ISSN 1870-4069

Estado del arte en los sistemas de recomendación

Oscar Escamilla González, Sergio Marcellin Jacques

Universidad Nacional Autónoma de México,
Posgrado en Ciencias e Ingeniería en Computación, Ciudad de México,
México

{oescamil, smarcellin}@gmail.com

Resumen. Los sistemas de recomendación (***SR***s) son sistemas automatizados cuyo propósito es el filtrado de información. Los SRs están pensados para apoyar a los usuarios a encontrar *ítems* dentro de un contexto determinado como por ejemplo, películas, libros, artículos académicos, etc. El objetivo principal de los SRs es mostrar de los ítems *recomendables*, aquellos que se *estime*, sean de interés para el usuario. Dicho de otra manera, los SRs estiman el interés que el usuario podría tener sobre los ítems que este aún no ha considerado. Con las estimaciones hechas, el SRs genera un ranking de ítems y le muestra al usuario los ítems que se encuentran en las posiciones más altas de éste. El objetivo de este trabajo es presentar el "*estado del arte*" de los SRs exponiendo distintas técnicas con las que se ataca el problema de estimar la relevancia de los ítems para un usuario determinado.

Palabras clave: sistemas de recomendación, filtrado colaborativo basado en confianza, basado en contenido, filtrado colaborativo, basado en modelos, sistemas de recomendación híbridos.

State of the Art of Recommendation Systems

Abstract. Recommender Systems (RS) are automatic systems which purpose is filtering information. The RS are designed to help users to find *items* within a specific context, for example: movies, books, research articles, etc. Their main objective is to show users *recommendable items* those items that RS estimates will be of interest for the user. In other words, the RS estimate the posible interest a user may have on items that have not been considered by the user. Then, with the estimations made, the RS creates a ranking of items end shows the best ranked items to the user. The main purpose of this job is to present the "*state of the art*" of the RS, exposing different techniques used to tackle the problem of estimating the relevance of items for a certain user.

Keywords: recommender systems, thrust collaborative filtering, content base filtering, model based recommender system, collaborative filtering, hibrid recommender system.

1. Introducción

Actualmente la cantidad de información a la que podemos acceder por medio de la Internet excede la capacidad de una persona para poder procesarla en términos de lo que un usuario necesita encontrar. Por ejemplo, supongamos que tenemos un usuario que es un estudiante que está buscando artículos académicos relacionados con su tema de tesis. La cantidad de artículos que el usuario puede encontrar relacionados es exorbitante [12]. Aun así, por lo general sólo unos pocos artículos de este conjunto son relevantes o están acordes a las necesidades de su investigación.

Otro ejemplo típico es la adquisición de productos o servicios por medio de la Internet. La cantidad de posibles resultados al hacer una búsqueda es colosal, de tal manera que puede que los resultados que se obtengan sean servicios o productos que en realidad no se ajustan del todo a las necesidades, o en la multitud de resultados se pasen por alto algunas opciones que pudieran ser de interés. Este no es sólo un problema para los usuarios si no también para los proveedores, ya que éstos pueden perder oportunidades de ventas por el simple hecho de que el usuario no encuentre el producto que el proveedor ofrece en la multitud de opciones, a pesar de que éste sea de interés para el usuario.

Por otro lado, ante la falta de conocimiento de las alternativas o la inexperiencia, de manera natural nos apoyamos de las recomendaciones de terceros para la toma de decisiones. Pero muchas veces esta información es subjetiva o incluso puede resultar complicado encontrarla.

Por estas razones se ha vuelto necesario la construcción de herramientas automáticas o semiautomáticas que provean algún tipo de recomendación, ayudando a los usuarios a encontrar información, productos, servicios, etc. de mejor manera, filtrando la información del universo disponible logrando así un mejor uso de ella. Además, desde el punto de vista de los proveedores, se tiene un interés creciente por el desarrollo de este tipo herramientas, las cuales faciliten la vinculación entre sus productos y los usuarios que los necesitan e incluso aumenten las oportunidades de venta al recomendar productos adecuados al perfil de compra del usuario. Uno de los ejemplos más famosos de este tipo de proveedores es la compañía *Netflix*, que en 2007 [23] organizó un concurso ofreciendo un premio de 1,000,000 USD a quien pudiera implementar un sistema mejorara las predicciones en al menos en un 10 % con respecto al sistema de *Netflix*. El estudio de los SRs es relativamente nuevo y se independizó como un campo de investigación a mediados de los noventas y desde el 2007 se ha incrementado el interés por estos [10]. Como muestra de esto, los SRs son parte fundamental en importantes sitios Web entre los que destacan: Amazon, YouTube, Netflix, Yahoo, Tripadvisor, Last.fm, e IMDb.

2. Sistemas de recomendación

Los sistemas de recomendación (***SR***s) están muy relacionados con sistemas de "búsqueda o recuperación de información", dado que ambos están pensados

para que a partir de un conjunto de datos se obtenga información relevante para el usuario [2]. Pero a su vez tienen diferencias fundamentales. Por ejemplo, en los sistemas de búsqueda se espera un uso puntual de éstos, mientras que los SRs se enfocan en usos repetidos a lo largo de tiempo. Otra diferencia es que en los sistemas de búsqueda los criterios de filtrado son expresados por el usuario de manera explícita cada vez que interactúa con él, mientras que en los SRs, estos criterios se obtienen de forma implícita del *perfil del usuario*. El hecho de que se necesite un perfil del usuario para realizar el filtrado, implica que se debe obtener y almacenar este perfil en alguna parte y esta necesidad, es otra de las diferencias con los sistemas de búsqueda. En los sistemas de búsqueda el usuario puede interactuar de forma anónima mientras que en los SR, el usuario necesita tener un perfil asociado.

