
Sistemas recomendadores

PID_00279462

Marc Maceira Duch
Carles Ventura Royo

Tiempo mínimo de dedicación recomendado: 3 horas



Marc Maceira Duch**Carles Ventura Royo**

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por los profesores: Marc Maceira Duch, Carles Ventura Royo

Primera edición: febrero 2021

© de esta edición, Fundació Universitat Oberta de Catalunya (FUOC)

Av. Tibidabo, 39-43, 08035 Barcelona

Autoría: Marc Maceira Duch, Carles Ventura Royo

Producción: FUOC



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia Creative Commons de tipo Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0. Se puede copiar, distribuir y transmitir la obra públicamente siempre que se cite el autor y la fuente (Fundació per a la Universitat Oberta de Catalunya), no se haga un uso comercial y ni obra derivada de la misma. La licencia completa se puede consultar en: <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

Índice

1	Introducción a los sistemas recomendadores	5
2	Filtraje colaborativo	9
2.1	Filtraje colaborativo basado en usuarios.....	10
2.2	Filtraje colaborativo basado en ítems	13
2.3	Filtraje colaborativo basado en factorización de matrices	16
2.4	Dificultades	18
3	Sistemas recomendadores basados en contenido	19
4	Sistemas recomendadores basados en conocimiento	22
4.1	Sistemas recomendadores basados en restricciones.....	24
4.2	Sistemas recomendadores basados en casos	25
5	Sistemas recomendadores híbridos	28
6	Evaluación de sistemas recomendadores	31
6.1	Evaluación basada en datos históricos	31
6.2	Evaluación basada en estudios con usuarios reales	32
	Bibliografía	34

1. Introducción a los sistemas recomendadores

Actualmente, hay una gran oferta de contenidos en las diversas plataformas que ofrecen servicios por internet. Servicios como YouTube, Spotify o Netflix generan más contenidos de los que una persona es capaz de consumir en toda una vida. De forma similar, tiendas en línea como Amazon, eBay o Wallapop disponen de una gran cantidad de productos en venta. Uno de los problemas que tienen estas plataformas es que los usuarios están desconcertados por la cantidad de servicios o productos disponibles. Ante la gran cantidad de oferta, los usuarios no saben por qué producto decidirse. Hay un problema parecido en redes sociales como Facebook, Instagram o Twitter. El objetivo de estas redes consiste en tener a los usuarios conectados el máximo tiempo posible, puesto que así les pueden mostrar más anuncios.

Para ordenar los productos disponibles y facilitar la elección a los clientes, disponemos de los sistemas recomendadores.

El sistema de software que determina qué productos o servicios se tienen que mostrar a un visitante particular es un sistema de recomendación.

Si consideramos el ejemplo de una tienda en línea, cuando escribimos el nombre de un producto, este aparece como uno de los productos recomendados. Adicionalmente, la tienda nos propone otros productos en un apartado de la página: «Los clientes que han comprado este producto también han comprado...». Esta lista de recomendaciones es personalizada, utiliza el producto buscado y el historial del cliente para ofrecerle productos que le pueden interesar.

En general, los sistemas de recomendación tienen dos objetivos diferenciados: por un lado, combatir la sobrecarga de información seleccionando los artículos más interesantes para el usuario; por otro lado, estimular a los usuarios a hacer alguna acción como comprar un libro específico o ver una película en concreto. Muchas veces los dos objetivos son complementarios, pero dependiendo del modelo de negocio de la empresa, será el uno o el otro. Por ejemplo, a YouTube le interesa combatir la sobrecarga de información para que los usuarios sigan viendo vídeos en la plataforma. En cambio, Amazon estimula la compra de productos concretos a los usuarios.

Un problema similar al de los sistemas recomendadores aparece al intentar extraer información útil a partir de datos. La gran mayoría de instrumentos de

medida (desde magnitudes físicas como longitud, temperatura y tiempo hasta magnitudes de comportamiento como patrones de búsqueda y navegación en la web y preferencias de compra en línea, pasando por herramientas comunes como las cámaras digitales) son capaces de volcar las medidas en algún formato digital; de este modo, todos estos datos quedan disponibles y pueden ser utilizados por algoritmos de aprendizaje automático.

En este momento, el problema no es disponer de datos, puesto que los tenemos en abundancia: el reto es conseguir extraer información a partir de los datos, es decir, darles un sentido y obtener conclusiones útiles. Esta tarea se conoce con el nombre de minería de datos.*

* En inglés, *data mining*

La investigación en sistemas recomendadores tiene una gran relación con el filtraje y la extracción de la información. Muchas de las técnicas desarrolladas en estas áreas explotan información derivada del contenido de los documentos para clasificarla entre contenidos relevantes y no relevantes.

Podemos utilizar el escenario de compras por internet para hablar de varios aspectos de los sistemas recomendadores:

- **Las recomendaciones son personalizadas:** cada usuario recibe unas recomendaciones diferentes dependiendo de sus gustos. La alternativa sería la opción impersonal de recomendar a todos los usuarios los productos más vendidos. Si ofrecemos el producto más vendido a todo el mundo (la canción más escuchada, el *best-seller* más leído, la cerveza más popular, etc.) nos adaptaremos a mucha gente pero no seremos capaces de recomendar productos específicos a cada usuario.
- **El sistema necesita conocer al usuario:** la presentación de recomendaciones personalizadas hace necesario que el sistema sepa algo sobre los usuarios y los productos que ofrece. Por ejemplo, se pueden utilizar las valoraciones de los usuarios sobre los productos que han comprado. Alternativamente, podemos crear un modelo de usuario con las preferencias o el historial de compra del usuario. En un servicio de películas en reproducción en continuo, si sabemos que el usuario acostumbra a mirar películas románticas y no se ha interesado por ninguna película de terror, no será necesario que le recomendemos películas de este género.
- **Hay que definir cómo se adquiere y se utiliza la información de los usuarios:** el modelo de usuario puede adquirir las preferencias de los usuarios preguntando explícitamente al usuario sobre sus preferencias. Muchos sistemas nos preguntan esta información una vez nos hemos registrado en un nuevo servicio. Otra opción es que el sistema monitorice el comportamiento de los usuarios para generar el modelo.
- **Hay que utilizar el comportamiento colectivo:** una vez se ha generado el modelo de usuario, el sistema puede explotar el comportamiento y los gustos de toda la comunidad para hacer las recomendaciones. Saber qué

historial de compra han tenido los usuarios similares a un usuario ayuda a recomendar nuevos productos. Estos sistemas se denominan colaborativos.

Este módulo presenta los aspectos de los sistemas de recomendaciones consolidados y probados extensamente. La figura 1 muestra las familias de sistemas recomendadores estudiadas en este módulo. A continuación destacamos las características principales:

- **Filtraje colaborativo***: la idea básica de estos sistemas es que, si los usuarios comparten los mismos intereses (por ejemplo, si han visto o han comprado los mismos libros), también tendrán gustos similares en el futuro. Por lo tanto, si, por ejemplo, dos usuarios tienen un historial de compras muy similar y uno de ellos ha comprado recientemente un libro que el otro todavía no ha visto, el sistema de recomendación colaborativo propondrá este libro al segundo usuario. Dependiendo del tipo de información utilizada para hacer la recomendación, tendremos diferentes tipos de filtraje colaborativo. Siguiendo con el ejemplo de los libros, si utilizamos perfiles de usuarios similares para hacer la recomendación, hablaremos de *filtraje colaborativo basado en usuarios*. Si usamos la forma en que han sido valorados los productos por otros usuarios, hablaremos de *filtraje colaborativo basado en ítems*. También veremos métodos que tratan las valoraciones de los usuarios sobre los ítems como una descomposición matricial. Denominaremos a estos métodos *filtraje colaborativo basado en factorización de matrices*.
- **Sistemas recomendadores basados en contenido***: en esta familia de métodos la recomendación se hace a partir de las descripciones de los elementos y de los perfiles de los usuarios. Igual que en la recomendación colaborativa solo se utilizan las valoraciones de los usuarios, en la basada en contenido se usa información de cada producto. Volviendo al ejemplo de la librería, podemos caracterizar los libros según el tema que tratan, el autor o el género. Tanto las descripciones de los productos como los perfiles de los usuarios pueden ser creados manualmente o extraídos automáticamente.
- **Recomendación basada en conocimiento***: en otros ámbitos de aplicaciones, como la electrónica de consumo, encontramos a menudo un gran número de compradores que solo compran un producto (por ejemplo, una nevera) cada pocos años. Esto quiere decir que no podemos confiar en la existencia de un historial de compras, requisito previo para enfoques de filtraje colaborativo basado en contenido. Por otro lado, a veces, podemos disponer de un contenido más detallado y bien estructurado para generar recomendaciones. En estos enfoques, que llamamos basados en conocimiento, el sistema recomendador utiliza información proporcionada manualmente del usuario actual y los datos estructurados de los elementos disponibles.

