



UNIVERSIDAD DE JAÉN

Un nuevo modelo ponderado para Sistemas de Recomendación Basados en Contenido con medidas de contingencia y entropía

Trabajo Tutelado de Iniciación a la Investigación

Alumno: Jorge Castro Gallardo

Tutor: Dr. Luis Martínez López

Cotutor: Dr. Manuel J. Barranco García

Curso 2011/2012

Jaén, Septiembre 2012



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Dr. D. Luís Martínez López, director del Trabajo Tutelado de Iniciación a la Investigación titulado: Un nuevo modelo ponderado para Sistemas de Recomendación Basados en Contenido con medidas de contingencia y entropía, que presenta D. Jorge Castro Gallardo, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2012

El alumno:

El director:

D. Jorge Castro Gallardo

Dr. D. Luís Martínez López

Índice general

Índice de figuras	1
1. Introducción	3
1.1. Contexto	3
1.1.1. Comercio Electrónico	4
1.1.2. Personalización	7
1.1.3. Sistemas de Recomendación	8
1.2. Propósito y Objetivos	10
1.3. Estructura de la memoria	11
2. Sistemas de Recomendación	13
2.1. Introducción	13
2.2. Historia	17
2.3. Técnicas de Recomendación	20
2.3.1. Sistemas de Recomendación Demográficos	21

2.3.2.	Sistemas de Recomendación Basados en Contenido	22
2.3.3.	Sistemas de Recomendación Colaborativos	26
2.3.4.	Sistemas de Recomendación Basados en Conocimiento	34
2.3.5.	Sistemas de Recomendación Basados en Utilidad	39
2.3.6.	Sistemas de Recomendación Híbridos	41
2.4.	Evaluación de los Sistemas de Recomendación	48
2.4.1.	Protocolos de evaluación	51
2.4.2.	Medidas de evaluación	55
2.4.3.	Predicción de valoraciones	56
2.4.4.	Clasificación binaria de productos	57
2.4.5.	<i>Ranking</i> de productos	60
2.4.6.	Medidas de eficiencia	62
2.5.	Temas emergentes	63
3.	Sistemas de Recomendación Basados en Contenido Con Modelado en Es-	
	pacios Vectoriales	69
3.1.	Modelo Booleano Básico	70
3.2.	Modelo Booleano con Ponderación Basada en TF-IDF	74
4.	Modelo Ponderado con Medidas de Contingencia y Entropía	79
4.1.	Problemas del modelado Booleano	80

4.2. Modelo Ponderado con medidas de Contingencia y Entropía	81
4.2.1. Herramientas para el cálculo de la ponderación	83
4.2.1.1. Medida de disimilitud inter-usuario	83
4.2.1.2. Medidas de similitud intra-usuario	84
4.2.2. Configuración del sistema	88
4.2.3. Descripción detallada del algoritmo	89
5. Caso de Estudio	97
5.1. Exposición del proceso de experimentación.	97
5.1.1. Descripción de los conjuntos de datos	98
5.1.2. Configuración del experimento	100
5.2. Resultados	101
6. Conclusiones	105
Bibliografía	109

Índice de figuras

2.1. Tareas básicas de un Sistema de Recomendación	16
2.2. Configuración de una red neuronal	25
2.3. Proceso recomendación en el filtrado colaborativo	27
2.4. Cálculo de la similitud entre productos	29
2.5. Proceso de resolución en Razonamiento Basado en Casos	36
2.6. Recomendación de productos basada en utilidad	40
2.7. Esquema general de hibridación por ponderación	42
2.8. Esquema general de hibridación por conmutación	44
2.9. Esquema general de hibridación en cascada	45
2.10. Esquema general de hibridación por combinación de características	47
2.11. Esquema general de hibridación multi nivel	47
2.12. Protocolo de evaluación Validación Cruzada	53
2.13. Curva ROC de distintos Sistemas de Recomendación	61
2.14. Visualización de las recomendaciones	65

4.1. Ponderación de características basada en la entropía y medidas de contingencia	82
4.2. Ponderación de características basada en la entropía y medidas de contingencia	90
4.3. Algoritmo de la propuesta de recomendación basada en contenido	91
5.1. Precisión de los algoritmos en el experimento	101
5.2. Recall de los algoritmos en el experimento	102
5.3. F1-medida de los algoritmos en el experimento	103
5.4. Tiempos de construcción del modelo de los métodos	104
5.5. Tiempo de respuesta de los métodos	104

Introducción

La presente documentación constituye la memoria del trabajo de investigación titulado *Un nuevo modelo ponderado para Sistemas de Recomendación Basados en Contenido con medidas de contingencia y entropía* y que sirve como Trabajo Tutelado de Iniciación a la Investigación en la etapa formativa del *Programa de Doctorado en Ingeniería y Arquitectura*, impartido en la Universidad de Jaén. La memoria comienza con una introducción al contexto, el Comercio Electrónico y la Personalización, en que se ubica esta investigación. Tras esto se revisan algunos trabajos relacionados con esta investigación, se fijan el propósito y objetivos de la misma y se define la estructura que se seguirá en el resto de la memoria.

1.1. Contexto

En la actualidad, existe gran cantidad de información sobre productos, contenidos y servicios debido a la disponibilidad de la misma mediante *Internet*. Esto puede hacer que un usuario se sienta desbordado al intentar discriminar qué producto, contenido o servicio cubre sus necesidades y/o gustos de una forma óptima [12]. Para ayudar al usuario en su búsqueda, surgieron los Sistemas de Recomendación, que son “sistemas que producen recomendaciones personalizadas como salida o tienen el efecto de guiar al usuario de una forma personalizada a productos interesantes o útiles de entre una gran cantidad de productos disponibles” [23].

Los Sistemas de Recomendación aportan un valor añadido en los procesos de comercialización que beneficia tanto a las tiendas de Comercio Electrónico que los incorporen como a los usuarios que acceden a las mismas. Para la tienda de Comercio Electrónico, un Sistema de Recomendación supone una potente herramienta de fidelización de los usuarios, se incrementa el volumen de ventas del sitio y se obtiene información sobre los usuarios sin que tengan que adquirir los productos o servicios que se ofrecen en el mismo [12]. Para el usuario, un Sistema de Recomendación hace que la búsqueda de productos interesantes sea más sencilla, reduciendo el número de opciones que tiene que revisar y mejorando su experiencia de uso en el sitio que lo incorpore. Esta característica es importante, ya que muchos usuarios eligen una tienda de Comercio Electrónico teniendo en cuenta la facilidad del uso de la misma y un Sistema de Recomendación le facilita el trabajo.

Los Sistemas de Recomendación pueden ubicarse en el ámbito del Comercio Electrónico, más concretamente dentro de los procesos de Personalización que se utilizan para la mejora de los procesos de comercialización en las tiendas electrónicas (en inglés, *e-shops*).

1.1.1. Comercio Electrónico

El Comercio Electrónico (CE) [52] se puede definir como “la compra y venta de productos o de servicios a través de medios electrónicos, tales como Internet y otras redes de comunicación”.

El primer antecedente del CE apareció a principios de los años 20: la venta por correspondencia. Para ello, las empresas de comercialización de productos distribuían un catálogo de productos a vender en los que se daba una imagen descriptiva de cada uno. La venta por catálogo permite que el consumidor pueda consultar en la tranquilidad de su hogar, y sin la presión de un vendedor, diversos artículos que se ofertan y tomar una decisión de compra. La facturación de este tipo de venta se incrementó aún más en los años 50 con la aparición de las tarjetas de crédito.

A mediados de la década de los 80, con la aparición del televisor, surgió una nueva forma de comercio: la televenta. En esta modalidad, los productos son anunciados en distintos medios (televisión, radio, etc) y el consumidor los solicita por teléfono y los paga mediante tarjeta de crédito.

Un tipo temprano de CE de gran importancia es el Intercambio Electrónico de Datos (en inglés, *Electronic Data Interchange*). El EDI es un estándar que permite el intercambio de documentos en formato electrónico entre empresas. Los documentos que más frecuentemente se intercambian, utilizando un sistema de EDI, son facturas y pedidos (por orden de importancia), pero también se utiliza para intercambiar catálogos, inventarios, etc. Esto permite a las empresas que lo utilizan reducir costes de gestión administrativa.

El surgimiento del CE, tal y como lo conocemos hoy en día, se debe al incremento del uso de la *World Wide Web* debido a la aparición del primer navegador Web (Tim Berners-Lee) en 1990. La fundación de empresas como la librería electrónica Amazon (en 1994), el portal de subastas eBay (en 1995), el sistema de pago PayPal (en 1998) y su índice de beneficios constatan que el Comercio Electrónico es una actividad positiva tanto para las empresas como para los consumidores.

Existen distintas formas de CE, dependiendo del tipo de actores que lleven a cabo las transacciones. A continuación se muestran las distintas formas de CE que existen [123]:

- ***Business to Consumer (B2C)***. Es la modalidad de CE en la que las operaciones comerciales son entre una empresa y consumidor (persona física o usuario final).
 - ***Business to Business (B2B)***. Es una modalidad de CE donde las operaciones comerciales son entre empresas y no con usuarios finales. Algunos ejemplos son las relaciones comerciales con proveedores o el intercambio de datos. En esta modalidad también se incluyen las transacciones realizadas por agrupaciones de compradores, que permiten a las empresas o consumidores unirse para realizar un intercambio comercial, lo que les permite negociar en mejores condiciones.
-

- ***Business to Government (B2G)***. Se da cuando las empresas ofrecen productos o servicios a las administraciones públicas, utilizando para ello las Tecnologías de la Información y la Comunicación. Un ejemplo de esta modalidad de CE son las relaciones comerciales entre las instituciones oficiales (ayuntamientos, diputaciones, etc) y sus proveedores.