Además tienen relación con la minería de datos ya que los SRs se pueden ver como un problema de estimar datos perdidos. De los ítems que el usuario aún no ha contemplado (*y por lo tanto desconocemos el nivel de interés del usuario sobre ellos*), queremos estimar el interés que el usuario puede tener sobre éstos a partir de observaciones previas o información extra y recomendar al usuario los que se estime sean de su interés, de tal manera que incluso, algunas técnicas de minería de datos se aprovechan directamente (ver sección "*2.2 Basado en modelos*") en los SRs.

Así los SR son sistemas de filtrado de información y su objetivo es mostrar al usuario ítems *recomendables*, de tal manera que sean de su interés, pero que además el usuario no haya tomado en cuenta. Por ejemplo, no tiene gran impacto recomendar al usuario un producto (en este caso nuestros ítems son productos) que ya compró, dado que ya lo conoce y además suponemos que es de su agrado y por ello hizo la compra.

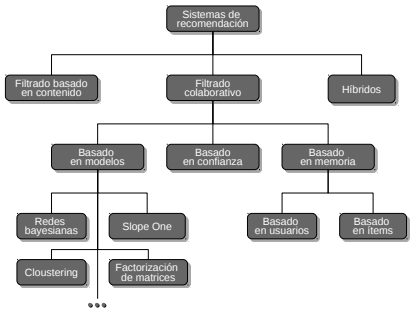

Fig. 1. Clasificación de los SRs.

Para cumplir este objetivo, los SRs deben estimar de alguna manera el interés que el usuario tiene sobre cada ítem recomendable y seleccionar los de mayor interés para éste con base en las estimaciones hechas. Por lo tanto un SR debe de tener una función que dado un usuario y un ítem, pueda estimar el nivel interés de ese usuario para el ítem dado. Esto de manera más formal lo podemos definir como en [2]:

Sea U el conjunto de todos los usuarios e I el conjunto de todos los posibles ítems recomendables. Definimos $\overline{r}$ como una función que estima la utilidad de un ítem i para el usuario u, es decir:

$$\overline{r} : U \times I \rightarrow R, \quad \text{donde } R \text{ es conjunto ordenado.} \tag{1}$$

En general dentro de los SRs, no sólo se escoge un único ítem, sino que se crea un ranking de ítems basándose en el nivel de interés estimado y se seleccionan los n mejores ítems [2].

Como se mencionó en párrafos anteriores, se han desarrollado varias técnicas para implementar la función $\overline{r}$, y podemos clasificar a los SR con base en la técnica seleccionada para ser usada en el SRs. En la Fig. 1 se muestran las técnicas más usadas para implementar SRs y a continuación se describen cada una de éstas.

2.1. Filtrado basado en contenido

El filtrado basado en contenido (**CBF**) recomienda ítems que estén dentro del perfil del usuario [6]. Este perfil se puede construir de manera explicita a partir de información solicitada al usuario, como por ejemplo usando formularios donde el usuario expresa preferencias o de manera implícita, extrayendo información de los ítems a los que el usuario ha mostrado interés anteriormente.

Este tipo de SRs depende mucho del contexto ya que se requiere que los ítems tengan un conjunto de atributos (también llamados *metadatos*) que lo describan. Estos atributos son especificados manualmente o se obtienen analizando información complementaria, cómo *tags*, comentarios, descripciones textuales o contenido multimedia, como imágenes, audio o video por ejemplo. Por otro lado, se necesita que el formato del perfil del usuario se pueda relacionar con los atributos de los ítems de tal manera que permita obtener una estimación del interés que el usuario puede tener sobre cada ítem.

La Fig. 2 muestra de manera general el proceso de los SR usando CBF, donde los pasos son **1)** extraer los atributos de los ítems, **2)** comparar éstos con el perfil del usuario y **3)** recomendar aquellos ítems que encajen mejor con el perfil del usuario[4].

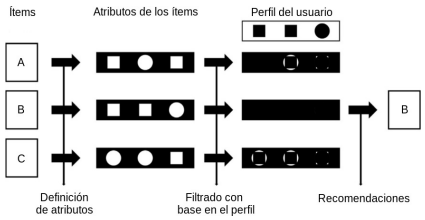

Fig. 2. Esquema general del filtrado basado en contenido.

Esta técnica es la que mejores resultados genera en las estimaciones, ya que se tiene información precisa tanto del perfil del usuario como de los ítems, pero a su vez es de las más difíciles de implementar. Por un lado está el trabajo de dar mantenimiento a la información de los ítems, ya que esto puede requerir de personas especializadas en el contexto o de una investigación extra y esto multiplicado por la cantidad de ítems que se registren en el sistema. Por otro lado, no todos los usuarios están dispuestos a llenar un perfil con sus preferencias o no se requiere de un análisis extra para definir qué actividad del usuario conviene registrar para construir su perfil de manera implícita.

2.2. Filtrado colaborativo

El filtrado colaborativo (**CF**) es el conjunto de técnicas más populares para desarrollar SRs [2]. En este conjunto de técnicas se intenta hacer la estimación

y recomendaciones con base en el comportamiento o en las calificaciones que los usuarios hacen sobre los ítems.

Algunas estas técnicas suponen que las opiniones de otros usuarios pueden ser utilizadas para poder estimar de manera las preferencias del usuario al cual se quieren hacer recomendaciones. La idea intuitiva de esto es que si un conjunto de usuarios está en cierta medida de acuerdo en el nivel de interés que tienen sobre un conjunto de ítems, entonces deberían coincidir en la misma medida en sus preferencias [9], es decir existe una relación entre los gustos de los usuarios y que encontrando esta relación se puede estimar el nivel de interés de un usuario sobre cada ítem.