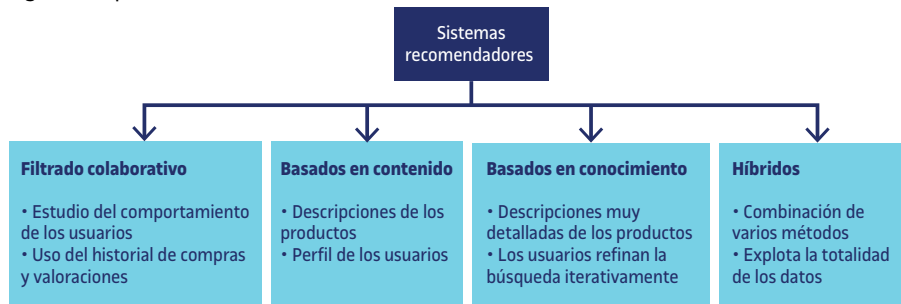
* En inglés, *collaborative filtering*

* En inglés, *content-based recommender system*

*En inglés, *knowledge-based recommender system*

- **Sistemas recomendadores híbridos:** son sistemas que combinan los sistemas recomendadores básicos (filtraje colaborativo, recomendadores basados en contenido y recomendadores basados en conocimiento) para ofrecer mejores recomendaciones. Cada método básico explora fuentes de información diferentes. Combinando diferentes sistemas recomendadores, se pueden explotar las ventajas de los distintos métodos para obtener un sistema recomendador que explote la totalidad de los datos disponibles.

Figura 1. Tipos de sistemas recomendadores estudiados en este módulo



2. Filtraje colaborativo

La idea básica de un sistema recomendador basado en filtraje colaborativo (*collaborative filtering*) es que, si dos usuarios comparten intereses parecidos y uno de ellos ha comprado o ha valorado positivamente un ítem que el otro no ha comprado, es razonable que este ítem se recomiende a este último usuario. Este sistema se denomina filtraje colaborativo porque la selección de unos ítems concretos de una gran colección de ítems implica un filtraje hecho gracias a una colaboración implícita de los usuarios de la misma plataforma. Como veremos con las diferentes técnicas que se presentarán en este apartado, las técnicas puras de filtraje colaborativo no utilizan ningún tipo de información de los ítems (en el apartado 3 veremos los sistemas recomendadores basados en contenido) y tampoco ningún tipo de información respecto a las preferencias del usuario (en el apartado 4 veremos los sistemas recomendadores basados en conocimiento).

Así pues, las técnicas de filtraje colaborativo están basadas únicamente en las valoraciones (implícitas o explícitas) que hacen los usuarios de los ítems. Se considera que las valoraciones son explícitas cuando el usuario valora el ítem en una escala de valores (por ejemplo, una escala de estrellas del 1 al 5). Por otro lado, las valoraciones implícitas son las que se consideran por el simple hecho que un usuario ha adquirido un determinado ítem. Aunque no haya hecho ninguna valoración del ítem, la adquisición se considera una acción relevante para un sistema recomendador (el usuario ha escogido este ítem entre un conjunto muy amplio de ítems posibles).

En la tabla 1 se muestra un ejemplo meramente ilustrativo de valoraciones implícitas que cuatro usuarios han hecho sobre ocho películas. Cada fila representa las valoraciones de un usuario y cada columna representa las valoraciones de una película. Tal como es habitual en los sistemas recomendadores, esta matriz de valoraciones está incompleta, puesto que no se dispone de las valoraciones de todas las películas por cada uno de los usuarios.

Tabla 1. Valoraciones de ocho películas (ejemplo)

	P1	P2	P3	P4	P5	P6	P7	P8
Usuario 1		3	1		4	3	5	
Usuario 2	4	1	3		5			2
Usuario 3	2	1		5				1
Usuario 4	3		2			5		4

En este apartado veremos cómo se encuentran usuarios que tienen preferencias similares, y analizaremos cómo se mide esta similitud entre los usuarios.

2.1 Filtrado colaborativo basado en usuarios

La estrategia basada en los usuarios más próximos es uno de los primeros métodos que apareció. Consiste en identificar cuáles son los usuarios que tienen las preferencias más similares al usuario objetivo (u_o) a quien queremos hacer la recomendación. Así pues, por cada ítem que el usuario u_o no haya valorado, se hará una predicción de la valoración de este ítem basada en las valoraciones hechas de este ítem por otros usuarios con las preferencias más similares. Si nos fijamos en la tabla 1, vemos que cada usuario está representado en una fila diferente, y por lo tanto lo que queremos buscar son otros usuarios (filas de la matriz) similares a aquel al que queremos hacer la recomendación. Este objetivo queda representado en la figura 2.

Figura 2. Representación del filtrado colaborativo basado en usuarios

	Ítem 1	Ítem 2	Ítem 3	Ítem 4	Ítem 5	Ítem 6	Ítem 7	Ítem 8
Usuario 1								
Usuario 2								
Usuario 3								
Usuario 4								
Filtrado basado en usuarios								
Usuario 5								
Usuario 6								
Usuario 7								
Usuario 8								
Usuario 9								
Usuario 10								

Interacciones positivas
 Interacciones negativas
 Interacciones neutras

Figura 2

Representación del filtrado colaborativo basado en usuarios. El objetivo es encontrar usuarios (filas de la matriz) similares al usuario al que tenemos que hacer una recomendación.

A la hora de medir la similitud entre los diferentes usuarios para encontrar a los usuarios que tienen las preferencias más similares, el planteamiento más básico que podríamos hacer es usar la distancia euclídea entre los vectores formados por las valoraciones de los ítems por parte de dos usuarios. Estos vectores se formarían considerando únicamente los ítems que los dos usuarios habrían valorado. Siguiendo el ejemplo de la tabla 1, si calculáramos la similitud entre los usuarios 1 y 2, lo haríamos solo considerando las películas 2, 3 y 5, puesto que son las que los dos usuarios han valorado. Así pues, calcularíamos la distancia euclídea de la siguiente forma:

$$d(u_1, u_2) = \sqrt{(3-1)^2 + (1-3)^2 + (4-5)^2} = \sqrt{9} = 3 \quad (1)$$

Hay que tener en cuenta que se trata del cálculo de una distancia y, por lo tanto, los usuarios más parecidos serán los que tengan unos valores de distancia más pequeños. En cualquier caso, podemos ver que uno de los problemas del cálculo de la distancia euclídea es que depende del número de ítems que tengan dos usuarios en común. Así pues, cuanto más ítems tengan en común dos usuarios, más probable es que el valor de la distancia sea mayor (más ele-

mentos tendrá el sumatorio), y esta no es una propiedad que nos interese a la hora de encontrar a los usuarios más similares.

Este es uno de los motivos por los que se suele utilizar el **coeficiente de correlación de Pearson** para calcular la similitud entre dos usuarios. Se trata de medir como los dos vectores, uno frente al otro, se organizan en torno a una línea recta (línea de mejor ajuste). Cuanto más parecidas sean las tendencias de valoraciones que hacen dos usuarios, más próximo a 1 será el coeficiente de correlación de Pearson. En cambio, cuando no se pueda establecer ninguna relación entre las valoraciones hechas por los dos usuarios, el coeficiente será próximo a 0. Por otro lado, si dos usuarios tienen comportamientos totalmente opuestos en cuanto a las valoraciones de los ítems, entonces el coeficiente será próximo a -1.

Si denotamos como a_i y b_i las valoraciones de usuarios sobre dos muestras de datos alineados (los dos usuarios han valorado los ítems $i = 1..n$). El cálculo del coeficiente de correlación de Pearson viene determinado por la formula:

$$sim(a,b) = \frac{\sum_{i=1}^n (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^n (a_i - \bar{a})^2} \sqrt{\sum_{i=1}^n (b_i - \bar{b})^2}} \quad (2)$$

donde \bar{a} es la media de los valores de a , \bar{b} la media de los valores de b y n el número de ítems valorados por ambos usuarios. Notad que para efectuar el cálculo los datos tienen que estar alineados: en nuestro caso, solo se tienen que tomar las valoraciones comunes a los dos usuarios. Notad también que con el resto de la media de los valores de cada usuario solucionamos el problema de tener usuarios con tendencia a hacer valoraciones muy altas y usuarios con tendencia a hacerlas muy bajas. De este modo, las valoraciones siempre son respecto al valor medio del usuario.

Como se deduce de la definición, el coeficiente de correlación de Pearson es simétrico y vale 1 al calcularlo respecto a la misma variable. En la tabla 2 se pueden ver los valores correspondientes a las valoraciones de los usuarios de la tabla 1. En este ejemplo la correlación entre los usuarios 1 y 3 es 0 porque solo tienen una película en común, con lo que la recta de ajuste no se puede definir.