El CE beneficia tanto a las empresas que operan en medios electrónicos como a los consumidores que utilizan esta vía. Los beneficios para las empresas son la reducción de los costes y mejora de los procesos de distribución, la posibilidad de realizar comunicaciones comerciales utilizando medios electrónicos, la mayor facilidad para entrar en nuevos mercados y la fidelización de los clientes.

En este trabajo nos centramos en los procesos de mejora del CE en su modalidad B2C. Existen diversas técnicas que se pueden aplicar para la mejora de los procesos de CE en esta modalidad, como son:

- ***E-marketing***: es la parte del CE que comprende la gestión de contenidos, las relaciones públicas, el servicio al cliente y las ventas. Comprende técnicas como el posicionamiento, boletines electrónicos, publicidad mediante banners, *SMS*, etc.
- **Personalización**: en los sitios *Webs* centrados en B2C se pueden aplicar técnicas de personalización como los Sistemas de Recomendación (ver Capítulo 2), que se utilizan para sugerir productos específicos a cada usuario atendiendo a sus necesidades y/o intereses [52].

Esta investigación se centra en el estudio y mejora de los Sistemas de Recomendación, que pertenecen a los procesos de Personalización en el CE, por lo que en el siguiente apartado se explican con más detalle.

1.1.2. Personalización

Debido a la cantidad de información que los usuarios tienen a su disposición mediante el uso de *Internet* pueden acceder en cualquier momento a información detallada sobre productos y servicios de cualquier tipo. Antes de *Internet*, el usuario solo conocía los productos que llegaban a él, es decir, que se presentaban mediante los anuncios, estando presentes en las tiendas que compraba habitualmente, etc. Ahora el usuario no sólo conoce estos productos sino que puede realizar una búsqueda en *Internet* y tener acceso a productos que de otra manera no hubiera conocido. El entorno actual permite también que el usuario no sólo conozca más productos sino que pueda tener más información (incluso procedente de fuentes distintas de la empresa comercializadora) sobre los productos que le interesan para comparar con otros y tomar una decisión con mayor información sobre las alternativas disponibles.

Esta *sobrecarga de información* hace que el usuario pueda sentirse desbordado, ya que evaluar todas las alternativas puede resultar una tarea tediosa. Para ello se utilizan técnicas de personalización, que tratan de reducir el espacio de alternativas seleccionando aquellas que más se adecuan a los gustos de cada usuario, a sus necesidades específicas o a las restricciones que tiene en un momento dado.

La Personalización [47] se puede aplicar en distintos aspectos del CE. A continuación se exponen los tipos de Personalización que existen en la literatura:

- **Personalización de los productos:** La personalización de los productos (en inglés *customisation*) es una técnica utilizada también en el comercio tradicional. El comprador puede seleccionar diversas propiedades del producto tales como el diseño (camisetas, carcasa del teléfono móvil, etc) o las funcionalidades que incorpora (automóviles, ordenadores personales, etc). N. Franke y otros en [45] estudian cómo la personalización de los productos hace que el usuario esté dispuesto a pagar más si tiene la posibilidad de configurar el producto, debido a que el usuario tiene un mayor sentimiento de que el producto es “suyo” y eso le otorga un valor adicional. A. Felfernig y otros en [42]
-

utilizan un Sistema de Recomendación para ayudar al usuario a elegir un conjunto de características que satisfaga sus necesidades en una Web que permite configurar las características de distintos teléfonos móviles (estilo, GPS, HSDPA) en función de las necesidades que especifique.

- **Personalización de la interfaz:** La personalización de la interfaz o uso de interfaces adaptativas se encarga de ajustar la presentación de una aplicación (Web, móvil) a las necesidades concretas del usuario para que su interacción con la misma sea más cómoda [13]. Existen técnicas de personalización de la interfaz que no sólo se centran en personalizar la distribución de los elementos, sino que personalizan de cierta manera el contenido que se presenta sobre los productos. Para ello, generan descripciones de los productos adaptadas al usuario que está revisando una lista de productos, de manera que esta descripción se acerque a las necesidades del usuario. Esto se hace destacando las características que son relevantes para él y omitiendo las características a las que da menor importancia [132].
- **Personalización de contenidos:** Se trata de realizar un filtrado de la información que se presenta (como productos o servicios de una tienda de CE) de manera que el usuario no se sienta desbordado por la cantidad de opciones disponibles. Para ello se utilizan los Sistemas de Recomendación, que se estudian con mayor detalle en el Capítulo 2.

1.1.3. Sistemas de Recomendación

Los Sistemas de Recomendación son la técnicas más exitosas dentro de los procesos de Personalización. Se definen como “sistemas que producen recomendaciones personalizadas como salida o tienen el efecto de guiar al usuario de una forma personalizada a productos interesantes o útiles entre una gran cantidad de productos disponibles” [23].

Un Sistema de Recomendación recoge las preferencias de los usuarios mediante valora-

ciones de preferencia sobre productos, información de navegación, etc. Con esta información, un algoritmo de recomendación calcula los productos más adecuados al usuario. Existen distintos enfoques para ello (para más detalles, ver Apartado 2.3 del Capítulo 2), pero los que tienen mayor difusión son los Sistemas de Recomendación Colaborativos y los Basados en Contenido.

Los Sistemas de Recomendación Colaborativos tratan de explotar la inteligencia colectiva. Tratan de predecir la valoración que un usuario daría a un producto que no ha valorado aún. Estos sistemas no utilizan conocimiento adicional sobre los productos que recomiendan, simplemente conocen los productos que ha valorado cada usuario y las valoraciones que ha dado a cada uno de ellos. La idea principal de los SRC es que una persona tiene gustos parecidos a otras personas que se encuentran en su entorno. Los SRC predicen las valoraciones de usuarios sobre productos buscando los usuarios similares (con gustos parecidos) al usuario al que se pretende recomendar. Una vez encontrados los usuarios similares (vecinos) se predice la valoración utilizando la valoración que han dado los vecinos sobre el producto que se predice [39, 51, 38].

Por otra parte, los Sistemas de Recomendación Basados en Contenido tratan de recomendar productos similares a los que le han gustado al usuario en el pasado [87]. Estos sistemas toman cada usuario como un individuo aislado, ya que para realizar recomendaciones a un usuario no utilizan valoraciones del resto de los usuarios del sistema. Estos sistemas utilizan, además de las valoraciones de los usuarios sobre los productos, información descriptiva de cada producto. Esta información puede ser de distinta naturaleza: descripción mediante lenguaje natural [9, 30, 37, 101] o descripción mediante atributos de los productos y valores asociados a los mismos [33, 34].

Esta investigación se centra en la Recomendación Basada en Contenido con Modelado en Espacios Vectoriales. En este método, se construye un vector que representa los gustos del usuario y se calcula su similitud con los vectores que representan las características de los productos. Más concretamente, esta investigación se centra en la mejora de la Ponderación

de Características [122], que trata de incorporar la peculiaridad de que un usuario considere unas características más importante que otras.

1.2. Propósito y Objetivos

En esta investigación se pretende elaborar un Estado del Arte del ámbito de los Sistemas de Recomendación y proponer un nuevo modelo de recomendación con ponderación de atributos que mejore los algoritmos de Recomendación Basados en Contenido existentes. Los objetivos que se establecen para llevar a cabo este propósito global son los siguientes:

- **Revisión del Estado del Arte en el Área de los Sistemas de Recomendación.** Revisión de los modelos que se han propuesto en el ámbito de los Sistemas de Recomendación, entendiendo los distintos enfoques para abordar el problema de la recomendación, señalando métodos de evaluación de los mismos y mostrando las nuevas tendencias en la investigación relacionada con los SR.
 - **Estudio de Técnicas Clásicas en Sistemas de Recomendación Basados en Contenido.** Se expondrán las técnicas clásicas para Sistemas de Recomendación Basados en Contenido y analizarán para encontrar problemas y debilidades de las mismas.
 - **Propuesta de un Sistema de Recomendación Basado en Contenido con medidas de entropía y contingencia.** Se estudiarán distintas opciones para la Recomendación Basada en Contenido con Ponderación de Características que solucione los problemas y debilidades de las técnicas básicas. Se analizarán las herramientas existentes para el cálculo de la entropía y contingencia.
 - **Implementación.** Se realizará la implementación del modelo propuesto.
 - **Caso de Estudio.** Se realizará un Caso de Estudio en el que se mostrarán las mejoras de funcionamiento del modelo propuesto en comparación con las alternativas clásicas en Recomendación Basada en Contenido.
-

1.3. Estructura de la memoria

Para alcanzar los objetivos que perseguimos, esta memoria se estructura en los siguientes capítulos:

- Capítulo 2: En este capítulo se describen los Sistemas de Recomendación, haciendo una clasificación, un repaso de su historia, exponiendo algunos ejemplos de aplicación de los modelos propuestos y realizando una breve reseña sobre las líneas de investigación que tienen mayor interés actualmente en el ámbito.
 - Capítulo 3: En este capítulo se explican en profundidad los modelos básicos para Sistemas de Recomendación Basados en Contenido con Modelado en Espacios Vectoriales, que son el tipo de Sistemas de Recomendación que tienen mayor interés en esta investigación.
 - Capítulo 4: En él se propone un nuevo modelo de recomendación basado en contenido con ponderación de atributos basada en medidas de contingencia y entropía.
 - Capítulos 5: Define la metodología utilizada para la validación del modelo presentado y muestra una comparativa con otros modelos existentes, exponiendo y analizando los resultados de dicha experimentación.
 - Capítulo 6: En este capítulo se exponen las conclusiones extraídas en esta investigación y se realizan algunas propuestas de trabajo futuro.
-

Sistemas de Recomendación

En este capítulo se repasan los conceptos en los que se basan los Sistemas de Recomendación (SR). Para ello, se hace en primer lugar una introducción que expone en qué consiste el problema de la recomendación. Tras esto se recorre la historia de los SR, exponiendo las técnicas que se utilizaban en los inicios de esta rama de investigación y la evolución de las mismas hasta el panorama actual. Después se realiza una clasificación de los mismos, revisando las ventajas e inconvenientes de cada tipo. Seguidamente se explica cómo se realiza una evaluación para comprobar la validez de un SR con un conjunto de datos concreto. Finalmente, se hace una breve revisión de las nuevas tendencias en la investigación en el ámbito de los SR.