La principal diferencia entre CBF y CF es que en este último no se requiere información o *metadatos* de los ítems, además de que los perfiles del usuario se reducen a tripletas del estilo *(usuario, ítem, calificación)*, donde *calificación* es un valor que refleja el nivel de interés del usuario, sobre un ítem dado. Los mecanismos que se usan para obtener esta calificación son muy variados, desde situaciones donde el usuario califica explícitamente un ítem como en *Facebook*, donde el usuario determina si una publicación le gusta, o sitios donde se le otorga una calificación numérica como *Amazon*, donde los usuarios pueden otorgar estrellas a los productos.

Dado que los perfiles de usuario son tripletas, de manera natural se pueden representar de forma matricial, donde los usuarios son las filas, los ítems las columnas y cada entrada de la matriz es la calificación correspondiente. En caso de que un usuario aún no produzca su respectiva calificación sobre un ítem tendremos un hueco en la matriz, tal y como se muestra en el ejemplo de la Fig. 3.

	A	**B**	**C**	**D**
u_1	3	2	2	1
u_2	3	1	2	5
u_3	∅	5	∅	2
u_4	5	4	2	4
u_5	2	3	4	2

Fig. 3. Matriz de calificaciones.

Basado en memoria La técnica de CF basado en memoria utiliza algoritmos que trabajan con el conjunto completo de tripletas para estimar el nivel de interés de un usuario sobre un ítem dado. Para realizar la estimación, se utilizan funciones de agregación, de tal modo que si tenemos una calificación desconocida ($r_{u,i} = \oslash$) del usuario u sobre el ítem i, podemos hacer una estimación de su valor ($\overline{r}_{u,i}$) ya sea utilizando las calificaciones de otros usuarios *similares* a u (*Basado en usuarios*) que sí han calificado i. Otra opción es usando las calificaciones que el usuario u ha hecho sobre ítems similares a i (*Basado en ítems*).

Basado en usuarios Como su nombre lo indica, esta técnica hace recomendaciones utilizando funciones de agregación sobre los usuarios. Esta es la técnica más usada para el CF. La idea intuitiva de esta técnica es que si u y un grupo de usuarios calificó de manera similar a un conjunto de ítems, las calificaciones de este grupo a un ítem i, desconocido para u, deberían ser similares a la calificación que u haría a ese ítem. De tal manera que podemos estimar la calificación de u al ítem i con base en las calificaciones de este grupo de usuarios.

El algoritmo de los *k-vecinos* es el referente el CF basado en memoria[4]. De manera general este algoritmo consiste en:

1. Se calcula la similitud entre el usuario al cual se desea hacer recomendaciones (usuario u) y cada uno de los usuarios utilizando una función $sim(u, u')$.
2. Para cada ítem i recomendable al usuario u, se selecciona el conjunto de los k usuarios más similares (también llamados *vecinos*) a éste que han calificado a i. Con base en este conjunto se estima la calificación de i ($\overline{r}_{ui}$) utilizando una función de agregación.
3. Se recomiendan los m ítems que mejor calificación tengan con base en nuestras estimaciones.

Algunos ejemplos de estas funciones de agregación que se utilizan para el CF con base en usuarios son [4]:

$$\overline{r}_{u,i} = \frac{1}{N} \sum_{u' \in \hat{U}} r_{u',i}, \tag{2}$$

$$\overline{r}_{u,i} = c \sum_{u' \in \hat{U}} sim(u, u') \times r_{u',i}, \tag{3}$$

$$\overline{r}_{u,i} = \overline{r}_u + c \sum_{u' \in \hat{U}} sim(u, u') \times (r_{u',i} - \overline{r}_{u'}), \tag{4}$$

donde $\hat{U}$ es el conjunto de los N usuarios más similares a u que sí cuentan con calificaciones sobre i. El término $\overline{r}_u$ es el promedio de las calificaciones del usuario u y se define como:

$$\overline{r}_u = \left(\frac{1}{|I_u|} \right) \sum_{i \in I_u} r_{u,i}, \quad \text{donde} \quad I_u = \{ i \in I | r_{u,i} \neq \oslash \}. \tag{5}$$

Aquí I se refiere al conjunto de todos los ítems. Y por último tenemos el término c que es un factor de normalización y en usualmente se toma como:

$$c = \frac{1}{\sum_{u' \in \hat{U}} |sim(u, u')|}. \tag{6}$$

El término $sim(u, u')$ se refiere una función que mide la similitud entre los usuarios u y u'. Como se puede observar esta función $sim(u_1, u_2)$ es parte fundamental para esta técnica más allá de la función de agregación que se escoja, por lo que en la literatura se puede encontrar una amplia gama de funciones para calcular la similitud entre usuario [2,4,14], las más usadas son: *correlación de Pearson (COR)*, *coseno (COS)*, *COR constreñido (CPC)*, *correlación de rankeo de Spearman (SRC)*, *Error cuadrático medio (MAE)* [4].

En la Fig. 4 se muestra un ejemplo de predicciones utilizando CF basado en usuarios retomado del ejemplo propuesto en [4, p. 114]. En este caso se utilizan tres vecinos ($k = 3$) y queremos recomendar tres ítems ($m = 3$). Se utilizó el

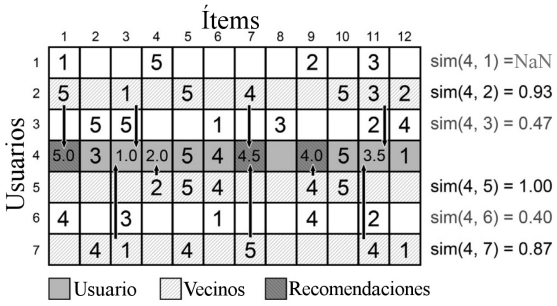

Fig. 4. Ejemplo del algoritmo *k-vecinos*

promedio como función de agregación (ecuación 2) y el error cuadrático medio para calcular la similitud entre usuarios.