Tabla 2. Coeficiente de correlación de Pearson de las valoraciones dadas en la tabla 1

	Usuario 1	Usuario 2	Usuario 3	Usuario 4
Usuario 1	1.0	0.33	0.0	1.0
Usuario 2	0.33	1.0	0.95	-0.5
Usuario 3	0.0	0.95	1.0	-1.0
Usuario 4	1.0	-0.5	-1.0	1.0

También se observa que los usuarios 1 y 4 tienen una correlación de 1.0; esto es porque su tendencia a valorar las películas es igual independientemente de factores de escala. Por otro lado, los usuarios 3 y 4 hacen valoraciones contrarias, por lo cual su coeficiente es -1.0 . Sin embargo, sería conveniente disponer de más datos para obtener medidas más realistas.

Una vez calculados los valores de similitud entre los usuarios, hay que decidir cuáles son los usuarios que consideraremos más parecidos para calcular la predicción de las valoraciones que faltan. Supongamos que queremos saber si recomendaríamos o no la película 8 al usuario 1 y, por lo tanto, queremos obtener una predicción de esta valoración. Según la tabla 2, hemos podido ver que el usuario 4 sería el más similar (con un coeficiente de 1.0), seguido del usuario 2 (0.33), y finalmente el usuario 3 sería el menos similar (0.0).

Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson es útil como medida de similitud porque es independiente de los desplazamientos y escalas de los valores estudiados.

A la hora de decidir qué usuarios tenemos en cuenta, lo podemos hacer de dos maneras. La primera consistiría en decidir un número fijo de usuarios más similares para considerar. Si, por ejemplo, este número fuera 2, en nuestro ejemplo consideraríamos a los usuarios 4 y 2, puesto que son los dos usuarios más similares al usuario 1. La segunda manera consistiría en decidir un umbral de similitud, de forma que consideraríamos todos los usuarios que tengan un valor de similitud superior a este umbral. Si, por ejemplo, este umbral fuera 0.5, en nuestro ejemplo solo consideraríamos el usuario 4.

Una vez decididos los usuarios más similares, podemos calcular la predicción de la valoración que hace un usuario a de un ítem i de la forma siguiente:

$$pred(a,i) = \bar{a} + \frac{\sum_{b \in N} sim(a,b)(b_i - \bar{b})}{\sum_{b \in N} sim(a,b)} \quad (3)$$

donde N es el conjunto de usuarios más parecidos que consideramos, ya sea mediante un número fijo o un umbral mínimo de similitud. Notad que en esta fórmula, el uso de los valores de similitud entre los dos usuarios $sim(a,b)$, se utiliza para dar más peso a las valoraciones hechas por los usuarios que tienen unas preferencias más similares. Así pues, siguiendo con el ejemplo de la tabla 2, si seleccionamos a los usuarios más parecidos según un número fijo (por ejemplo, 2), daremos más peso al usuario 4 que al usuario 2, puesto que el valor de similitud del primero es de 1.0, mientras que el del segundo es de 0.33. Así pues, la predicción de la valoración de la película 8 por parte del usuario 1 la calcularíamos de la forma siguiente:

$$pred(1,8) = 3.2 + \frac{1.0 * (4 - 3.5) + 0.33 * (2 - 3)}{1.0 + 0.33} = 3.2 + \frac{0.17}{1.33} = 3.33 \quad (4)$$

Si, en lugar de utilizar los dos usuarios más parecidos, hubiéramos usado el umbral de 0.5, solo habríamos considerado al usuario 4 y el valor de la predicción habría sido diferente:

$$pred(1,8) = 3.2 + \frac{1.0 * (4 - 3.5)}{1.0} = 3.2 + 0.5 = 3.7 \quad (5)$$

2.2 Filtrado colaborativo basado en ítems

En escenarios realistas de sistemas recomendadores, suele ser habitual que el número de usuarios sea significativamente más elevado que el número de ítems. Por ejemplo, podríamos estar hablando de un sistema recomendador con millones de usuarios y miles de ítems. En estos casos, el cálculo de los usuarios más parecidos sería una operación costosa, puesto que habría que hacer millones de comparaciones y realizar una ordenación de estos usuarios para escoger a los N más similares, lo que haría impracticable dar recomendaciones a los usuarios en tiempo real.

En estos casos, otra técnica de filtrado colaborativo es la basada en ítems en lugar de usuarios. Esta técnica consiste en encontrar los ítems que tienen unas valoraciones más parecidas entre el resto de los usuarios.

Volviendo al ejemplo inicial de la tabla 1, supongamos que queremos calcular de nuevo la predicción que haría el usuario 1 de la película 8, pero ahora utilizando esta técnica de filtrado colaborativo basada en ítems. Tenemos que la película 8 ha obtenido unas valoraciones de 2, 1 y 4 por parte del resto de usuarios. Lo que tendríamos que hacer es encontrar qué otros ítems han obtenido unas valoraciones parecidas. En la figura 3 podemos ver una representación del filtrado colaborativo basado en ítems.

Figura 3. Representación del filtrado colaborativo basado en ítems

	Ítem 1	Ítem 2	Ítem 3	Ítem 4	Ítem 5	Ítem 6	Ítem 7	Ítem 8
Usuario 1	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones negativas	Interacciones positivas
Usuario 2	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 3	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 4	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 5	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 6	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 7	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 8	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 9	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas
Usuario 10	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones positivas	Interacciones negativas	Interacciones positivas	Interacciones positivas

Figura 3

Representación del filtrado colaborativo basado en ítems. El objetivo, en este caso, es encontrar ítems (columnas de la matriz) en que varios usuarios hayan tenido valoraciones similares a las del usuario al que tenemos que hacer una recomendación.

A diferencia del filtraje colaborativo basado en usuarios, donde la similitud se suele calcular con el coeficiente de correlación de Pearson, en el filtraje colaborativo basado en ítems la **similitud coseno** suele ser la medida de similitud más establecida. La similitud coseno es la métrica de similitud estándar, ya que se ha demostrado que es la que obtiene resultados más afinados. La similitud coseno entre dos vectores x e y se define:

$$\text{sim}(x,y) = \frac{x \cdot y}{|x| * |y|} \quad (6)$$

donde $x \cdot y$ es el producto escalar de los vectores x e y , y $|x|$ e $|y|$ son las normas de los vectores x e y , respectivamente. Recordad que la norma de un vector se calcula como la raíz cuadrada del producto escalar del vector consigo mismo.

En nuestro caso, los vectores x y y corresponden a valoraciones de ítems por parte de usuarios diferentes. En el ejemplo de las películas, la similitud entre la película 8, con valoraciones 2, 1 y 4, y la película 1, con valoraciones 4, 2 y 3, se calcularía:

$$\text{sim}(1,8) = \frac{4 * 2 + 2 * 1 + 3 * 4}{\sqrt{4^2 + 2^2 + 3^2} * \sqrt{2^2 + 1^2 + 4^2}} = 0.89 \quad (7)$$

Los valores posibles de los valores de similitud coseno son entre 0 y 1, donde los valores más próximos a 1 indican una similitud mayor.

Uno de los problemas de la similitud coseno es que no tiene en cuenta la diferencia entre el valor medio de las valoraciones que hace cada usuario. Para poner un ejemplo, no es lo mismo una valoración de 4 de un usuario cuya media de valoraciones es un 2 que una valoración de 4 de un usuario cuya media de valoraciones es un 4.5. En el primer caso, hablaríamos de un usuario que tiene una tendencia a hacer valoraciones bajas y, por lo tanto, la película valorada con un 4 sería una valoración muy positiva, mientras que en el segundo caso hablaríamos de un usuario que tiene una tendencia a hacer valoraciones altas y, por lo tanto, a pesar de que la valoración de 4 pueda parecer buena, hay que tener en cuenta que está por debajo de su valor medio de valoraciones.

Para resolver el problema anterior, se calcula una medida alternativa llamada similitud ajustada coseno. Esta medida es equivalente a calcular la similitud coseno sobre unas valoraciones que hayan sido normalizadas previamente restando a cada valoración la media de las valoraciones del usuario correspondiente. Los valores de la similitud ajustada coseno tienen valores entre -1 y 1, tal como pasa con el coeficiente de correlación de Pearson.

Si seguimos con el ejemplo de las valoraciones de la tabla 1 y las normalizamos restando a cada valoración la media de las valoraciones del usuario correspondiente, obtenemos las valoraciones de la tabla 3.