2.1. Introducción

R. Burke en [23] define los SR como “sistemas que producen recomendaciones personalizadas como salida o tienen el efecto de guiar al usuario de una forma personalizada a productos interesantes o útiles entre una gran cantidad de productos disponibles” .

Los SR proveen al usuario como salida una serie de productos, servicios o contenidos que se ajustan a sus gustos o necesidades, que previamente han sido recogidos por el sistema. La información que se tiene de los usuarios (en inglés, *feedback*) se utiliza para conocer sus

gustos o preferencias y describe qué productos le han gustado en el pasado al usuario y en qué medida. Esta información puede haber sido obtenida de maneras distintas:

- **Información implícita:** [80, 107] La información de los usuarios se obtiene a partir de su interacción con el sistema, como pueden ser las canciones que ha escuchado, páginas Web visitadas, si ha consultado la descripción de un producto, tiempo que ha permanecido en la misma, acciones sobre los productos (guardado, impresión, etc), etc [74]. Este tipo de información tiene la ventaja de que se puede obtener de una manera relativamente sencilla, pero suele ser poco precisa [95] y, en algunos casos, difícil de aplicar al proceso de recomendación.
- **Información explícita:** En este tipo de información, el usuario debe proporcionar sus preferencias activamente. Para ello se utiliza un sistema de valoración de los productos (mediante una escala numérica [39], una escala lingüística) en el que el usuario indica, en qué medida, un conjunto de productos le han gustado o cubren sus necesidades (productos favoritos, lista de deseos, etc.). La información explícita es una fuente de información más precisa que la información implícita, pero resulta más complicada de obtener, ya que requiere de la colaboración de los usuarios [135].

En ciertos sistemas se puede plantear la utilización de información implícita y explícita a la vez. Esto se puede hacer mediante una combinación de las dos [64], transformando la información implícita a explícita [79, 80] o mejorando los resultados de la recomendación de un tipo refinándolas con los del otro [75].

El criterio para elegir qué tipo de información usa un SR depende del dominio al que se aplique. Como ya se ha comentado, la información implícita es más sencilla de obtener y refleja la interacción del usuario con el sistema, pero es menos exacta que la información explícita. En [5] se demuestra empíricamente que la utilización de ambos tipos de información (implícita y explícita) a la vez, mejora los resultados de recomendación. Además, hay que tener en cuenta qué tipo de información hay disponible en el entorno, el coste de la misma, etc.

Independientemente de que la información sea implícita o explícita, tendremos información de cada uno de los usuarios sobre los productos. Por simplicidad en las explicaciones, se considera que la información que se tiene sobre los usuarios es explícita y que se utilizan valoraciones de preferencia en una escala numérica. De esta manera, el conjunto de valoraciones de preferencia (V) se puede representar de la siguiente manera:

$$V : U \times P \rightarrow D \quad (2.1)$$

donde V es el conjunto de valoraciones de preferencia que los usuarios U han dado sobre los productos P en un dominio D . Uno de los dominios más utilizados para las valoraciones de preferencia es $D = \{1, 2, 3, 4, 5\}$, siendo 1 el valor de preferencia más bajo, que indica que un producto no es del gusto del usuario o no se ajusta a sus necesidades, y 5 el valor de preferencia más alto, que indica que el producto se ajusta perfectamente las necesidades del usuario.

Matemáticamente, el problema de la recomendación se puede definir como un problema de predicción [1]. De esta manera, se toma el conjunto de valoraciones de preferencia como una función incompleta que se pretende aproximar por diferentes técnicas. Una vez predichas las valoraciones de cada producto se selecciona el producto (o k productos) con mayor valoración de preferencia predicha (ver Fórmula 2.2) de entre los productos que el usuario no ha valorado todavía.

$$\forall u \in U, p'_u = \arg \max_{p \in PNV_u} P(u, p) \quad (2.2)$$

donde U es el conjunto de todos los usuarios, PNV_u el conjunto de productos que aún no ha valorado el usuario u y $P(u, p)$ es el valor de utilidad que el SR predice para el producto p y usuario u .

Otro enfoque para solucionar el problema de la recomendación es mediante clasificación binaria de los productos. De esta manera, se intenta discriminar si un producto es adecuado para un usuario o no lo es.

Las tareas básicas que un SR tiene que realizar son tres [47]: el cálculo del perfil de

usuario, el modelado de los productos y el filtrado (ver Figura 2.1):

- **Perfil de usuario.** El SR necesita tener información sobre el usuario al que se pretende recomendar. Para ello se calcula un perfil de usuario a partir de la información que se conoce sobre él. Para esta tarea se han aplicado distintos enfoques, como el modelado del comportamiento, modelado de intereses o de intenciones. Con esta información se pueden realizar clasificaciones de los usuarios, agruparlos (en inglés, *clustering*), extraer patrones de comportamiento, realizar predicciones sobre los mismos, etc.
- **Modelado de productos.** Para conocer los productos que existen en el sistema se calcula también un perfil. Estos perfiles de productos se pueden utilizar para analizar los productos, realizar una extracción de variables semánticas, analizar la diversidad de los mismos, etc.
- **Filtrado.** Una vez conocidos los perfiles de usuarios y de los productos se pretende realizar un filtrado basándose en cómo de adecuado es un producto a un usuario concreto (utilidad) o predecir la valoración que un usuario daría a un producto. Para ello existen distintos enfoques, como el filtrado basado en las características de los productos, basado en reglas o en filtrado colaborativo.

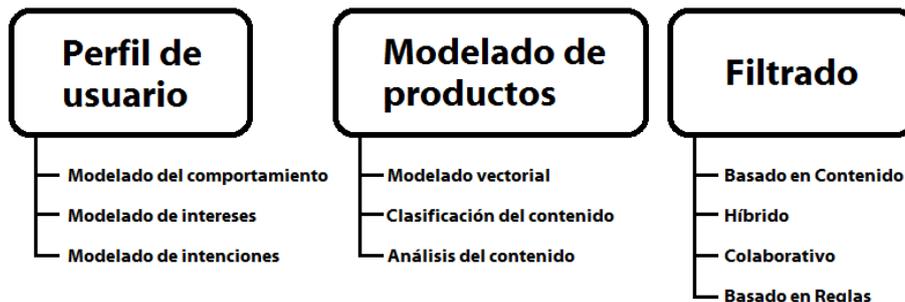


Figura 2.1: Tareas básicas de un Sistema de Recomendación

Estas tareas se realizan de formas distintas según el tipo de SR que se esté desarrollando, incluso algunos sistemas no realizan modelado de productos o de usuarios (para más información, ver Apartado 2.3).

En el siguiente apartado se realiza un repaso de las técnicas que se han utilizado para la realización de recomendaciones a usuarios desde los inicios de los mismos hasta la actualidad.

2.2. Historia

La rama de investigación de los SR surgió como tal a mediados de la década de los 90. Previamente las *recomendaciones* se realizaban mediante algoritmos procedentes de la Recuperación de Información [44], realizando recomendaciones sin personalización, es decir, a todos los usuarios del sistema se recomienda el mismo conjunto de productos. Para ello, se utilizaban distintas herramientas:

- **Filtrado basado en características.** [44] Esta técnica pretende buscar un producto o conjunto de productos que satisfaga una serie de propiedades que el usuario define en una consulta. El mayor problema de esta técnica es que el usuario debe conocer las características de los productos que se encuentran en el sistema y puede especificar consultas que no se correspondan con ningún producto. Para solucionar este problema se han propuesto diversas soluciones, como la búsqueda con la ayuda de una taxonomía [108], la modificación de consultas para eliminar restricciones conflictivas (*constraint relaxation*) [62] o guiar al usuario en el filtrado mediante un refinamiento (*critiquing*) de una consulta inicial [24].
- **Filtrado de productos sin personalización (*Unbiased search*).** Con esta técnica se calcula una lista de productos que se corresponde con los productos mejor valorados (mayor valoración media) o con los productos más populares del sistema (el más comprado).
- **Filtrado con datos generales de los consumidores.** Utilizan datos generales que el sistema conoce, sin realizar ninguna operación con los datos del usuario activo¹.

¹usuario al que se desea recomendar

Se han utilizado técnicas como la recomendación a partir otros productos. En [7] se recomiendan productos que han sido comprados por otros usuarios que también compraron/revisaron el producto actual (el que se está revisando en el momento de la recomendación). También se pueden calcular recomendaciones en tiendas electrónicas utilizando el conjunto de productos que hay en el momento de la recomendación en el *carrito de la compra*.

Los Sistemas de Recomendación Demográficos (SRD) fueron los primeros SR que se comenzaron a utilizar. En este tipo de sistemas, el usuario es clasificado en un perfil según sus características demográficas y las recomendaciones se calculan basándose en la información de necesidades, preferencias o gustos de dicho perfil demográfico. Grundy [103], fue el primer sistema de estas características. En este sistema se recomiendan libros a los usuarios mediante una interfaz de diálogo interactivo que solicita al usuario la información demográfica que necesita conocer en cada momento sobre el mismo. A partir de los datos proporcionados, se activan una serie de perfiles que, conjuntamente, describen las necesidades y gustos del usuario. Los SRD no son muy utilizados, ya que utilizan información sobre los datos personales de los usuarios y muchos usuarios no se sienten seguros proporcionando tal información.