Para estimar las calificaciones que el usuario 4 asignaría a los ítems 1, 3, 4, 7, 8, 9 y 11, que son los ítems que este usuario aun no califica, iniciamos calculando la similitud del usuario 4 con los demás utilizando el error cuadrático medio. Por ejemplo tomemos la similitud entre los usuarios 4 y 5:

$$C_{ab} = \{5, 6, 10\}\,,\; n = |C_{ab}| = 3.$$

Por lo tanto:

$$\begin{aligned} sim(4,5) &= 1 - \frac{1}{n} \sum_{i \in C_{ab}} |r_{4,i} - r_{5,i}|^2 \\ &= 1 - \frac{|r_{4,5} - r_{5,5}|^2 + \cdots + |r_{4,10} - r_{5,10}|^2}{3} \\ &= 1 - \frac{|5-5|^2 + |4-4|^2 + |5-5|^2}{3} = 1 - \frac{0}{3} = 1. \end{aligned}$$

Entonces seleccionamos a los k usuarios más cercanos, en este caso 2, 5 y 7. Con estos usuarios estimamos las calificaciones desconocidas usando la función de agregación seleccionada. Si tomamos como ejemplo el ítem 3, tendríamos que $\hat{U} = \{2, 7\}$ y por lo tanto:

$$\begin{aligned} \overline{r}_{4,3} &= \frac{1}{2} \sum_{u' \in \hat{U}} r_{u',3} = \frac{1}{2}(r_{2,3} + r_{7,3}) \\ &= \frac{1}{2}(1+1) \quad = \frac{2}{2} \quad = 1. \end{aligned}$$

Por último seleccionamos los m ítems con las estimaciones más altas para formar nuestra lista de recomendaciones, es decir, recomendamos los ítems 1, 7 y 9. Cabe destacar que no se puede calcular la similitud entre los usuarios 4 y 1 con la función de similitud seleccionada. Esto debido a que la expresión $\frac{1}{n}$ queda

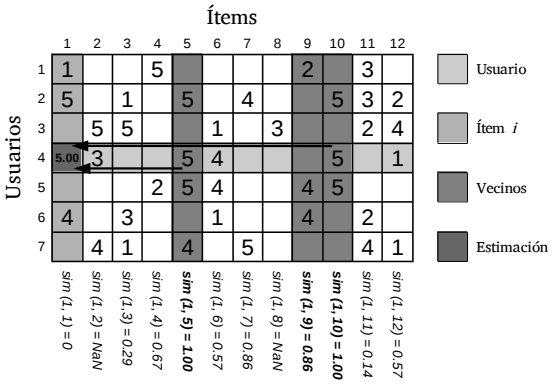

Fig. 5. Estimación para el ítem de la calificación de i ($\overline{r}_{ui}$) usando el algoritmo *k-vecinos* basado en ítems.

indeterminada al no existir ítems que ambos usuarios hayan calificado, por lo que el usuario 1 no se tomó en cuenta. De igual forma, no se puede estimar la calificación para el ítem 8 debido a que ninguno de los k usuarios más cercanos lo ha calificado. En muchos se calcula de antemano una matriz de similitud entre usuarios, de tal forma que cuando se desee hacer recomendaciones, solo se consulta esta, aunque esto tiene la desventaja de que cada vez que se agrega un ítem, usuario o calificación, se requiere volver a calcular la matriz de similitudes.

Basado en ítems El CF basado en ítems es casi idéntico al basado en usuarios, pero como su nombre indica, las funciones de similitud y de agregación se realizan sobre los ítems. En este caso para estimar el valor de $\overline{r}_{ui}$, en lugar de tomar los usuarios parecidos a u que han calificado i se toman los ítems más parecidos a i que el usuario u a calificado.

En este caso el proceso a seguir para recomendar ítems al usuario sería:

1. Para cada ítem i recomendable al usuario u :
   a) Se calcula su similitud con los ítems que u ha calificado, usando una función de similitud entre ítems ($sim(i, i')$).
   b) De los ítems que u ha calificado se seleccionan los k más similares a i y se utiliza una función de agregación para estimar la calificación del ítem i ($\overline{r}_{ui}$)
2. Se recomiendan los m ítems que mejor calificación tengan con base en nuestras estimaciones.

En el ejemplo de la Fig. 5 se puede observar la estimación de la calificación del usuario 4 al ítem 1 ($\overline{r}_{1,1}$).

De igual forma que en el CF basado en usuarios, es común que se almacene una matriz de similitud entre ítems para optimizar el tiempo de respuesta al calcular la lista de ítems recomendados.

Basado en confianza También llamados recomendaciones sociales. En esta técnica las recomendaciones se basan en la confianza entre usuarios. Es similar al filtrado colaborativo con base en usuarios pero en este caso la noción de *vecinos* y *similitud* entre usuarios es remplazada por *amigos* y *confianza* entre usuarios respectivamente. Bajo esta perspectiva, es el propio usuario quien elige a sus *vecinos*, nombrados ahora *amigos*. Y ahora se agrega una función de confianza para emular el escenario de amigos de mis amigos. Esto se basa en la idea de que las personas confían en las opiniones de sus amigos y confían en las opiniones de amigos de sus amigos en mayor o menor medida[13]. De tal manera que se puede modificar la técnica de CF con base en usuarios descrita en la sección "*2.2 Basado en usuarios*" y remplazar las funciones de *similitud* por una nueva función que determine la confianza entre usuarios.