Tabla 3. Valoraciones normalizadas de ocho películas (ejemplo)

	P1	P2	P3	P4	P5	P6	P7	P8
Usuario 1		-0.2	-2.2		0.8	-0.2	1.8	
Usuario 2	1	-2	0		2			-1
Usuario 3	-0.25	-1.25		2.75				-1.25
Usuario 4	-0.5		-1.5			1.5		0.5

Así pues, la similitud entre la película 8 y la película 1 con las valoraciones normalizadas se calcularía:

$$sim(1,8) = \frac{1 * (-1) + (-0.25) * (-1.25) + (-0.5) * 0.5}{\sqrt{1^2 + (-0.25)^2 + (-0.5)^2} * \sqrt{(-1)^2 + (-1.25)^2 + 0.5^2}} = -0.49 \quad (8)$$

Tal y como pasaba con el filtraje colaborativo basado en usuarios, donde acabábamos seleccionando N usuarios basándonos o bien en un cierto umbral de similitud o en un número prefijado de usuarios, en el filtraje colaborativo basado en ítems el procedimiento es análogo pero seleccionando los N ítems más similares.

Una vez tenemos seleccionados los N ítems más parecidos al ítem para el que queremos hacer la predicción de valoración y sus valores de similitud correspondientes, se calcula la predicción de valoración haciendo una ponderación de las valoraciones que ha hecho el usuario de estos ítems N:

$$pred(a,i) = \frac{\sum_{j \in N} sim(i,j) * a_j}{\sum_{j \in N} sim(i,j)} \quad (9)$$

donde a_j corresponde a la valoración del usuario a al ítem j .

Hay que tener en cuenta que los N ítems más parecidos que se consideren en la fórmula anterior tendrán que ser ítems para los que el usuario haya hecho una valoración. Siguiendo con el ejemplo de querer calcular la predicción de la valoración de la película 8 por parte del usuario 1, no sería necesario calcular la similitud de los ítems 1 y 4 respecto al ítem 8, puesto que no tenemos las valoraciones del usuario 1 para estas dos películas. Por lo tanto, necesitaríamos calcular la similitud para el resto de películas valoradas. Entre estas películas, también podemos descartar la 7, puesto que no tenemos ninguna otra valoración disponible. Por lo tanto, tendríamos que calcular las similitudes de las películas 2, 3, 5 y 6 con la película 8, y serían las siguientes:

$$sim(2,8) = \frac{(-2) * (-1) + (-1.25) * (-1.25)}{\sqrt{(-2)^2 + (-1.25)^2} * \sqrt{(-1)^2 + (-1.25)^2}} = 0.94 \quad (10)$$

$$sim(3,8) = \frac{0 * (-1) + (-1.5) * 0.5}{\sqrt{0^2 + (-1.5)^2} * \sqrt{(-1)^2 + 0.5^2}} = -0.45 \quad (11)$$

$$\text{sim}(5,8) = \frac{2 * (-1)}{\sqrt{2^2} * \sqrt{(-1)^2}} = -1.0 \quad (12)$$

$$\text{sim}(6,8) = \frac{1.5 * 0.5}{\sqrt{1.5^2} * \sqrt{0.5^2}} = 1.0 \quad (13)$$

Si escogemos los dos ítems más similares al 8, nos quedaremos con los 6 y 2, que tienen similitudes de 1.0 y 0.94, respectivamente. Así pues, la valoración de la película 8 por parte del usuario 1 sería la siguiente:

$$\text{pred}(1,8) = \frac{\text{sim}(6,8) * a_6 + \text{sim}(2,8) * a_2}{\text{sim}(6,8) + \text{sim}(2,8)} = \frac{1.0 * (-0.2) + 0.94 * (-0.2)}{1.0 + 0.94} = -0.2 \quad (14)$$

Hay que tener en cuenta que esta predicción es una predicción normalizada, de forma que si queremos tener el valor de predicción sin normalizar habrá que sumar la media de valoraciones que tenía el usuario:

$$\text{pred}(1,8) = -0.2 + \bar{a} = -0.2 + 3.2 = 3.0 \quad (15)$$

También habría sido posible utilizar la fórmula 14 con las valoraciones originales (sin normalizar) para obtener directamente este valor.

2.3 Filtraje colaborativo basado en factorización de matrices

En los subapartados anteriores hemos visto dos tipos diferentes de filtraje: por un lado, el filtraje basado en usuarios, en el que buscamos las filas de la matriz que se asemejen más al usuario deseado; por otro lado, el filtraje basado en ítems, en el que buscamos las columnas de la matriz que se asemejen más al ítem deseado. En este subapartado veremos una forma alternativa de explotar esta información basada en la factorización de matrices.

El año 2006, Netflix inició una competición llamada Netflix Prize que otorgaba un premio de un millón de dólares al primer participante que consiguiera una mejora del 10 % respecto a la mejor técnica de la que se disponía al inicio de la competición. No fue hasta el año 2009 que se logró esa meta y la competición finalizó. En esta competición, muchos participantes utilizaron métodos avanzados de factorización de matrices y demostraron que pueden ser muy útiles para construir sistemas recomendadores.

Concretamente, los métodos de factorización de matrices se utilizan en los sistemas recomendadores para encontrar un conjunto de características latentes a partir de patrones de las valoraciones. Las características latentes (ocultas) caracterizan tanto a los usuarios como a los ítems con vectores. En algunos

casos, estas características latentes pueden ser interpretables, como el género o tipo de una película, libro o canción, mientras que en otros casos las características latentes no son interpretables. Un cierto ítem i es recomendado a un usuario u si los vectores que representan al ítem i y al usuario u en el espacio de características latentes son parecidas.

Uno de los métodos utilizados en la factorización de matrices para sistemas recomendadores es la descomposición de valores singulares*. Recordemos que la descomposición de valores singulares de una matriz M consiste en encontrar las matrices U y V tales que:

$$M = U\Sigma V^T \quad (16)$$

donde U y V son las matrices de los vectores singulares y Σ es una matriz diagonal que contiene los valores singulares.

Llegados a este punto, podéis pensar qué relación hay entre la técnica SVD y los sistemas recomendadores. Si la matriz M representa la matriz de valoraciones, la respuesta es que la matriz U es una matriz que representa a los usuarios en un nuevo espacio de características, que es el mismo que aquel en el cual la matriz V representa los ítems. Así pues, del mismo modo que con técnicas anteriores podíamos encontrar la similitud entre usuarios (filtraje colaborativo basado en usuarios) o entre ítems (filtraje colaborativo basado en ítems), ahora, además, también nos es posible calcular la valoración que haría un usuario u de un ítem i simplemente haciendo el producto escalar (equivalente a la similitud coseno en caso de vectores unitarios) entre los vectores que caracterizan al usuario u y al ítem i en el espacio común de características latentes. La figura 4 muestra una descomposición SVD donde se pueden observar las representaciones de los usuarios y los ítems una vez hecha la descomposición SVD.

* En inglés, *singular value decomposition, SVD*.

Figura 4. Descomposición SVD aplicada a los sistemas recomendadores



Figura 4

Descomposición SVD aplicada a los sistemas recomendadores. Una vez hecha la descomposición SVD de la matriz de valoraciones, podemos calcular la valoración de cualquier usuario de un ítem concreto haciendo el producto escalar entre las matrices del usuario u y el ítem i .

Hay que añadir una consideración en lo referente al cálculo de la descomposición de la matriz de valoraciones en valores singulares. En el contexto de los sistemas recomendadores, esta matriz es una matriz dispersa (no disponemos de muchos valores de esta matriz), puesto que en caso contrario significaría que tendríamos las valoraciones de todos los ítems por parte de todos los

usuarios. Así pues, la búsqueda de las matrices U y V está basada en un problema de optimización que busca que el error entre la matriz resultante de la operación $U\Sigma V^T$ y la matriz original M sea mínimo para el conjunto de valoraciones de las que disponemos. Una vez disponemos de las matrices U y V podemos obtener todas las valoraciones de todos los ítems por parte de todos los usuarios haciendo la operación $U\Sigma V^T$ o, si simplemente estamos interesados en la valoración de un ítem i por parte de un usuario u , solo hacer el producto escalar de estos vectores en concreto.

2.4 Dificultades

Los sistemas recomendadores basados en técnicas de filtraje colaborativo presentan problemas cuando nos encontramos en casos de usuarios nuevos (no tenemos ninguna valoración previa de los ítems por parte de estos usuarios) o ítems nuevos (no tenemos ninguna valoración de estos ítems por parte de ningún usuario). A pesar de que hay estrategias para resolver estos problemas, quedan fuera de los objetivos de esta asignatura.