En [117] se comienza a utilizar la recuperación de información personalizada, que intenta refinar la búsqueda de documentos incorporando información sobre las necesidades específicas del usuario, además de la consulta en lenguaje natural. De esta manera, una consulta puede devolver una recomendación distinta, dependiendo del usuario que la formule, ya que filtra la información de forma diferente según el perfil de usuario.

Un tipo de sistemas que también se utilizaba en los inicios de los SR son los Sistemas de Recomendación Basados en Contenido (SRBCont). En este tipo de sistemas se tiene información sobre el contenido de los productos, como sus atributos o descripciones de los mismos. El perfil de usuario se calcula utilizando la información de contenido de los productos que le gustan. Uno de los primeros SRBCont es el sistema propuesto en [78], que es un SRBCont de noticias en el que se toma cada palabra como una característica, para describir

el contenido de la misma mediante representaciones en espacios vectoriales. También se han propuesto otros métodos en estos SR, tales como redes neuronales [29] y redes bayesianas [36].

Posteriormente surgieron los Sistemas de Recomendación Colaborativos (SRC) (ver Apartado 2.3.3). El primer trabajo sobre Filtrado Colaborativo [50] propone un sistema de correo electrónico que soporta la agregación de filtros para la recepción de correos sin la utilización de listas de correo. Estos filtros son diseñados por el usuario mediante un lenguaje de consulta propio (TQL, en inglés *Tapestry Query Language*), que permite construir filtros para recibir, entre otros, correos de listas de distribución que han sido valorados positivamente por personas que sabemos que tienen intereses afines a los nuestros.

Un sistema posterior es el propuesto por P. Resnick en [101], que se basa en el filtrado de noticias combinando información de contenido de las mismas (texto en lenguaje natural) con información de las preferencias de otros usuarios en el sistema. La mejora que aporta sobre la propuesta mencionada en el párrafo anterior, es que el usuario no necesita conocer otros usuarios del sistema, sino que calcula los usuarios similares al usuario al que se recomienda (por medio de la correlación de valoraciones pasadas).

Los sistemas anteriores utilizan perfiles a largo plazo, es decir, suponen que las preferencias del usuario no cambian a lo largo del tiempo y son iguales en sesiones distintas. Los Sistemas de Recomendación Basados en Conocimiento (SRBCono) [26] calculan las recomendaciones intentando hacer inferencias sobre las necesidades y preferencias de los usuarios en el momento de la recomendación. La información que utilizan es, para el usuario, las necesidades y preferencias del mismo y, para los productos, cómo satisfacen cada una de las necesidades que un usuario puede tener. Uno de los primeros sistemas de este tipo es el sistema Entree [26], que intenta hacer recomendaciones de restaurantes considerando los restaurantes que a un usuario le han gustado en el pasado y, además, permite navegar por las recomendaciones refinando la consulta mediante ajustes como “restaurantes más baratos”, “restaurantes más románticos”, etc.

Un punto de inflexión en la historia de los SR es la competición organizada por Netflix desde 2006 a 2009. Para dicha competición, Netflix² puso a disposición pública un conjunto de valoraciones de preferencia de 100 millones de registros, en el que 480189 usuarios dan valoraciones sobre 17770 películas y los participantes competían por minimizar el error cometido en la predicción de las valoraciones mediante técnicas de recomendación colaborativas. S. Funk en [46] describe cómo mediante la descomposición en valores singulares (en inglés, *Singular Value Decomposition*, SVD) se conseguía una mejora con la que se podía estar entre los diez primeros del *ranking* del concurso en el momento de su publicación. Este trabajo expone cómo calcular la descomposición en valores singulares de la matriz de valoraciones de dimensión 480189×17770 , que contiene celdas cuyo valor no es conocido. Finalmente el equipo ganador de la competición (BellKor's Pragmatic Chaos) consiguió mejorar el error de predicción un 10 % sobre el algoritmo original de Netflix. Se basa en integrar las predicciones de varios (107) enfoques en vez de mejorar una única técnica.

A lo largo de la historia de los SR encontramos con frecuencia sistemas híbridos [23, 25] en los que se intentan conseguir mejores resultados (error absoluto, precisión, etc) [36], así como dar solución a alguno de los problemas que los SR presentan de manera aislada [105, 137].

2.3. Técnicas de Recomendación

En la literatura se han propuesto muchas soluciones al problema clásico de recomendación (ver Fórmula 2.1). Dichas soluciones se clasifican en los siguientes tipos [3]:

- Sistemas de Recomendación Colaborativos (SRC).
- Sistemas de Recomendación Basados en Contenido (SRBCont).
- Sistemas de Recomendación Demográficos (SRD).
- Sistemas de Recomendación Basados en Utilidad (SRBU).

²www.netflix.com

- Sistemas de Recomendación Basados en Conocimiento (SRBCono).
- Sistemas de Recomendación Híbridos.

En los siguientes apartados se detalla el funcionamiento, técnicas aplicadas y ventajas y desventajas de cada uno de estos SR.

2.3.1. Sistemas de Recomendación Demográficos

Los Sistemas de Recomendación Demográficos (SRD) se basan en la idea de que personas con unas características demográficas dadas (edad, sexo, nivel de educación, domicilio) tengan gustos similares a otras personas con características demográficas similares. A continuación se exponen algunos trabajos que proponen distintos modelos para aprovechar la información demográfica de los usuarios para mejorar el proceso de recomendación. Técnicas

En [103] el sistema intenta asociar al usuario con una serie de estereotipos a partir su información demográfica. Con esto, se pretende realizar una extensión de la información inicial mediante el conocimiento adicional almacenado en los estereotipos y así realizar recomendaciones más precisas y acordes al perfil del usuario. Además, el sistema organiza los distintos estereotipos de una forma jerárquica, de manera que se tiene un conocimiento a mayor nivel de los estereotipos, permitiendo realizar razonamientos de inferencia. En la fase de recomendación, se combinan los distintos estereotipos a los que el usuario pertenece para generar un resumen de su información (ya sea información que ha especificado explícitamente o información inferida). Finalmente el sistema usa este resumen para filtrar los productos que se presentan al usuario, así como la información que se presenta al usuario sobre cada producto.

Otro modelo que funciona de una manera similar es el propuesto por B. Krulwich [76]. En este modelo, se utilizan los perfiles demográficos extraídos por el sistema demográfico PRIZM [131]. PRIZM divide la población de Estados Unidos en 62 *clusters* demográficos atendiendo a información relativa al historial de compras, estilo de vida y otra información

extraída mediante encuestas. El SR utiliza la información demográfica proporcionada por el usuario para clasificarlo en los *clusters* que define PRIZM.

M.J. Pazzani [97] utiliza, como información sobre los usuarios, los términos (palabras) que aparecen en en la página inicial de su *Web*, así como las valoraciones de preferencia que cada usuario otorgó a una serie de restaurantes. Con esta información, aplica el algoritmo de Winnow [85] para realizar un proceso de aprendizaje que determine las características relevantes de las páginas *Web* asociadas a los usuarios a los que le gusta un restaurante concreto.

M. G. Vozalis y K. G. Margaritis [129] aplican una técnica más compleja para la predicción de valoraciones de preferencia en recomendación: la descomposición en valores singulares (SVD). Para ello, se aplica SVD a los vectores de usuario con información demográfica y a la matriz de valoraciones y aplica una medida de correlación que tiene en cuenta tanto la información demográfica como la similitud de las valoraciones de preferencia.

S. Schiaffino, y A. Amandi [111] proponen un SR de paquetes turísticos, los cuales se componen de múltiples *tours*. Este sistema realiza una hibridación que integra recomendación basada en contenido, filtrado colaborativo y filtrado demográfico. La recomendación demográfica compara las características demográficas del usuario y de los paquetes turísticos para dar un valor de utilidad del mismo para el usuario.

Aunque en los inicios de los SR se utilizaban mucho los SRD, actualmente se aplican a dominios muy concretos, debido a la dificultad de disponer de los datos demográficos necesarios para su funcionamiento, debido a la preocupación de los usuarios por su privacidad.

2.3.2. Sistemas de Recomendación Basados en Contenido

En los SRBCont se tiene información sobre las características de cada producto y se intentan extraer relaciones entre estas y la valoraciones de preferencia de un usuario. Para ello, se construye un perfil de usuario a partir de los productos que ha valorado y se utiliza

para evaluar los productos no experimentados. Este tipo de SR no trata de predecir la valoración que un usuario daría, sino que asigna una puntuación a los productos, que indica la idoneidad del producto para el usuario.

$$Utilidad(u, p) = Puntuación(Perfil(u) , Contenido(p))$$

La información que estos sistemas utilizan se describe en la Tabla 2.1:

- $a_{x_{u,1}}, \dots, a_{x_{u,nu}}$ son los productos que el usuario u ha valorado. $x_{u,1}$ denota el primer producto que el usuario u valoró.
- r_1^u, \dots, r_{nu}^u son los respectivos valores de preferencia que el usuario ha dado.
- c_1, \dots, c_m son las características que tienen dichos productos.
- $v_1^{x_{u1}}, \dots, v_m^{x_{u1}}$ son los valores de las características para el producto x_{u1} .

	c_1	\dots	c_j	\dots	c_m
a_1	v_1^1	\dots	v_j^1	\dots	v_m^1
\dots	\dots	\dots	\dots	\dots	\dots
a_n	v_1^n	\dots	v_j^n	\dots	v_m^n

Tabla 2.1: Datos de un Sistema de Recomendación Basado en Contenido

En los SRBCont propuestos en la literatura se utilizan dos tipo de información de contenido distintos:

- **Descripción textual:** La información de contenido de los productos son una serie de términos asociados al mismo, que aparecen en su descripción. Las características en este caso serán términos, y el valor de la misma sera 1 si el término aparece en la descripción, 0 si no aparece.

