los resultados son prometedores para ambos modelos, ya que el añadir los votos de calidad propuestos por ambos sistemas hace que sus valores de MAE rocen el valor del baseline y, en algunos casos, lo superen, lo cuál certifica la valía de nuestra idea.

Tabla 6: Resultados con votos de calidad.

	U_1	U_2	U_3	U_4	U_5
CP+MP	0.846	0.818	0.812	0.815	0.821
CP+DP	0.818	0.794	0.787	0.790	0.805
CA	0.741	0.738	0.732	0.725	0.733

5 CONCLUSIONES

Hemos presentado una idea novedosa para aplicar a los SR Colaborativos de tal forma que podemos obtener información indirecta de las Bases de Datos de votos haciendo uso de las predicciones que hace el propio sistema. Para ello hemos tenido que disminuir el número de votos de dichas Bases de Datos ya que, tras distintas experimentaciones, hemos comprobado que en aquellos usuarios en los que sus usuarios similares no le dan información, no es posible obtenerla indirectamente. Por este motivo, y como trabajo futuro, nos planteamos utilizar distintas Bases de Datos como NextFlix para probar nuestra propuesta, con el fin de poder mejorar las predicciones en conjunto del sistema a partir de los datos en su totalidad. Además, haremos un estudio más exhaustivo del conjunto de padres de los usuarios, dividiéndolos en intervalos y estudiando el comportamiento del sistema en cada uno de ellos.

Agradecimientos

Trabajo respaldado por el Ministerio de Educación y Ciencia y la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía bajo los proyectos TIN2005-02516 y TIC-276, respectivamente.

Referencias

- [1] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete. A collaborative recommender system based on probabilistic inference from fuzzy observations. *Fuzzy Sets and Systems*, doi: 10.1016/j.fss.2008.01.016, 2008.
- [2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. *SIGIR.*, ACM 1-58113-096-1/99/0007, 1999.
- [3] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. T. Riedl. Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77-87, 1997.
- [4] S. Kangas. Collaborative filtering and recommendation systems. *VTT Information Technology*, Research Report TTE4-2001-35, 2002.
- [5] V. Robles, P. Larrañaga, J.M. Peña, O. Marbán, J. Crespo, and M.S. Pérez. Collaborative filtering using interval estimation naive Bayes. *Lecture Notes in Artificial Intelligence*, 2663:46–53, 2003.
- [6] K. Miyahara and J. Pazzani. Collaborative filtering with the simple Bayesian classifier. *Proc. of the Pacific Rim Int. Conf. on Artif. Intell.*, 679–689, 2000.
- [7] D. Heckerman, C. Chickering, D. Meek, C. Rounthwaite and C. Kadie. Dependency networks for inference, collaborative filtering and data visualization. *Jour. Mach. Learning Res.* 1, pp. 49-75, 2000.
- [8] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proc. 14th Conference on Uncertainty in Artificial Intelligence*, 43–52, 1998.
- [9] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5-53, 2004.
- [10] C.J. Butz. Exploiting contextual independencies in web search and user profiling. *Proc. of World Congress on Computational Intelligence*, 1051–1056, 2002.
- [11] P. Resnick and H.R. Varian. Recommender Systems. *Communications of the ACM*, 40(3):56-58, 1997.
- [12] S.N. Schiaffino and A. Amandi. User profiling with case-based reasoning and Bayesian network. *Proc. of the Iberoamerican Conf. of Artificial Intelligence*, 12–21, 2000.