3. Sistemas recomendadores basados en contenido

A diferencia de las técnicas de filtraje colaborativo, que solo se basan en una matriz de valoraciones, los sistemas recomendadores basados en contenido utilizan características de los ítems y también un perfil que asigna qué importancia tiene cada una de estas características. Estas características pueden ser obtenidas manual o automáticamente. Continuando con el ejemplo inicial de las películas, las características podrían incluir el género (drama, ciencia-ficción, romántica, etc.) o los actores que participan. Por otro lado, el perfil de usuario indicaría qué importancia tiene cada una de estas características en sus preferencias. Este perfil puede ser obtenido de manera explícita formulando preguntas al usuario sobre sus intereses y preferencias o de manera automática analizando el comportamiento del usuario en la plataforma. La figura 5 muestra las diferencias entre el filtraje colaborativo y el basado en contenido.

Figura 5. Comparación entre el filtraje basado en contenido y el filtraje colaborativo

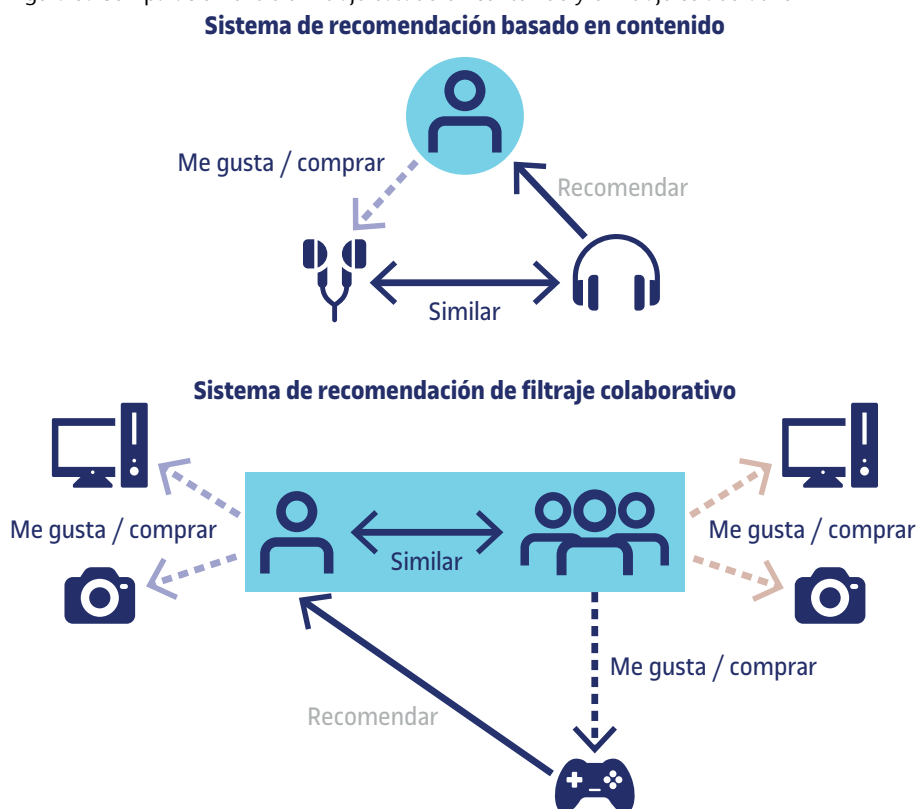


Figura 5

Comparación entre el filtraje colaborativo y el filtraje basado en contenido. En el primer caso, se analizan las valoraciones de los usuarios para encontrar nuevas recomendaciones. En el filtraje basado en contenido no está el factor social, y las comparaciones se hacen conociendo las características del producto y las preferencias del usuario. En este segundo caso, no nos hace falta ninguna valoración de un objeto nuevo antes de recomendarlo a un usuario.

Este tipo de sistemas recomendadores tiene dos ventajas principales respecto a los basados en filtraje colaborativo. La primera es que no requieren un gran volumen de usuarios y valoraciones para obtener recomendaciones bastante

buenas. La segunda es que si se dispone de las características de un ítem nuevo, no existe el problema de nuevos ítems sin valoraciones (presente en el filtraje colaborativo). Con las características del nuevo ítem hay suficiente para predecir la recomendación del ítem en cuestión al conjunto de usuarios. Como desventajas, destacan la necesidad de obtener y actualizar las descripciones de los ítems y la dificultad de caracterizar las características más subjetivas de los elementos. La subjetividad de elementos como la comodidad de una prenda de ropa o la facilidad de uso de un aparato electrónico no se pueden obtener automáticamente y son propensas a errores.

La mayoría de las técnicas de sistemas recomendadores basados en contenido se desarrollaron originariamente para recomendar documentos de texto, como artículos de noticias o páginas web. En estos dominios se podían extraer las características automáticamente del mismo contenido del documento o extraer descripciones textuales no estructuradas, como las palabras clave de un documento.

En los sistemas recomendadores basados en contenido, la predicción de la valoración que haría un usuario de un ítem se calcula comparando las diferentes características del ítem con las preferencias del perfil del usuario. Para cada una de las características, es posible que se tenga que utilizar una función de similitud diferente. Por ejemplo, si una de las características es el género de una película, utilizaremos una función binaria que devuelva 1 si el género del ítem está entre los géneros del perfil del usuario o que devuelva 0 en caso contrario. En cambio, si consideramos la característica de los actores que aparecen en la película, una posible función de similitud podría ser el cálculo del coeficiente de Dice, que se calcula:

$$\frac{2 \times |\text{actores}(i) \cap \text{actores}(u)|}{|\text{actores}(i)| + |\text{actores}(u)|} \quad (17)$$

donde $\text{actores}(i)$ hace referencia al conjunto de actores de la película i y $\text{actores}(u)$ hace referencia al conjunto de actores preferidos por el usuario u . Si, por ejemplo, la película tiene cinco actores, el usuario tiene un listado de diez actores preferidos y tres de estos actores salen en la película, entonces el coeficiente de Dice sería de $3/(10 + 5) = 0.2$, en contraste con otra película también con cinco actores pero solo un actor de los preferidos, que tendría un coeficiente de Dice de $1/(10 + 5) = 0.067$.

Una vez calculados los valores de similitud por cada una de las características, se calcula la valoración final por medio de una ponderación de los valores de similitud obtenidos previamente.

En cuanto a funciones de similitud basadas en contenido textual, una de las más habituales es la función *tf-idf*, donde *tf* hace referencia a *term frequency* e *idf* hace referencia a *inverse document frequency*. El término *tf* calcula la fre-

cuencia de cada palabra en un determinado documento, mientras que el término *idf* da más importancia a las palabras que son menos frecuentes en nuestro conjunto de documentos. Así pues, una palabra que aparece a menudo en un determinado documento pero es poco frecuente en otros documentos será más relevante que otra palabra que sea frecuente en otros documentos y/o aparezca poco en el documento en cuestión. También otras técnicas de preprocesamiento de textos suelen ser habituales en los sistemas recomendadores de documentos, como la eliminación de ciertas palabras (*stop words*) o la reducción de las palabras a sus raíces (el método *stemming*).

Hasta aquí hemos visto como podemos valorar un ítem conociendo el perfil del usuario. Para hacer estas recomendaciones no necesitamos ninguna valoración del usuario. Esto es especialmente útil cuando el usuario todavía no ha hecho ninguna valoración. Si, además, disponemos de valoraciones del usuario sobre determinados ítems, podemos hacer uso de esta información tal como se utiliza en el filtraje colaborativo. La diferencia es que ahora solo se tienen en cuenta las valoraciones hechas por el mismo usuario y no se consideran las valoraciones hechas por el resto de los usuarios de la comunidad.

Por lo tanto, una vez tenemos un histórico de valoraciones de un determinado usuario, como tenemos las características de todos los ítems (independientemente de que tengamos su valoración o no), por cada ítem no valorado podemos buscar los N ítems valorados previamente por el usuario más parecidos al ítem en cuestión y hacer una predicción de la valoración que haría el usuario de este ítem en función de la valoración hecha para los N ítems encontrados. El peso que se daría a cada ítem podría ser proporcional al valor de similitud que tienen los ítems, de forma que daríamos más importancia a los ítems más parecidos (y, por lo tanto, más relevantes). Tal como pasa con las técnicas de filtraje colaborativo, este conjunto de N ítems puede venir dado por un número prefijado de ítems o bien por un umbral de similitud mínimo.

4. Sistemas recomendadores basados en conocimiento

En los apartados anteriores hemos visto los sistemas de recomendaciones más utilizados. Tanto los sistemas recomendadores de filtraje colaborativo como los basados en contenido tienen como ventaja el bajo coste de adquisición y mantenimiento de los datos necesarios para hacer recomendaciones.