De qué manera modelar la confianza y así poder determinar ésta entre dos usuarios se ha convertido en una tarea indispensable en muchas ramas de la web social, como la seguridad, la computación en nube e incluso en el ámbito de los SRs. Se han propuesto varios modelos para determinar la confianza entre usuarios e incorporarlos dentro de técnicas de SR. Algunos ejemplo de éstos modelos son: Modelo de Simon, MoleTrust, TidalTrust, Modelo de O'Donovan[22,18,11,20].

Basado en modelos A diferencia de los SRs basados en memoria, esta técnica de SR basados hacen recomendaciones construyendo previamente un modelo con base en las calificaciones de los usuarios. Se ataca el problema de las estimaciones como un problema de datos perdidos o de clasificación [15] y se utilizan algoritmos de *machine learning* como redes neuronales, redes bayesianas, clustering o incluso toman inspiración de algoritmos de otro tipo de problemas como factorización de matrices. Una ventaja de estas técnicas, es que son más rápidas al momento de estimar la relevancia de los ítems, aunque su desventaja es que al agregar nuevos datos, ya sea usuarios, ítems o calificaciones, el modelo necesita ser actualizado.

Slope One La técnica de *Slope One* [17] considera que se puede estimar una calificación desconocida a partir de una función lineal de otro valor conocido, de tal forma que tenemos:

$$\overline{r}_{ui} = f(r_{uj}) = r_{uj} + \delta_{ij}. \tag{7}$$

Por lo que el problema se reduce a encontrar δ_{ij} para cada par de ítems. Formalmente, dados dos vectores $v = (v_1, \ldots, v_n)$ y $w = (w_1, \ldots, w_n)$, buscamos una función $f(x) = x + \delta$ para predecir w a partir de v minimizando la expresión $\sum_{i=1}^{n} (v_i + \delta - w_i)^2$. Derivando la expresión con respecto a δ, igualando a cero y despejando tenemos que $\delta = \frac{1}{n} \sum_{i=1}^{n} (w_i - v_i)$. Por lo tanto δ es el promedio de la diferencia de cada entrada.

Ahora para encontrar δ_{ij} necesitamos un conjunto de usuarios $\hat{U}_{ij}$ tal que éstos hayan calificado ambos ítems (los ítems i y j) y calculamos el promedio de las diferencias:

$$\delta_{ij} = \frac{1}{|\hat{U}_{ij}|} \sum_{u \ in \hat{U}_{ij}} (r_{uj} - r_{ui}). \tag{8}$$

Dado que ya tenemos un estimador de $\overline{r}_{uj}$ dado r_{ui} podemos sacar la estimación promedio usando el conjunto (R_u) de todos los ítems que ha calificado el usuario u:

$$\overline{r}_{uj} = \frac{1}{|R_u|} \sum_{i \in R_u} (r_{ui} + \delta_{ij}). \tag{9}$$

Redes Bayesianas Al igual que en los SRs basados en memoria los SRs que utilizan modelos basados en redes bayesianas estiman el valor de calificación $\overline{r}_{ui}$ como una función de agregación de las calificaciones que se tienen de otros usuarios, pero en este caso se hace una suma ponderada por un factor de probabilidad [8].

Clustering La técnica de *clustering* o *k-medias* divide el conjunto de usuarios en k subconjuntos de tal manera que los elementos de cada conjunto estén lo más cerca posible en relación a una medida de distancia dada. Cada conjunto C_j (*clouster* C_j) está definido por N_j elementos y un centroide λ_j. Este centroide λ_j es un punto donde se minimiza la suma de las distancias de éste con todos los elementos que pertenecen al clouster C_j [21].

Trasladándolo a los SRs, si tomamos elemento x_n como las calificaciones del usuario n sobre el conjunto de ítems, se puede usar esta técnica para determinar el cluster C_j al que pertenece el usuario y tomar los valores λ_j para estimar los valores para los ítems que el usuario no a calificado.

Factorización de matrices La idea detrás de esta técnica es que existen factores latentes que pueden explicar por qué un usuario le da cierta calificación a un ítem dado. En caso de hablar de películas, estos factores pueden ser el genero, los actores, la historia etc. de tal manera que la calificación que el usuario u da a un ítem i tomando en cuenta estos f factores, se puede expresar como:

$$r_{ui} = q_i p_u^T, \tag{10}$$

donde q_i y p_u son vectores en $\mathbb{R}^f$ donde cada entrada se refiere a cada uno de estos factores latentes. En el caso de q_i, cada elemento refleja el grado de apego al factor dado y cada elemento de p_u refleja la relevancia de ese factor para los gustos del usuario.

Si hacemos esto para cada usuario y cada ítem, se puede construir una matriz R de tal forma que:

$$R = QP^T, \tag{11}$$

donde cada fila de Q corresponde a cada uno de los vectores q_i y las filas de P son cada uno de los vectores p_u. Entonces el problema se convierte en tratar de estimar las matrices Q y P tomando como base la matriz de calificaciones usando por ejemplo *descomposición en valores singulares* (SVD) [3, p. 44].

2.3. Sistemas de recomendación híbridos

Los SRs híbridos combinan varias técnicas de SR para mejorar las recomendaciones y mitigar los problemas particulares que presenta cada una de ellas.

La idea principal de esta aproximación es combinar distintas técnicas de SR de tal manera que preserven las virtudes de cada una y que las desventajas de una técnica particular pueden ser mitigadas con las propiedades de las otras [14,13].