- **Atributos:** En este caso, las características son una serie de atributos (en una película, actores, directores, género, etc.) para los que un producto dado tiene un valor concreto (género: *drama,comedia,terror,etc.*).

En SRBCont se han aplicado distintas técnicas, muchas de ellas relacionadas con la clasificación. A continuación se presentan las más utilizadas:

- **Modelado en Espacios Vectoriales.** Esta técnica procede del área de la recuperación de información. Para calcular las recomendaciones, el sistema calcula un vector asociado al usuario y obtiene la similitud (coseno, correlación) con los vectores asociados a los productos que el usuario no ha valorado. Finalmente se recomiendan los N productos cuya similitud con el perfil del usuario sea mayor.

Por ser la técnica utilizada para el SR propuesto en este trabajo, esta técnica se explica con mayor detalle en el Capítulo 3.

- **Redes Bayesianas.** Una red bayesiana [63] es un modelo probabilístico que relaciona variables aleatorias mediante un grafo acíclico dirigido (en inglés, *Directed Acyclic Graph*, DAG), cuyos nodos representan variables aleatorias y los arcos representan relaciones de dependencia entre las mismas. Un grafo acíclico dirigido es aquel cuyas aristas sólo se pueden seguir en una dirección y que para cualquier los nodos v no existe un camino que empiece y termine en v . Este tipo de redes permite realizar inferencias para calcular la probabilidad de una variable condicionada al valor de las variables de las que depende (ascendientes).

L.M. de Campos y otros [36] aplican redes bayesianas para el cálculo de recomendaciones basadas en contenido. La red bayesiana contiene nodos para cada valor de cada característica y para cada producto. Utilizando las valoraciones que otros usuarios dan a productos similares, se calcula una puntuación para el producto. Se recomiendan los k productos con mayor puntuación

- **Redes Neuronales.** Una red neuronal intenta simular el sistema nervioso de los ani-
-

males. Se representa mediante un grafo, en el que los nodos son las neuronas artificiales y los arcos son conexiones entre las distintas neuronas. Dichas conexiones tienen asociado un valor que representa el peso de la conexión. Un tipo de redes neuronales ampliamente utilizado es el perceptrón (ver Figura 2.2).

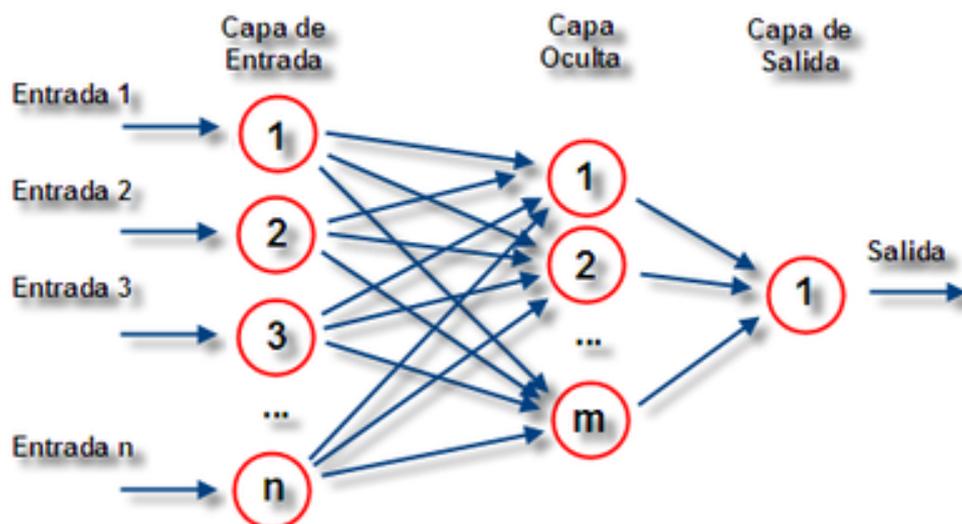


Figura 2.2: Configuración de una red neuronal

C. Christakou y A. Stafylopatis [29] aplican redes neuronales para recomendar películas. Para ello, construyen 2 redes neuronales: la primera se utiliza para calcular las recomendaciones en base a las características de las películas (género, actores y directores) y la segunda utiliza los términos (palabras) en el resumen de la película. La lista de recomendaciones final combina los resultados de ambas redes neuronales y de un SRC.

Los SRBCont necesitan que el usuario valore una cantidad suficiente de productos para poder recoger sus gustos o necesidades (arranque en frío en usuarios). Además, por las características de su funcionamiento, los SRBCont no pueden realizar recomendaciones con serendipia (en inglés, *serendipity*). La serendipia en SR es la capacidad del sistema para realizar recomendaciones de productos inesperados, es decir, productos distintos de los que

el usuario ha valorado previamente.

2.3.3. Sistemas de Recomendación Colaborativos

Los SRC aparecieron a mediados de los años 90 [50, 101, 115]. Estos SR funcionan basándose en el concepto del “boca a boca” (en inglés, “*word-of-mouth*”), que calcula recomendaciones para un usuario concreto (usuario activo) utilizando la información proporcionada por el resto de los usuarios que hay en el sistema. Algunos sistemas que utilizan este enfoque en el cálculo de las recomendaciones son [50, 101].

Las fases generales que tienen en común los SRC en el cálculo de las recomendaciones son las siguientes (ver Figura 2.3):

1. **Cálculo de vecinos:** Se realiza un cálculo del vecindario, esto es, los usuarios con gustos o necesidades más parecidas al usuario activo.
2. **Predicción de la valoración de preferencia:** Una vez obtenidos los vecinos se realiza una predicción que estima el valor de preferencia que el usuario activo daría a cada uno de los productos que no ha valorado.
3. **Recomendación de los mejor valorados *Top-N*:** Se ordena la lista de los productos recomendados por su valor de preferencia predicho en orden descendente y se recomiendan los N primeros productos de la lista.

Los SRC tienen la ventaja de que no es necesario tener información sobre los productos que se van a recomendar. Esto es porque los SRC tratan los productos como una caja negra, de la que sólo se conocen las valoraciones de preferencia que los distintos usuarios del sistema han dado sobre los mismos. Otra ventaja de los SRC es que su precisión mejora con el tiempo, cuantos más usuarios haya en el mismo y mayor cantidad de valoraciones compartan con el resto, mejores serán sus resultados.

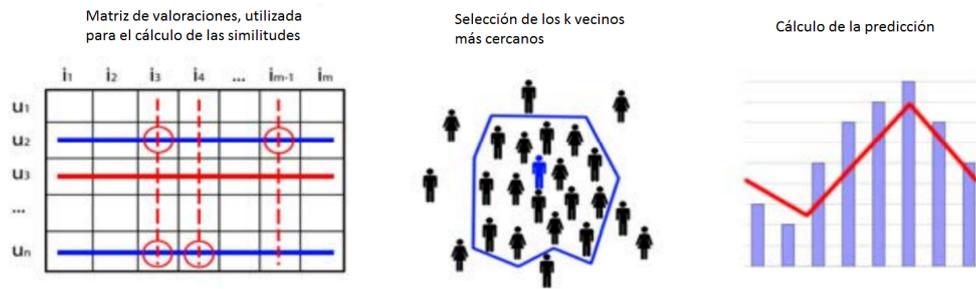


Figura 2.3: Proceso recomendación en el filtrado colaborativo

John S. Breese y otros [18] hacen una revisión de las técnicas básicas de minería de datos aplicadas al filtrado colaborativo y una evaluación de qué técnicas tienen una mayor eficacia en distintos conjuntos de datos. Las técnicas más utilizadas para solucionar los problemas asociados al filtrado colaborativo se describen a continuación:

- **Vecinos Cercanos (*Neighborhood*)**. El algoritmo más extendido para el cálculo de recomendaciones colaborativas es el algoritmo de vecinos cercanos [31]. Este algoritmo se basa en buscar los los k elementos más cercanos (vecinos más cercanos) al ítem que se desea clasificar. Conociendo la clase de los elementos más cercanos se calcula la clase a la que pertenece el ítem que se desea clasificar.

Aplicada a la recomendación, el algoritmo de vecinos cercanos basado en memoria (también denominado *User-User*) calcula los usuarios más parecidos al usuario activo (vecindario) y realiza una predicción de la valoración de preferencia que el usuario daría a un producto utilizando las valoraciones de preferencia que otorgaron sus vecinos más cercanos. En este tipo de sistemas, las variables a tener en cuenta son el número de vecinos que se utilizan (k), la medida de similitud a utilizar para el cálculo de los vecinos (distancia Euclídea, medida del coseno, similitud de Pearson, etc) y la técnica de predicción que se aplica para agregar las valoraciones de preferencia de los vecinos (suma ponderada, suma ponderada ajustada con la media, etc). Este algoritmo se denomina *basado en memoria* porque no almacena explícitamente un perfil de usuario, sino que se almacenan todas las valoraciones de preferencia y se recalcula la lista de

vecinos cada vez que se solicitan recomendaciones al sistema. Esto hace que los SRC con vecinos cercanos basados en memoria tengan problemas de escalabilidad cuando el número de usuarios crece. Para mitigar este problema en [6] se utilizan las preferencias de expertos en lugar de todos los usuarios del sistema.