En este apartado veremos qué podemos hacer cuando las condiciones para utilizar los sistemas estudiados hasta ahora no se cumplen. Cuando disponemos de muy pocas valoraciones o en casos en que la información pasada cambia rápidamente, necesitamos una nueva tipología de sistemas recomendadores. Por ejemplo, si queremos recomendar la compra de un electrodoméstico o de una casa, no dispondremos de un historial de compras similares, puesto que no hacemos este tipo de compras con mucha frecuencia. En este caso, un sistema de filtraje colaborativo no funcionará bien debido al bajo número de valoraciones disponible. Además, el tiempo pasado entre las compras anteriores tiene una gran importancia. Por ejemplo, las clasificaciones de hace cinco años para electrodomésticos serán inadecuadas, puesto que evolucionan muy rápidamente. En dominios de productos más complejos, como los coches, los clientes a menudo quieren definir sus requisitos explícitamente, por ejemplo, «estoy dispuesto a pagar como máximo X y el color debe ser rojo».

Los sistemas de recomendación basados en conocimiento nos ayudan a afrontar los retos mencionados. Permiten formular requisitos de una manera que no es posible en sistemas recomendadores colaborativos ni en los basados en el contenido. La ventaja de estos sistemas es que las recomendaciones se calculan independientemente de las valoraciones individuales de los usuarios. Las recomendaciones en los sistemas basados en conocimiento se calculan únicamente a partir de las relaciones entre los requerimientos del usuario y las características de los objetos.

Tabla 4. Características de lavadoras (ejemplo)

Ítem	Precio	Capacidad	Nº de programas	Carga frontal	Eficiencia
p1	499	8	20	no	sí
p2	399	9	18	no	sí
p3	259	7	12	no	no
p4	350	8	10	sí	sí
p5	199	9	3	no	no
p6	387	7	12	no	sí
p7	350	6	14	sí	sí

En general, los sistemas basados en el conocimiento necesitan conocimientos detallados sobre las características de los ítems. En la tabla 4 se muestra un ejemplo de las características de un catálogo de lavadoras. El problema de recomendación consiste en seleccionar elementos de este catálogo que coincidan con las necesidades, preferencias o requisitos del usuario. Podemos expresar los requisitos del usuario en intervalos de valores deseados para una característica de un artículo, como por ejemplo «el precio debe ser inferior a 400 €» o según la funcionalidad deseada, como «la lavadora debería ser eficiente».

El proceso de recomendación basado en el conocimiento es muy interactivo, puesto que son sistemas que guían al usuario de manera personalizada hasta encontrar los productos interesantes dentro de un amplio espacio de posibles opciones. El sistema habitual de funcionamiento en estos sistemas acostumbra a ser el siguiente: el usuario tiene que especificar los requisitos y el sistema intenta identificar una solución. Si no se puede encontrar ninguna solución, el usuario tiene que cambiar los requisitos. Este proceso se repite hasta que el usuario encuentra un producto que le satisface. A veces el sistema también proporciona explicaciones sobre por qué se han seleccionado los elementos recomendados.

El flujo de interacción general de un recomendador basado en restricciones se puede resumir con los pasos siguientes:

- 1) El usuario especifica sus preferencias iniciales, por ejemplo, con un formulario basado en web. Estos formularios pueden ser idénticos para todos los usuarios o personalizados para la situación específica del usuario actual.
- 2) Cuando se ha recopilado suficiente información sobre los requisitos y las preferencias del usuario, se presenta un conjunto de elementos coincidentes.
- 3) El usuario puede revisar sus requisitos, por ejemplo, para ver soluciones alternativas o reducir el número de elementos coincidentes.

A pesar de que, en primer lugar, este esquema general de interacción con el usuario parece bastante sencillo, normalmente se requieren aplicaciones prácticas para implementar patrones de interacción más elaborados y apoyar al usuario final en el proceso de recomendación. Pensad, por ejemplo, en situaciones en que ninguno de los elementos del catálogo cumple todos los requisitos del usuario. En estas situaciones un recomendador interactivo tendría que dar apoyo inteligente al usuario final para resolver el problema y proponer de forma proactiva algunas alternativas de acción.

Existen diferentes técnicas para dar apoyo a los usuarios en la interacción con aplicaciones de recomendaciones basadas en restricciones. Estas técnicas ayudan a mejorar la usabilidad de estas aplicaciones. Son mecanismos muy importantes para conseguir una mayor aceptación del usuario en cuanto a la

confianza, la satisfacción con el proceso de recomendación y también para mejorar la calidad de las recomendaciones proporcionadas. A continuación listamos las principales:

- **Proponer valores por defecto:** los valores predeterminados son un medio importante para dar apoyo a los clientes en el proceso de especificación de requisitos, especialmente en situaciones en que no saben qué opción seleccionar o simplemente no conocen los detalles técnicos de los productos.
- **Seleccionar la siguiente pregunta:** en el proceso de especificación de requisitos, se puede proporcionar un registro de interacciones y el mecanismo por defecto para identificar propiedades. Este orden de peticiones guía al usuario durante la sesión de recomendación proporcionando elecciones interesantes.
- **Tratamiento de requisitos imposibles y conjuntos de resultados vacíos:** muchos sistemas de recomendación no son capaces de proponer una salida cuando no pueden encontrar ninguna solución. En estos casos una opción para ayudar al usuario es relajar de manera incremental y automática las limitaciones del problema de recomendación hasta que se haya encontrado la solución correspondiente.
- **Clasificación de los elementos:** cada vez que se proporciona una recomendación al usuario, es importante clasificar los artículos según la utilidad para el cliente. Los usuarios acostumbran a mirar y a seleccionar solo los primeros artículos recomendados. Si el sistema es capaz de ordenar exitosamente las recomendaciones, el cliente tendrá una mayor confianza en la aplicación. Esta mejor clasificación aumentará significativamente la voluntad de comprar los productos recomendados.

Podemos clasificar los sistemas recomendadores basados en conocimiento en dos familias principales según cómo utilizan el conocimiento proporcionado. En primer lugar, los recomendadores basados en *restricciones* se basan en un conjunto de reglas de recomendación definidas de forma explícita, buscando un conjunto de elementos que cumplan las reglas de recomendación. Por otro lado, tenemos los recomendadores basados en *casos*, que utilizan métricas de semejanza para recuperar elementos que sean similares a los requisitos de los clientes.

4.1 Sistemas recomendadores basados en restricciones

Un problema de recomendación basado en restricciones se basa en un conjunto de condiciones estrictas especificadas por el cliente sobre el producto que quiere. Volviendo al ejemplo de las lavadoras de la tabla 4, unas posibles restricciones serían que la lavadora tuviera como mínimo quince programas

o que la carga fuera frontal. En general, este tipo de problema se puede representar como un problema de satisfacción de restricciones* o como un problema de consultas conjuntivas.** En este subapartado veremos su formulación como un problema de consultas conjuntivas. Un problema de consultas conjuntivas consiste en una serie de consultas que se tienen que cumplir todas a la vez, lo que equivale a una serie de condiciones con la conjunción lógica (AND).

* En inglés, *constraint satisfaction problem*

** En inglés, *conjunctive queries*

Para recuperar los elementos basados en restricciones para un catálogo determinado, como en el ejemplo de la tabla 4, podemos ver el problema de selección de elementos como una tarea de filtraje de datos. En este caso lo que queremos es construir una consulta de base de datos conjuntiva que se ejecute contra el catálogo de elementos.

Por ejemplo, si definimos el conjunto de elementos disponibles $P = \{p1, p2, p3, p4, p5, p6, p7\}$ de la tabla 4, podemos definir la consulta o selección σ con los criterios de selección [*capacidad* ≥ 8 AND *precio* < 400] sobre la base de datos.

Variando las restricciones impuestas durante la selección σ , obtendremos diferentes conjuntos de resultados. Estas consultas se derivan directamente de las condiciones de filtro que definen la relación entre los requisitos del cliente y las propiedades del artículo correspondientes. Una opción para facilitar la elección al cliente es proporcionarle peticiones más intuitivas. Por ejemplo, podríamos definir una restricción que fuera que los clientes quieren tener una lavadora con una gran capacidad (requisito = gran capacidad). Esta condición implicaría poner una condición del filtro = *capacidad* ≥ 8 indicando que si los clientes quieren tener la posibilidad de lavar mucha ropa, la capacidad de la lavadora (kg) tiene que ser > 8.0 . De este modo, el cliente no necesita tener unos conocimientos tan técnicos sobre el producto. Así pues, si un cliente define el uso del requisito = gran capacidad, la condición de filtro correspondiente (*capacidad* ≥ 8) está activa y se integrará en una consulta conjuntiva correspondiente. La existencia de una recomendación para un determinado requisito y un surtido de productos P se hace consultando P con las condiciones derivadas (en el ejemplo, *capacidad* ≥ 8).