Existen varias aproximaciones para combinar diferentes técnicas de SR, como ejemplo de las comúnmente usadas tenemos las siguientes técnicas:

Hibridación ponderada En esta técnica de hibridación se combinan las estimaciones de dos o más SRs utilizando una combinación lineal entre ellas de tal forma que se puede ponderar la importancia o relevancia que tendrán las estimaciones hechas por cada SR que se están combinando al asignar distintos pesos (coeficientes) a cada uno de ellas [13]. De tal manera que tendríamos:

$$\overline{r_{h}}_{ui} = \sum_{j \in \widehat{SR}} \alpha_j \overline{r_j}_{ui}, \tag{12}$$

donde:

- $\overline{r_{h}}_{ui}$ Es la estimación final de nuestro SR híbrido,
- $\overline{r_j}_{ui}$ Es la estimación del SR j del conjunto de SRs que estamos combinando ($\widehat{SR}$),
- α_j Es el factor de ponderación para el SR j. De tal forma que: $\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$ y $0 < \alpha_j < 1 \ \forall j \in \widehat{SR}$.

Hibridación en cascada La técnica de hibridación en cascada consiste en filtrar los ítems de manera escalonada. Como se muestra en la Fig. 6, se toma uno de los SRs (SR_1) de los que se desea combinar y se calcula un conjunto de ítems candidatos usando las estimaciones de éste. Se vuelve a filtrar el conjunto anterior con la siguiente técnica de SR elegida (SR_2) y se recomienda el conjunto de ítems resultante (SR_h).

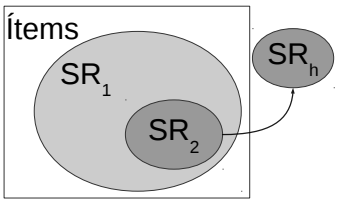

Fig. 6. Híbridación en cascada.

Hibridación por selección En este caso, se calcula la estimación de cada uno de los sistemas de recomendación a combinar y con base en un conjunto de condiciones se selecciona el valor para la estimación global [5]. De tal manera que tenemos:

$$\overline{r_{h}}_{ui} = \begin{cases} \overline{r_1}_{ui} & \textbf{si} \quad cond_1(\overline{r_1}_{ui}, \ \ldots, \ \overline{r_n}_{ui}), \\ \overline{r_2}_{ui} & \textbf{si} \quad cond_2(\overline{r_1}_{ui}, \ \ldots, \ \overline{r_n}_{ui}), \\ & \vdots \\ \overline{r_n}_{ui} & \textbf{si} \quad cond_n(\overline{r_1}_{ui}, \ \ldots, \ \overline{r_n}_{ui}), \end{cases} \tag{13}$$

donde $cond_j(\overline{r_1}_{ui}, \ldots, \overline{r_n}_{ui})$ es la condición que debe cumplirse para seleccionar la estimación del SR j ($\overline{r_j}_{ui}$). Esta notación es para generalizar la estructura y a pesar de que indicamos que se utilizan todas las estimaciones para calcular la condición, puede ser que éstas sólo involucren algunas de las estimaciones, como por ejemplo:

$$\overline{r_h}_{ui} = \begin{cases} \overline{r_1}_{ui} & \textbf{si} \quad \overline{r_1}_{ui} \neq null, \\ \overline{r_2}_{ui} & \textbf{si} \quad \overline{r_1}_{ui} = null. \end{cases} \tag{14}$$

Hibridación por mezcla Esta técnica de hibridación mezcla las recomendaciones de varios SRs en una lista de recomendaciones global. Trabaja sobre los rankings generados a partir de las estimaciones del nivel de interés del usuario sobre los ítems recomendables.

Supongamos que queremos combinar tres SRs (SR_1, SR_2 ySR_3) y que tenemos tres ítems recomendables (A, B y C) de los cuales se obtuvieron las estimaciones que se muestra el tabla (a) para el usuario u. Al calcular los rankings tenemos los resultamos como se muestra en la tabla (b) y utilizando el algoritmo de votación de Hare [3, p. 322] obtenemos como resultado el ranking de la tabla (c):

	A	B	C
SR$_1$	3	5	4
SR$_2$	2	4	3
SR$_3$	4	6	3

(a)

	Ranking		
SR$_1$	B	C	A
SR$_2$	B	C	A
SR$_3$	B	A	C

(b)

	Ranking		
SR$_h$	B	C	A

(c)

3. Comparación de técnicas de filtrado

Al intentar estimar el nivel de interés que el usuario puede tener sobre el conjunto de ítems recomendables, se presentan una serie de problemas colaterales que han propiciado el desarrollo de una gran variedad de técnicas para intentar subsanar uno o varios de estos. A continuación se listan los más importantes:

Serendipia La serendipia se define como la ocurrencia de un evento afortunado poco probable. Cuando el SR carece de ésta, se puede dar el caso de que a pesar de existir ítems de interés para el usuario, éstos se dejen de lado por tener muy poca similitud con los ítems por los que el o los usuarios han mostrado interés [16]. Este rasgo se considera una especie de sobre especialización. Por ejemplo, supongamos que tenemos un sistema donde los ítems son canciones. El usuario gusta de canciones de Rock de los años 70, pero sólo ha reproducido canciones del genero Pop, por lo que el SR no recomendaría las canciones de Rock ya que el usuario no ha mostrado interés en ellas.

Arranque en frío También llamado "*cold start*", ocurre cuando no se tiene suficiente información para generar una buena aproximación. Ya sea porque no se cuenta con un conjunto de calificaciones sobre un ítem dado o no se cuenta con información sobre los gustos del usuario. Este es un caso frecuente ya que se produce cuando un sitio es puesto en línea por primera vez o se agrega un nuevo usuario/ítem al sistema [4].