Otra solución para el problema de la escalabilidad es la aplicación del algoritmo de vecinos cercanos con otro enfoque: basado en modelo [109, 38, 130] (también denominado *Item-Item*). En este enfoque no se calculan los vecinos del usuario, sino que para cada producto se calculan sus productos similares (vecinos) y a partir de dicho modelo se realiza la predicción. Este algoritmo se denomina *basado en modelo* porque se calcula un modelo de predicción *off-line* que se actualiza periódicamente (diariamente, semanalmente, cuando se añadan n valoraciones). El modelo almacena para cada producto los k productos más similares (junto con el grado de similitud), utilizando para el cálculo de la similitud las valoraciones comunes (ver Figura 2.4) de los productos a comparar. Para predecir la valoración que un usuario u daría a un producto p , se utilizan las valoraciones que dicho usuario ha dado a los vecinos más próximos del producto p . Para componer la predicción de la valoración de preferencia se agregan las valoraciones realizando una suma ponderada por la similitud de los vecinos con el producto objetivo.

- **Clustering.** Existen otras técnicas aplicables a la realización de recomendaciones colaborativas, como el *clustering*. El *clustering* [55] consiste en clasificar los individuos en grupos de manera que los individuos de un mismo grupo sean lo más parecidos posible entre sí y en dos grupos distintos sean lo más distintos posible entre sí.

C.S. Hwang y P.J. Tsai [60] aplican técnicas de *clustering* para dividir el conjunto de usuarios en grupos de usuarios similares y en dichos grupos se aplica recomendación colaborativa con vecinos cercanos basada en modelo. De esta manera, se genera un modelo teniendo en cuenta sólo las valoraciones de los usuarios que pertenecen al mismo cluster al que pertenece el usuario activo y se realiza la recomendación.

- **Algoritmos basados en Reglas de Asociación.** En SRC se han aplicado algoritmos
-

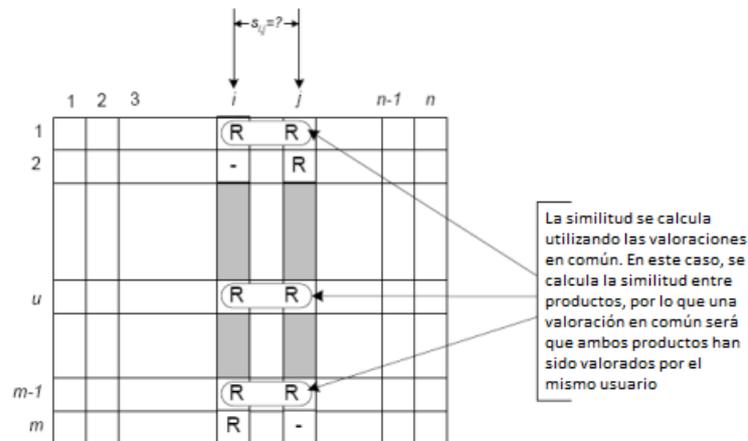


Figura 2.4: Cálculo de la similitud entre productos

basados en reglas de asociación. Las reglas de asociación son sentencias de tipo:

si [antecedente] *entonces* [consecuente]

donde antecedente es una condición (o conjunción de condiciones) y el consecuente es la acción (o conjunción de acciones) que se realiza cuando se cumple el antecedente. Sobre las reglas de asociación se definen dos medidas:

- La **confianza** es el porcentaje de individuos de la base de datos que cumple el antecedente.
- El **soporte** es el porcentaje de individuos de la base de datos que cumple el antecedente y además cumple el consecuente.

Weiyang Lin y otros [84] utilizan reglas de asociación para la realización de recomendaciones, dividiendo el intervalo de valoraciones en dos clases: Gusta y NoGusta. Una vez procesados los datos se extraen reglas de los usuarios o de los productos³. El sistema permite utilizar las reglas de usuarios, de producto o los dos tipos a la vez. La recomendación se realiza asociando a cada producto a recomendar una puntuación,

³Usuarios: Los artículos que le gustaron al usuario A y B les gustan al usuario C. Productos: Los usuarios a los que les gustó el producto A y B les gusta el producto C

que se calcula teniendo en cuenta la confianza y el soporte de las reglas que se activan para dicho producto.

Cane Wing-ki Leung y otros [82] aplican reglas difusas para el cálculo de las recomendaciones. Una regla difusa es similar a una regla de asociación, pero utiliza variables difusas en el antecedente y consecuente. De esta manera, se transforman las valoraciones de preferencia al valor difuso de la variable *PreferenciaDifusa* que está definida con las etiquetas difusas *Gusta*, *Neutral* y *NoGusta*. Una vez las valoraciones del usuario se han transformado, se aplica un algoritmo de extracción de reglas difusas para los usuarios y para los productos⁴, de manera que en la fase de recomendación se elabora un conjunto de productos candidatos determinado por las reglas difusas que se activan, es decir, las reglas cuyo antecedente se cumple con grado de pertenencia distinto de cero. Para estos productos se predice la valoración por medio de un proceso de *defuzzificación*, en la que se transforma un valor en un dominio difuso a un valor en el dominio real (*crisp*).

En lugar de utilizar reglas con la base de datos completa, Fatih Gedikli y otros [48] genera una lista de reglas que se aplican a cada usuario. Dichas reglas se generan de manera que para el antecedente se tiene un producto (o conjunto de productos) que ha valorado el usuario activo y el consecuente contiene los productos valorados positivamente por los vecinos que cumplen el antecedente. Por lo tanto el perfil de un usuario en el sistema es el conjunto de reglas generado a partir de sus vecinos. La recomendación se calcula de forma similar a [84].

- Redes Bayesianas** Las redes bayesianas se utilizan a menudo para tareas de clasificación. El problema de la clasificación consiste en asignar una clase a las instancias extrayendo conocimiento de una serie de ejemplos cuya clase es conocida [93]. Un clasificador bayesiano [40] utiliza las características de los individuos para calcular el valor de la clase, haciendo la suposición de que las características son independientes entre

⁴Usuarios: si $\langle Usuario1, Gusta \rangle$ entonces $\langle UsuarioActivo, Gusta \rangle$. Productos: si $\langle Producto1, Gusta \rangle$ entonces $\langle ProductoActivo, Gusta \rangle$

sí dada la clase. Un clasificador bayesiano da como salida la probabilidad de que un individuo pertenezca a una clase (ver Fórmula 2.3). La clase con mayor probabilidad es la clase que el clasificador asigna al individuo.

$$p(\text{individuo} \in \text{clase}_j) = p(\text{clase}_j) \prod_i^n p(c_i | \text{clase}_j) \quad (2.3)$$

donde clase_j es la j -ésima clase del conjunto de clases, c_i es la i -ésima característica del individuo y $p(\text{clase}_j)$ y $p(c_i | \text{clase}_j)$ pueden ser estimados a partir del conjunto de individuos de entrenamiento.

Koji Miyahara y M.J.Pazzani [94] utilizan la clasificación bayesiana para recomendación colaborativa convirtiendo la matriz de valoraciones de preferencia (definidas en dominio real) en una matriz que contiene para cada usuario las clases Gusta y NoGusta [14]. De esta manera, para un producto, el valor de Gusta es verdadero y el de NoGusta falso si la valoración de preferencia está por encima de cierto umbral y al contrario si está por debajo. Con la matriz transformada se realiza el cálculo de los vecinos y se genera un clasificador bayesiano, que se utiliza para predecir qué productos pertenecen con mayor probabilidad a la clase Gusta y se recomiendan.

L.M. de Campos y otros [35] dan un enfoque distinto a la aplicación de redes bayesianas en SRC. En su trabajo no se utilizan para predecir un producto para el usuario activo si pertenece a la clase Gusta o a la clase NoGusta [14, 94, 121] sino que se utiliza para predecir las valoraciones de preferencia que el usuario activo daría a los productos no valorados. Para ello, utiliza una topología que refleja las relaciones de los usuarios con los productos en un nivel y, en el siguiente nivel, las relaciones entre usuarios. La dependencia de los usuarios con el resto se calcula mediante una combinación del Coeficiente de Correlación de Pearson (ver Apartado 4.2.1.2) y número de valoraciones en común que tienen sobre el total de valoraciones. La red bayesiana da como salida la probabilidad de que el usuario valore un producto con cada uno de los valores posibles propagando los valores de probabilidad conocidos (con las valoraciones de preferencia de los vecinos). Dichas probabilidades se traducen finalmente en una valoración en el dominio original, obteniendo una predicción de la valoración de preferencia.

- **Reducción de Dimensionalidad.** La Reducción de Dimensionalidad consiste en realizar una transformación de los datos originales, de manera que los datos transformados sean más simples. Esta transformación se basa en la hipótesis de que pueden existir variables ocultas que condicionen el valor de múltiples características de los individuos de un conjunto de datos. Por lo tanto, la tarea de las técnicas de reducción de dimensionalidad consisten en encontrar dichas variables ocultas para realizar una *simplificación* de los datos originales. Esto se traduce en realizar una transformación de un espacio de alta dimensionalidad en un espacio de dimensionalidad reducida.

Los SRC son un campo cuyos datos se adecuan a la aplicación de técnicas de este tipo, ya que existen dominios con gran cantidad de productos, como puede ser la recomendación de libros, de películas o de canciones. El Análisis de Componentes Principales y la Descomposición en Valores Singulares han sido aplicados a los SRC con éxito, por lo que se presentan a continuación.

- **Análisis de Componente Principales.** Análisis de Componente Principales (en inglés *Principal Component Analysis, PCA*) [59, 66] se aplica sobre conjuntos de datos en los que se desea descubrir relaciones de covarianza en los datos originales. Este método realiza una transformación en los datos para intentar que los datos resultantes no presenten dichas covarianzas, lo que permite trabajar con datos cuyas características son independientes entre sí.