4.2 Sistemas recomendadores basados en casos

Los sistemas de recomendaciones basados en casos*** utilizan métricas de semejanza para recuperar elementos de un catálogo. Los ítems se recomiendan en función de medidas de semejanza que describen hasta qué punto las propiedades de los ítems coinciden con los requisitos determinados por el usuario. Podemos definir la distancia de similitud de un ítem p con los requisitos $r \in \text{REQ}$ tal como se muestra en la fórmula 18. En este contexto, $\text{sim}(p, r)$ expresa, para cada valor de atributo de elemento $\phi_r(p)$, su distancia al requisito

*** En inglés, *Case-Based recommender systems*

del cliente $r \in \text{REQ}$. Por ejemplo, un usuario puede definir como requisito r “ $6.0 < \text{capacidad} < 8.0$ ”. Para calcular la $\text{sim}(p, r)$, compararemos esta capacidad deseada por el usuario con los valores de los ítems $\phi_r(p)$ (la columna “Capacidad” de la tabla 4). Adicionalmente, se pueden definir pesos w_r para dar diferente importancia a cada requisito r .

$$\text{similitud}(p, \text{REQ}) = \frac{\sum_{r \in \text{REQ}} w_r * \text{sim}(p, r)}{\sum_{r \in \text{REQ}} w_r} \quad (18)$$

La fórmula anterior nos define la similitud global entre cada ítem y los requisitos del usuario. Pero hay que definir, por cada requisito r , cuál es la similitud con el producto p , es decir, $\text{sim}(p, r)$. Dependiendo de cada característica, queremos definir una similitud diferente. En los escenarios del mundo real, hay propiedades que un cliente querría maximizar, por ejemplo, la capacidad de una lavadora. También hay propiedades que los clientes quieren minimizar, por ejemplo, el precio del producto. En el primer caso hablamos de propiedades *más es mejor*; en el segundo caso las propiedades correspondientes se denotan con *menos es mejor*. Para tener en cuenta estas propiedades básicas en nuestros cálculos de semejanza, introducimos las fórmulas siguientes para calcular la semejanza con cada condición.

En el caso de *más es mejor* la similitud entre p y r se define:

$$\text{sim}(p, r) = \frac{\phi_r(p) - \min(r)}{\max(r) - \min(r)} \quad (19)$$

En el caso de *menos es mejor* la similitud entre p y r queda definida:

$$\text{sim}(p, r) = \frac{\max(r) - \phi_r(p)}{\max(r) - \min(r)} \quad (20)$$

Finalmente, hay situaciones en que la similitud se tiene que basar únicamente en la distancia a los requisitos definidos originariamente. Por ejemplo, si el usuario requiere un tamaño de televisor determinado, el tamaño mayor no representará una solución óptima. Para estos casos tenemos que introducir un tercer tipo de función de semejanza local:

$$\text{sim}(p, r) = 1 - \frac{|\phi_r(p) - r|}{\max(r) - \min(r)} \quad (21)$$

De manera similar a los recomendadores basados en restricciones, las primeras versiones de recomendadores basados en casos siguieron un enfoque puro basado en las consultas. Los usuarios tenían que especificar sus requisitos hasta encontrar un elemento objetivo (un elemento que se adapta a los deseos y necesidades del usuario). Este tipo de proceso de solicitud de requisitos conduce a sesiones de recomendación tediosas, puesto que las propiedades interdependientes de los elementos requieren un conocimiento sustancial del dominio

para funcionar bien. Una evolución para solucionar esta problemática son los enfoques basados en la navegación para recuperar elementos. En este caso los usuarios navegan (quizás sin saber qué buscan) por el espacio del elemento con el objetivo de encontrar alternativas útiles hasta encontrar un elemento adecuado. Criticar* es una manera eficaz de apoyar estas búsquedas. La idea de criticar parte de tener un elemento recomendado al que le falta alguna característica. El usuario puede especificar una solicitud de cambio en forma de objetivos que el elemento que se considera actualmente no satisface. Si, por ejemplo, el precio de la lavadora que se muestra actualmente es demasiado alto, se puede activar una crítica más barata; si el usuario quiere tener una lavadora con más capacidad, se puede seleccionar una crítica en este sentido.

* En inglés, *critiquing*

5. Sistemas recomendadores híbridos

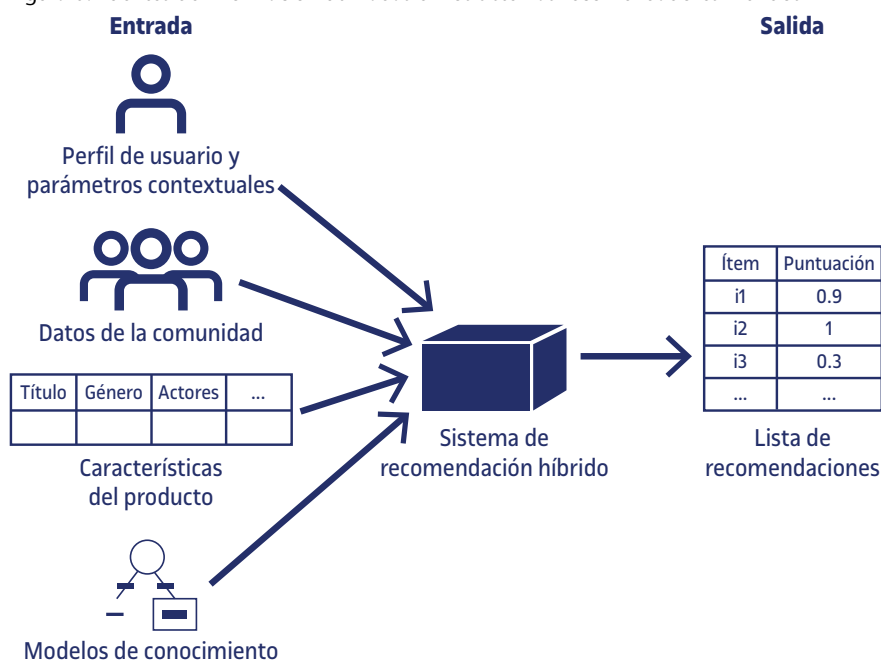
Los tres enfoques de recomendación más destacados discutidos en los apartados anteriores explotan diferentes fuentes de información y siguen diferentes paradigmas para hacer recomendaciones. A pesar de que producen resultados que se consideran personalizados en función de los intereses asumidos por sus destinatarios, funcionan con diferentes grados de éxito en diferentes dominios de aplicación. Cada uno de los enfoques básicos tiene sus ventajas y sus inconvenientes.

Si tratamos un sistema de recomendación como una caja negra que transforma los datos de entrada en una lista clasificada de elementos como salida podemos disponer de diferentes tipos de información:

- Los modelos de usuario y la información contextual
- Los datos de la comunidad
- Los datos de los productos
- Los modelos de conocimiento

Esta variedad de información de entrada está representada en la figura 6.

Figura 6. Fuentes de información utilizadas en los sistemas recomendadores híbridos



Ninguno de los enfoques básicos planteados hasta ahora es capaz de explotar todos estos tipos de información plenamente. En consecuencia, podemos

intentar construir sistemas híbridos que combinen los puntos fuertes de diferentes algoritmos y modelos para superar algunas de las carencias y problemas mencionados. Los sistemas híbridos de recomendación son enfoques que combinan varias implementaciones de algoritmos o componentes de recomendación.

Los sistemas recomendadores híbridos usan los tres paradigmas básicos de recomendación: colaborativos, basados en el contenido y basados en el conocimiento. Estos esquemas híbridos permiten combinar las características positivas de los paradigmas básicos para proporcionar recomendaciones mejores. El principio de colaboración considera que hay clústeres de usuarios que se comportan de una manera similar y que tienen necesidades y preferencias comparables. El paradigma basado en el contenido sigue un enfoque «más de lo mismo» al recomendar artículos que son similares a los que le gustaban al usuario en el pasado, pero la recomendación basada en el conocimiento considera una fuente adicional de información: conocimiento explícito de personalización. Aprovechándonos de esta diversidad de comportamiento, podremos generar un recomendador híbrido incorporando las mejores características de cada paradigma.

En consecuencia, la elección del paradigma de recomendación determina el tipo de datos de entrada necesarios. Tal como hemos definido antes, existen cuatro tipos de datos posibles. El modelo de usuario y los parámetros contextuales representan el usuario y la situación específica en que está actualmente: por ejemplo, los elementos que el usuario ha valorado hasta ahora; las respuestas que el usuario ha dado en un diálogo de solicitud de requisitos; información demográfica como dirección, edad o educación, y parámetros contextuales como la temporada del año, las personas que acompañarán al usuario cuando compre o consuma el elemento (por ejemplo, ver una película) o la ubicación actual del usuario.