Usuarios maliciosos Se produce cuando usuarios intentan manipular el SR para que oculte/muestre ciertos ítems. Por ejemplo, un usuario podría dar una buena calificación a un libro que él mismo escribió y crear varios usuarios ficticios para votar por su libro. De esta manera, podría sesgar el SR para mostrar más a menudo dicho libro [18].

Escalabilidad Este problema se presenta cuando el volumen de los datos (número de usuario y/o número de ítems) comienza a crecer [14]. Una técnica de SR puede funcionar bien bajo un conjunto de datos limitado, pero la eficiencia y desempeño puede decaer al punto de ser no ser plausible cuando la cantidad de datos se incrementa.

Contexto del usuario En muchas de las técnicas utilizadas por los SRs se utiliza la calificación de otros usuarios sobre un ítem dado para estimar el nivel de interés que el usuario objetivo tiene sobre éste. Se busca los usuarios *similares*, entendiendo similar como que ha demostrado apreciaciones similares en varios ítems, pero dentro de esta similitud no se toma en cuenta el trasfondo de la calificación. Por ejemplo, puede que a dos usuarios les guste la misma película y que hayan asignado la misma calificación a ésta, pero mientras el primero sólo está interesado en la fotografía, el otro la considera una buena película por los actores involucrados [1].

Dificultad de implementación En muchos casos el SR resulta difícil de implementar. Por ejemplo puede requerir de un esfuerzo extra por parte de los usuarios, como es el caso del CBF, o porque se requiere de un entendimiento profundo del contexto de los ítems o de los usuarios a pesar de que en la parte computacional sea relativamente sencillo de procesar. En otro casos, la solución computacional exige herramientas y conocimientos extra, como es el caso de los SRs basados en modelos.

En la tabla 1 se desglosan brevemente los problemas comunes en SRs y si afecta o no a cada técnica descrita.

4. Otras áreas de investigación en sistemas de recomendación

Recomendaciones con base en dominios cruzados

La idea detrás de esta área de investigación es que se pueden mejorar las recomendaciones haciendo uso de la información recolectada en distintos sistemas (nombrados *dominios*), de tal modo que se pueda construir un perfil unificado y más completo del usuario juntando la información recolectada en cada sistema mejorando las recomendaciones en cada uno de ellos.

Tabla 1. Comparación de sistemas de recomendación

			Serendipia	Arranque en frío	Usuarios maliciosos	Escalabilidad	Contexto del usuario	Dificultad al implentar
Basado en contenido			**N**	**N**	**N**	**N**	**N**	**S**
Filtrado colaborativo	Basado en memoria	Usuarios	**S**	**S**	**S**	**S**	**S**	**N**
		Ítems	**S**	**S**	**S**	**S**	**S**	**N**
	Basado en confianza		**N**	**S**	**N**	**S**	**S**	**M**
	Basado en Modelos	Red bayesiana	**S**	**N**	**S**	**N**	**N**	**M**
		Slope One	**S**	**N**	**S**	**N**	**N**	**N**
		Cloustering	**S**	**N**	**S**	**N**	**N**	**M**
		Factorización de matrices	**N**	**N**	**S**	**N**	**N**	**M**
Híbridos			**N**	**N**	**N**	**N**	**N**	**S**

N = no afecta, **S** = sí afecta, **M** = un punto intermedio

Sistemas sociales de etiquetado STS (Social Tagging Systems)

Hoy en día existen muchos sitios Web que permiten al usuario publicar y compartir contenido propio, como por ejemplo: Flickr, Delicius o Youtube. Muchos de estos sitios permiten a los usuarios caracterizar sus contenidos con etiquetas (*tags*) de texto sin ninguna restricción en los términos que se utilizan. Para evitar ambigüedad y mejorar el etiquetado se busca un SR que oriente a los usuarios sobre los términos a usar, esto bajo la idea de que dos contenidos similares deberían tener etiquetas similares y que el sistema puede hacer uso de las etiquetas de otros contenidos para extraer las más relevantes para el usuario en orden de etiquetar su contenido [19].

Problemas de privacidad en SRs

Los SRs recolectan información acerca del usuario a lo largo del tiempo, de tal manera que las recomendaciones personalizadas van siendo mejores, pero esto tiene un impacto negativo en la privacidad ya que el usuario puede comenzar a sentir que el sistema sabe demasiado acerca de él y temer que gente externa pueda obtener su información sensible. Por lo que ha surgido la necesidad de crear mecanismos que mantengan la privacidad, evitando que la información sea visible para agentes maliciosos pero al mismo tiempo permita al SR hacer recomendaciones acertadas [7].

Integración de las preferencias a corto y largo plazo

Actualmente los SRs hacen recomendaciones utilizando el conjunto completo de los datos sobre los intereses del usuario que se han recolectado hasta el momento sin tomar en cuenta que éstos pueden diferir a lo largo del tiempo o que en este instante de tiempo el usuario tiene gustos particulares [10].

5. Conclusiones

Los SRs están cada vez más presentes en la vida diaria del ser humano. Las personas los usan consciente o inconscientemente para encontrar información en diferentes contextos tales como libros, música, noticias, viajes e incluso relaciones románticas. Cada vez más sistemas informáticos que ofrecen productos, servicios o simplemente información tienen inmersos algún tipo de SR para apoyar a las personas a escoger entre las innumerables alternativas que éstos ofrecen.