Una de las aplicaciones más exitosas de PCA a los SRC fue la aportación de Ken Goldberg y otros [51]. En su trabajo, aplicaron PCA a las valoraciones de preferencia de los usuarios para realizar una separación de los mismos que permitiera mejorar los resultados de clustering sobre los usuarios del SR. En este sistema, los usuarios dan valoraciones de preferencia en un intervalo continuo para expresar el grado de satisfacción del usuario con un producto. Como inconveniente, la aplicación de PCA a la recomendación de productos obliga a que los usuarios den sus valoraciones de preferencia sobre un conjunto común de productos, es decir, que todos los usuarios hayan valorado un conjunto fijo de productos. Esto hace que

PCA no se pueda aplicar en ciertos dominios, ya que según el tipo de productos que recomiendan, puede no ser sencillo que el usuario experimente los productos para dar de su valoración de preferencia.

- **Descomposición en Valores Singulares.** La Descomposición en Valores Singulares (en inglés *Singular Value Decomposition*, SVD) [46] es una técnica de factorización de matrices que permite descomponer una matriz A de la siguiente manera:

$$SVD(A) = U \times S \times V^t \quad (2.4)$$

Reduciendo el número de valores singulares (en la matriz S) que se almacenan a los primeros k valores, se obtiene una aproximación de la matriz original A , que permite reconstruirla a partir de las versiones reducidas de las matrices en las que se descompone cometiendo cierto error:

$$A_k = U_k \times S_k \times V_k^t \quad (2.5)$$

Este método resulta en una transformación de los datos originales de alta dimensionalidad en una representación reducida de los mismos (de dimensión k). La i -ésima fila de la matriz U_k se corresponde con la representación del usuario i (fila) de la matriz A en el espacio reducido. De forma análoga, la j -ésima columna de la matriz V_k^t representa al producto j (columna) de la matriz A en el espacio reducido.

SVD puede ser utilizado como técnica de pre-procesamiento de los datos para posteriormente aplicar el algoritmo de vecinos cercanos sobre los usuarios o productos, o aplicar técnicas de *clustering* sobre los datos de los usuarios transformados [70].

Como el cálculo de SVD puede ser costoso, se han propuesto modelos que permiten agregar nueva información sin tener que recalcular completamente la descomposición [67, 110].

Un punto de inflexión en la aplicación de SVD en SRC se produjo en 2006 por S. Funk. Mientras se celebraba la competición organizada por Netflix, propuso una

forma de calcular SVD para matrices de grandes dimensiones [46], lo que abrió las puertas a la aplicación de SVD en sistemas de mayor escala [96].

Los SRC tienen ciertas desventajas que se han intentado solucionar: necesitan una gran cantidad de datos para funcionar y sufren problemas de escalabilidad. Este último se ha intentado solucionar mediante técnicas de clustering [60], reducción de los usuarios vecinos [6] y otras técnicas comentadas en los siguientes apartados. Un problema adicional de los SRC es el arranque en frío (en inglés, *cold-start problem*). Este problema se da de dos formas:

- **Arranque en frío en usuarios:** Un usuario debe dar un número suficiente de valoraciones de preferencia para que se pueda calcular correctamente su vecindario.
- **Arranque en frío en productos:** Se produce cuando se introduce un nuevo producto en el sistema. Dicho producto no puede ser recomendado hasta que se tenga una cantidad de información suficiente sobre el mismo, es decir, hasta que sea valorado por un número suficiente de usuarios del sistema.

Existen distintos métodos para solucionar este problema, como la hibridación con otros métodos [105, 120], la incorporación de fuentes de información adicionales [128], el uso de reglas de asociación [83, 116] o la utilización de otras medidas de similitud entre usuarios.

2.3.4. Sistemas de Recomendación Basados en Conocimiento

Los Sistemas de Recomendación Basados en Conocimiento (SRBCono) son sistemas que utilizan una base de conocimiento que describe cómo los distintos productos satisfacen las necesidades de un usuario y en qué medida (conocimiento funcional del entorno) [22]. De esta manera, el sistema encuentra el producto (o productos) que se ajusta a las necesidades que el usuario ha especificado. Esta búsqueda de productos a partir de las necesidades se realiza mediante un proceso de inferencia de algún tipo. Esta arquitectura permite a estos sistemas obtener y almacenar la información de distintas formas.

Se puede decir que este tipo de SR son los más heterogéneos en cuanto a la interacción con el usuario. En la literatura existen SRBCono que:

- Solicitan al usuario sólo valoraciones y realizan la inferencia a partir de ellas [16].
- Solicitan al usuario productos similares a los que busca y devuelven una lista de productos similares [90].
- Permiten que el usuario refine las recomendaciones [24, 23, 28, 99, 100, 125, 126].
- Permiten que el usuario establezca relaciones de preferencia entre una lista de productos [105].
- Permiten que el usuario establezca relaciones de preferencia entre las características de los productos [19].

Para realizar un proceso de recomendación correcto con pocas cantidades de información sobre los usuarios, estos sistemas necesitan tener un conocimiento adicional sobre el entorno en el que se utilizan. Además de tener información sobre las características de los productos, necesitan información sobre las relaciones entre los mismos, su capacidad para satisfacer las necesidades del usuario, etc. Esto puede resultar una tarea costosa, ya que se necesita realizar un proceso de Ingeniería del Conocimiento para obtenerla.

Por ello, la aplicación de SRBCono en distintos dominios (recomendación de películas, coches, libros, música, noticias) no resulta trivial y requiere un gran esfuerzo para trasladarlo de un dominio a otro.

A continuación se describen los distintos enfoques que se utilizan para la integración del conocimiento en los procesos de recomendación:

- **Razonamiento Basado en Casos.** El Razonamiento Basado en Casos (en inglés, *Case-Based Reasoning*) [73] es una técnica de resolución de problemas que utiliza el conocimiento que proporcionan las experiencias previas. Para ello, se realiza un
-

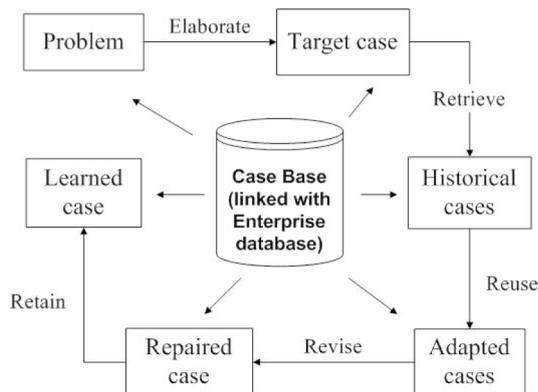


Figura 2.5: Proceso de resolución en Razonamiento Basado en Casos

proceso de evaluación del problema que se plantea y de las experiencias pasadas para dar solución al mismo [88]. Esto se hace mediante las siguientes fases (ver Figura 2.5):

1. Recuperación. El sistema busca los casos similares al problema que se está planteando en este momento. Para ello, se puede calcular qué casos son similares al problema actual o realizar un análisis que determine las características relevantes de los casos que se deben recuperar.
2. Reutilización. Se estudia la información y el conocimiento que tenemos sobre el problema y se utilizan los casos recuperados en la fase anterior para proponer una solución.
3. Revisión. En algunos casos, la solución propuesta a partir de las experiencias anteriores puede no ser óptima, por lo que se puede realizar un ajuste de la misma utilizando experiencias anteriores o conocimiento adicional sobre el entorno.
4. Almacenamiento. Se almacena un nuevo caso, indicando si la solución propuesta fue correcta o no. Esta información (experiencia) será utilizada en futuras resoluciones.

El Razonamiento Basado en Casos se ha aplicado al problema de la recomendación de diferentes maneras. Por ejemplo, en [21, 22, 23] un caso es un producto a recomendar y, sobre el mismo, el usuario realiza un refinamiento que lo guía a nuevos productos,

pudiendo explorar opciones de manera guiada. En este caso, el usuario realiza una modificación de la formulación del problema, proporcionando un producto similar al que busca y una dirección en el espacio de búsqueda. Esta dirección viene dada por una re-formulación de sus necesidades y restricciones, por ejemplo, buscar restaurantes románticos, buscar coches más económicos, etc.

- **Conocimiento mediante ontologías.** Una forma de especificar conocimiento sobre un dominio concreto son las ontologías. Una ontología [118] describe diferentes propiedades y relaciones sobre las entidades del dominio al que se refiere, como pueden ser:

- Tipo de entidades que existen en el dominio.
- Relaciones entre las distintas entidades y su significado.
- Propiedades comunes de las entidades de un tipo concreto.
- Descripción de una jerarquía de entidades mediante la relación *es un*.
- Herencia de propiedades de los objetos que pertenecen a un mismo tipo de entidad.

Las ontologías proporcionan un conocimiento completo y detallado sobre el dominio que describen. Estas se pueden utilizar para realizar un proceso de inferencia que permita mejorar los procesos de recomendación añadiendo información de un nivel superior.

Existen múltiples lenguajes que permiten desarrollar ontologías. Las primeras ontologías se realizaban mediante *Resource Description Framework* (RDF) y *Resource Description Framework Schema* (RDFS). Posteriormente se utilizaron *DARPA Agent Markup Language* (DAML) y *Ontology Inference Layer* (OIL), que finalmente se fusionaron para dar lugar a la definición estandarizada por *World Wide Web Consortium* (W3C). *Web Ontology Language* (OWL) es el lenguaje que permite describir el conocimiento sobre un dominio con mayor expresividad.

Una característica interesante de las ontologías es que pueden utilizarse para compartir información entre distintos sistemas. De esta manera, se incentiva la cooperación de

distintos sistemas y la reutilización de la información que han recogido en su funcionamiento para su posterior reutilización.

Esto ha propiciado que distintas organizaciones hayan construido ontologías sobre distintos dominios, como el acceso a información de disponibilidad de recursos en bibliotecas (*Document Availability Information Ontology*⁵) o películas (*Movie Ontology MO*⁶).

En el problema de la recomendación se han aplicado las ontologías para completar la información sobre los productos y usuarios. Existen trabajos que aplican la información de una ontología sobre el dominio del SR para calcular una medida de similitud entre objetos y usuarios del sistema [8, 17].