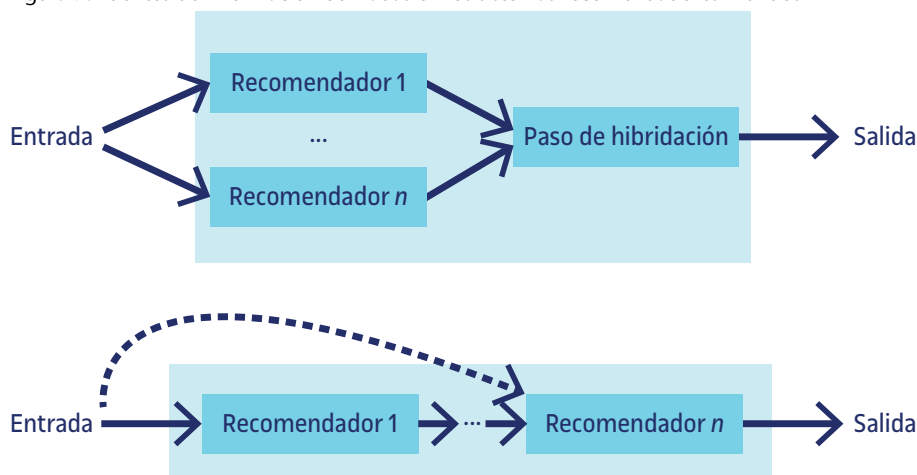
Se puede suponer que todos estos datos específicos del usuario y de la situación se almacenan en el perfil del usuario. En consecuencia, todos los paradigmas de recomendaciones requieren acceder a este modelo de usuario para personalizar las recomendaciones. Aun así, en función del dominio de la aplicación y del escenario de uso, solo tendremos partes del modelo de usuario disponibles. Por lo tanto, no todas las variantes de hibridación serán posibles o recomendables en todos los campos de aplicación.

Podemos clasificar los métodos de hibridación según la manera en que combinamos dos o más algoritmos. Los componentes de recomendación pueden funcionar en paralelo antes de combinar sus resultados, o bien se pueden conectar dos o más sistemas de recomendación individuales en una arquitectura de canalización en que la salida de un recomendador sirva de entrada para el siguiente. Según cómo combinamos los sistemas de recomendación individuales, disponemos de los diseños siguientes:

- **Híbrido monolítico:** denota un diseño de hibridación que incorpora aspectos de varias estrategias de recomendación en una implementación de un único algoritmo. Los híbridos monolíticos consisten en un componente de recomendación único que integra muchos enfoques preprocesando y combinando varias fuentes de conocimiento. Por lo tanto, la hibridación se consigue mediante una modificación integrada del comportamiento del algoritmo para explotar diferentes tipos de datos de entrada.
- **Paralelizado:** requiere al menos dos implementaciones de recomendaciones separadas, que se combinan en paralelo. Funcionan independientemente los unos de los otros y producen listas de recomendaciones separadas. En un siguiente paso de hibridación, su producción se combina en un conjunto final de recomendaciones.
- **Canalizado:** implementa un proceso escalonado en el que varias técnicas se construyen secuencialmente entre ellas antes de que la última produzca recomendaciones para el usuario. Las variantes híbridas canalizadas se diferencian principalmente según el tipo de producción que producen para la siguiente etapa. Un componente puede preprocesar los datos de entrada para construir un modelo que sea explotado en la fase posterior o dar una lista de recomendaciones que serán mejoradas para la etapa posterior.

La figura 7 muestra dos tipos de sistemas recomendadores híbridos. Arriba podemos ver un sistema paralelizado donde diferentes sistemas recomendadores son combinados en una etapa final. Debajo observamos un sistema canalizado donde un sistema recomendador sirve como entrada de la siguiente etapa.

Figura 7. Fuentes de información utilizadas en los sistemas recomendadores híbridos



6. Evaluación de sistemas recomendadores

Después de haber revisado las diferentes técnicas existentes para diseñar un sistema recomendador, la pregunta que nos hacemos en este apartado es cómo se evalúan estas técnicas. ¿Cuáles son las métricas de evaluación que se suelen considerar para decidir si un sistema recomendador es mejor que otro? Del mismo modo que en módulos anteriores hemos analizado las diferentes métricas existentes en problemas de aprendizaje automático como el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo, también lo veremos para los sistemas recomendadores.

La evaluación de los sistemas recomendadores se puede dividir principalmente en dos familias. Por un lado, está la evaluación de sistemas recomendadores que utilizan bases de datos históricas. Por otro lado, está la evaluación de sistemas recomendadores que usan estudios con usuarios reales. Desgraciadamente, ambos tipos de evaluaciones no se pueden comparar directamente la una con la otra.

6.1 Evaluación basada en datos históricos

La evaluación de sistemas recomendadores utilizando bases de datos históricas se basa en el uso de bases de datos en las que se dispone de las valoraciones que han hecho varios usuarios de varios ítems. Este tipo de evaluación es análoga a la que se suele hacer en los problemas de aprendizaje supervisado, en los que tenemos un conjunto de datos que utilizamos como entrenamiento y otro que usamos para validar el modelo. En los sistemas recomendadores procederíamos de forma parecida, puesto que el conjunto de valoraciones de los ítems por parte de los usuarios de los que disponemos lo dividiríamos en dos conjuntos: uno que utilizaríamos para entrenar el modelo y otro para evaluarlo.

En función de la naturaleza del problema, el tipo de métricas de evaluación que utilizaríamos sería el propio de un problema de clasificación o de un problema de regresión. En el caso de que quisiéramos determinar si recomendaríamos o no un determinado ítem a un usuario, utilizaríamos las métricas propias de los problemas de clasificación, como precisión, exhaustividad* o exactitud.** En cambio, si lo que pretendemos es que el sistema recomendador haga una predicción de la valoración que haría un usuario de un ítem (por ejemplo, de una película en una escala del 1 al 5), usaríamos las métricas

* En inglés, *recall*

** En inglés, *accuracy*

propias de los problemas de regresión, como el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE) o el error medio absoluto (MAE).

En el caso de los sistemas recomendadores, también existe otro tipo de métricas de evaluación, que son las basadas en comparación con listas ordenadas o *rankings*. Si nos encontramos en el caso de valoraciones numéricas (no binarias), para un cierto usuario podemos ordenar los ítems de mayor a menor preferencia según la valoración del usuario. Si tenemos esta ordenación realizada tanto por los valores reales de los datos históricos como por los valores predichos, entonces se pueden utilizar métricas de comparación de listas ordenadas como la MAP (Mean Average Precision) o la MRR (Mean Reciprocal Rank). No obstante, la definición de estas métricas de evaluación queda fuera del alcance de este curso.

Una de las ventajas que tiene la evaluación basada en datos históricos es que permite usar bases de datos públicas (MovieLens o Netflix, entre otras) en las que es posible hacer una comparación de varios sistemas recomendadores sobre la misma base de datos.

Por el contrario, este tipo de evaluación también tiene varios inconvenientes. Por un lado, las valoraciones que se suelen tener de los ítems por parte del conjunto de usuarios suele ser bastante dispersa, es decir, de cada usuario solemos tener las valoraciones de un porcentaje de los ítems bastante bajo. Esto hace que no podamos evaluar la bondad de un sistema recomendador para todos los ítems de los que no disponemos de las valoraciones. Por otro lado, las bases de datos disponibles para sistemas recomendadores puede sesgar el diseño de estos sistemas hacia estas bases de datos concretas.

6.2 Evaluación basada en estudios con usuarios reales

A diferencia de la evaluación previa basada en datos históricos, la evaluación de sistemas recomendadores basada en estudios con usuarios reales consiste en pedir a un usuario que decida de manera instantánea si le gusta o no el ítem propuesto por el sistema recomendador. En este caso no es necesario que el usuario haya hecho una valoración previa del ítem en cuestión, puesto que el sistema recomendador se evalúa en situaciones con usuarios reales.

En este tipo de evaluación podemos medir la tasa de acierto que ha tenido el sistema recomendador en la elección de los ítems escogidos para proponer al usuario basado en el retorno positivo y negativo que haya dado. Por ejemplo, podemos pensar en una plataforma de contenidos audiovisuales como Netflix, donde se recomienda una película a un usuario, y si la ve o la añade a su lista lo consideramos un acierto, mientras que en caso contrario sería un error. También se puede dar un ejemplo en que el sistema, en lugar de propo-

ner un único ítem, propone varios, y el usuario decide cuáles de estos ítems propuestos se adecúan a sus preferencias (aciertos) y cuáles no (errores).

Por el contrario, esta forma de evaluación también tiene sus inconvenientes. Por un lado, no es posible evaluar si al usuario le habrían gustado otros ítems que no le han sido propuestos por el sistema recomendador, puesto que solo tenemos el retorno sobre el ítem propuesto. Por otro lado, los sistemas recomendadores evaluados de este modo no permiten una comparación justa con otros sistemas recomendadores, puesto que probablemente se habrán analizado con bases de datos o usuarios diferentes.

Bibliografía

Jannach, D.; Zanker, M.; Felfernig, A.; Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511763113