Los SR basados en memoria en general dan buenos resultados si se tiene suficiente información y son simples de implementar y entender; en términos de estimaciones puntuales la mejor técnica hasta el momento parce ser SVD, la cual gano el "*Netflix's prize* " [23] en el 2009.

Alrededor del tema de SR se ha creado una comunidad de investigación creciente, que intenta innovar y solventar los muchos problemas que aún se encuentran por resolver. Para abrir nuevas líneas de investigación en esta área, la intuición por si sola ya no es suficiente y es necesario un enfoque multidisciplinario para traer mejores herramientas que puedan ayudar a explotar el inmenso potencial de los SR en aplicaciones del mundo real.

Referencias

1. Adomavicius, G., Tuzhilin, A.: 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 4(6), 734–749 (2005), http://pages.stern.nyu.edu/TILDEatuzhili/pdf/TKDE-Paper-as-Printed.pdf
2. Amatriain, X.: Recommender Systems . Machine Learning Summer School 2014 @ CMU (July 2014), http://www.slideshare.net/xamat/recommender-systems-machine-learning-summer-school-2014-cmu/
3. Blum, A., Hopcroft, J., Kannan, R.: Foundations of Data Science (2015), https://www.cs.cornell.edu/jeh/book2016June9.pdf
4. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowledge-Based Systems 16(46), 109–132 (April 2013), http://dx.doi.org/10.1016/j.knosys.2013.03.012
5. Burker, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction (2002), http://josquin.cs.depaul.edu/TILDErburke/pubs/burke-umuai02.pdf
6. Candillier, L., Jack, K., Fessant, F., Meyer, F.: Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling (2009), https://www.researchgate.net/publication/275890626_State_of_the_Art_Recommender_System
7. Canny, J.: Collaborative Filtering with Privacy via Factor Analysis (August 2002), https://people.eecs.berkeley.edu/~jfc/papers/02/SIGIR02.pdf
8. Dechoux, B.: Recommender Systems Naive Bayes Networks and the Netflix Prize (January 2009), http://lemire.me/fr/documents/publications/lemiremaclachlan_sdm05.pdf
9. Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative Filtering Recommender Systems. NOW the essence of knowledge 4(2), 81–173 (2011), http://files.grouplens.org/papers/FnTCFRecsysSurvey.pdf

10. Francesco Ricci, L.R., Shapira, B.: Introduction to Recommender Systems Handbook (October 2010), http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf
11. Golbeck, J.A.: Computing and Applying Trust in Web-based Social Networks. Ph.D. thesis, University of Maryland at College Park (2005), https://www.researchgate.net/publication/243786854_Computing_and_Applying_Trust_in_Web-Based_Social_Networks
12. Hanani, U., Shapira, B., Shoval, P.: User Modeling and User-Adapted Interaction. Information Filtering: Overview of Issues, Research and Systems 11 (2001), http://link.springer.com/article/10.1023/A:1011196000674
13. Haydar, C.A.: Les systèmes de recommandation à base de confiance. Ph.D. thesis, Université de Lorraine, France (September 2014), http://docplayer.fr/storage/23/1891872/1481315000/N5fnyTd388npoHe6T33jEg/1891872.pdf
14. Isinkaye, N., Folajimi, Y., Ojokoh, B.: Recommendation systems: Principles methods and evaluation. Egyptian Informatics Journal 16(20), 262–271 (Agosto 2015), http://dx.doi.org/10.1016/j.eij.2015.06.005
15. Iván, C.G.: Exploiting the conceptual space in hybrid recommender systems: a semantic-based approach. Ph.D. thesis, Universidad Autónoma de Marid, Espain (October 2008), http://josquin.cs.depaul.edu/rburke/pubs/burke-umuai02.pdf
16. Lakshmi, S.S., Lakshmi, T.A.: Recommendation Systems:Issues and challenges. International Journal of Computer Science and Information Technologies 5(4), 5771–5772 (2014), http://www.ijcsit.com/docs/Volume5/vol5issue04/ijcsit20140504207.pdf
17. Lemire, D., Maclachlan, A.: Slope One Predictors for Online Rating-Based Collaborative Filtering (February 2005), http://lemire.me/fr/documents/publications/lemiremaclachlan_sdm05.pdf
18. Massa, P., Avesani, P.: Trust-aware Collaborative Filtering for Recommender Systems. Via Sommarive 4(4), 1–17 (2005), https://pdfs.semanticscholar.org/2512/182cf3c4d7b3df549456fbeceee0a77c3954.pdf
19. Musto, C., Narducci, F., Gemmis, M.D., Lops, P., Semeraro, G.: A Tag Recommender System Exploiting User and Community Behavior (October 2009), https://pdfs.semanticscholar.org/e50b/ebdd1ed1f3c6a1107ef6c78697c234982bed.pdf#page=40
20. O'Donovan, J., Smyth, B.: Trust in recommender systems. Proceedings of the 10th international conference on intelligent user interfaces ACM p. 167–174 (2005), https://www.researchgate.net/profile/Barry_Smyth/publication/221608315_Trust_in_recommender_systems/links/0fcfd50f3fa6b4ec86000000.pdf
21. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommender Systems Handbook, vol. 1 (2011), http://www.cs.ubbcluj.ro/TILDEgabis/DocDiplome/SistemeDeRecomandare/Recommender_systems_handbook.pdf
22. Simon, M., Lionel, M., Frédérique, L.: Recommandation sociale et locale basée sur la confiance. Document numérique pp. 33–56 (2012), https://www.researchgate.net/publication/274665784_Recommandation_sociale_et_locale_basee_sur_la_confiance
23. Wikimedia Foundation, Inc: Netflix Prize (February 2017), https://en.wikipedia.org/wiki/Netflix_Prize