En otros trabajos se utilizan las ontologías para describir el contexto de los usuarios y productos del sistema [138], para realizar una recomendación guiada de destinos turísticos mediante un sistema de diálogo [86].

- **Sistemas de Recomendación Basados en Restricciones** Otro tipo de SRBCono son los Sistemas de Recomendación Basados en Restricciones [41, 61, 62, 136]. En estos sistemas se realiza una búsqueda incremental que elimina productos mediante la adición de nuevas restricciones a la misma. Dichas restricciones describen necesidades del usuario que deben ser satisfechas por el conjunto de productos recomendados. Un problema que intentan solucionar estos sistemas es que una búsqueda en los catálogos de productos no obtenga ningún resultado. Se proponen soluciones como la orientación del usuario cuando agrega restricciones o la detección y eliminación de restricciones conflictivas.
- **Relaciones de preferencia.** En los SRBCono se pueden utilizar relaciones de preferencia. Una relación de preferencia describe si un usuario prefiere una alternativa A sobre otra alternativa B. Existen tres relaciones de preferencia: indiferencia, preferencia débil y preferencia estricta.

⁵<http://uri.gbv.de/ontology/daia/>

⁶<http://www.movieontology.org/>

Estos sistemas realizan un proceso de toma de decisiones utilizando la información de preferencia que el usuario ha dado sobre los productos, de manera que el usuario define las relaciones de preferencia entre los productos de un conjunto de productos que conoce y el sistema realiza las recomendaciones a partir de dicha información.

Para proporcionar relaciones de preferencia entre todas las alternativas de un conjunto se deben dar P^2 evaluaciones (una relación de preferencia por cada par de alternativas) por lo que esta tarea puede ser tediosa si el conjunto es grande. En [105] se propone un modelo que sólo necesita un subconjunto de las posibles relaciones de preferencia para completarlo y calcular las recomendaciones.

Un inconveniente de los SRBCono es que, como información inicial, requieren al usuario que proporcione ejemplos de productos similares a los que busca. Si el catálogo de productos que el sistema proporciona es extenso, la búsqueda de dichos productos iniciales puede resultar una tarea difícil para el usuario.

Estos sistemas tienen la ventaja de poder funcionar correctamente con pequeñas cantidades de información del usuario, por lo que se ajustan a un entorno de recomendación a corto plazo. A menudo se han utilizado como complemento a otras técnicas de recomendación, en entornos en los que las otras técnicas no pueden realizar recomendaciones correctas a usuarios [20, 105].

2.3.5. Sistemas de Recomendación Basados en Utilidad

En los Sistemas de Recomendación Basados en Utilidad (SRBU) la información que se tiene sobre el usuario es una función que el usuario define para otorgar un valor de utilidad a los productos del catálogo. Esta función de utilidad se define mediante la combinación de los valores de los atributos de los productos. El SRBU evalúa todos los productos del catálogo y recomienda los N productos con mayor valor de utilidad al usuario [23, 81].

En estos sistemas se utiliza el esquema propuesto por la Teoría de la Utilidad Multi

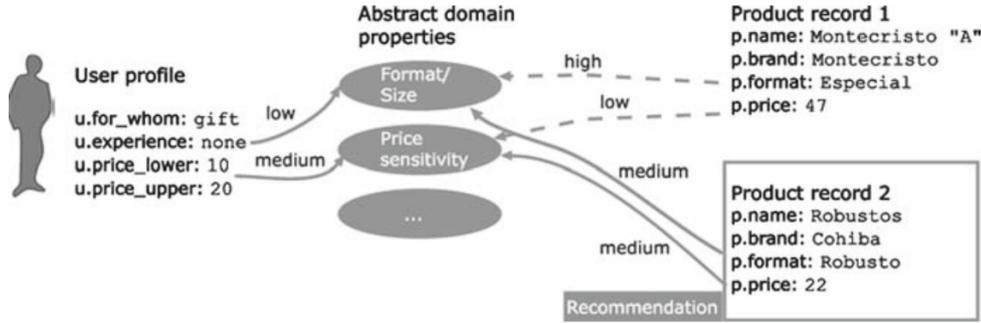


Figura 2.6: Recomendación de productos basada en utilidad

Atributo (en inglés, *Multi Attribute Utility Theory*, MAUT) [92] para el cálculo de la utilidad de los productos para un usuario (ver Ecuación 2.6) en el cual, un decisor puede combinar múltiples criterios para evaluar un conjunto de alternativas. Existen técnicas de este ámbito que permiten generar funciones de utilidad a partir de las preferencias de un decisor [98, 124].

$$u(p_i) = \sum_{c_{ij} \in p_i} w_j * utilidadCaracteristica(c_{ij}) \quad (2.6)$$

donde p_i es el producto que se está evaluando, $c_{ij} \in p_i$ son cada uno de los valores de las características que el producto tiene y w_j es la importancia que el usuario da a la característica j . $utilidadCaracteristica(c_{ij})$ mide cómo de adecuado es el valor que el producto i tiene para la característica j . Esta utilidad se calcula de forma diferente según si el atributo es numérico, categórico, multi-valuado, etc.

Una característica común de estos SR es que tienen un alto grado de interactividad con el usuario, realizando un diálogo con el usuario a través de formulario. De esta manera, el usuario puede especificar las características relevantes de los productos, así como los valores deseados de las mismas.

Algunos trabajos utilizan técnicas avanzadas para la generación de las funciones de utilidad, como algoritmos genéticos [53], reglas de puntuación [112] o realizan un proceso de aprendizaje activo de las preferencias de los usuarios [119].

Un trabajo interesante en SRBU es el de M. Scholz [113], que propone una metodología para el diseño e implementación de SRBU.

Una de las ventajas de esta técnica es que en el cálculo del valor de utilidad se pueden utilizar atributos que no dependen del producto, como la disponibilidad del producto, el tiempo de entrega, la fiabilidad del proveedor, etc. También se pueden modelar soluciones de compromiso como, por ejemplo, si el usuario está dispuesto a pagar un mayor precio por productos con mejores características, una entrega rápida, etc.

El mayor problema de estos sistemas es la dificultad en encontrar una función de utilidad que exprese las necesidades y gustos del usuario.

2.3.6. Sistemas de Recomendación Híbridos

Una forma de mejorar los resultados de recomendación es realizando una hibridación de distintas técnicas. En una hibridación ideal de dos SR A y B , se pretende que el algoritmo resultante presente todas las ventajas que los sistemas hibridados presentan por separado y ninguna de las desventajas (o problemas) de los mismos.

La hibridación se puede realizar de distintas maneras: realizando una combinación de las listas de recomendación, utilizando una media de la valoración predicha por cada sistema, refinando los resultados con otras técnicas o aplicando un SR distinto según el entorno en que se realice la recomendación.

En [23] se hace una clasificación y explicación de las distintas técnicas utilizadas para hibridar dos SR y M. Zanker [135] hace un estudio comparativo en eficacia de algunas de estas hibridaciones sobre técnicas de recomendación básicas. En los siguientes apartados se explican dichas técnicas de hibridación.

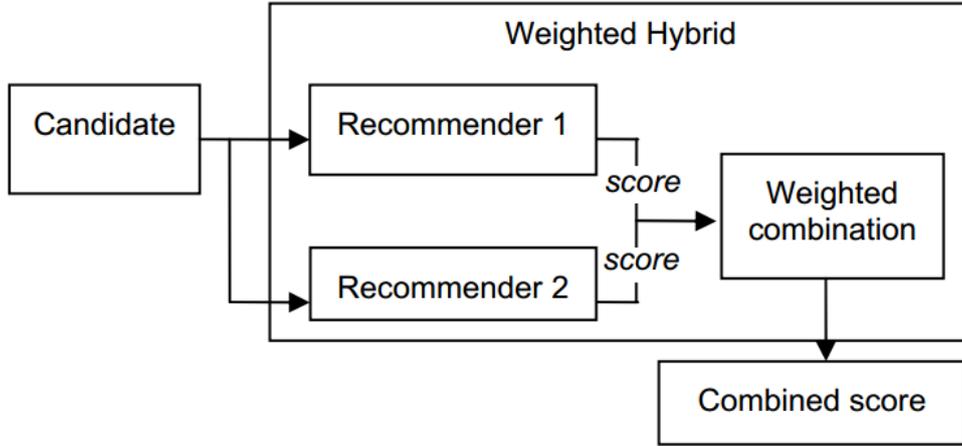


Figura 2.7: Esquema general de hibridación por ponderación

Ponderación (*Weighted*)

La hibridación por ponderación (ver Figura 2.7) consiste en agregar la puntuación que los distintos SR híbridos dan a un producto concreto. Una manera de hacer esto podría ser la combinación lineal o una combinación convexa [43, 120] de las puntuaciones (ver Ecuación 2.7).

$$P_{Híbrido}(u, I) = \alpha \cdot P_{SR_a}(u, I) + (1 - \alpha) \cdot P_{SR_b}(u, I) \quad (2.7)$$

donde α es un parámetro libre en el intervalo $[0, 1]$ que da mayor peso a la técnica SR_a cuanto más cercano a 1 sea su valor.

Al realizar una ponderación mediante una combinación convexa (o cualquier otra combinación basada en la puntuación de recomendación de varias técnicas) se debe realizar una unificación de las puntuaciones para que estén dadas en el mismo intervalo. Supongamos que SR_a es un algoritmo de recomendación colaborativo (ver Apartado 2.3.3) por lo que la puntuación que da a los productos es la valoración que el usuario daría al producto: $P_{SR_a}(u, I) \in [Valor_{min}, Valor_{max}]$; y que SR_b es un SRBU: $P_{SR_b}(u, I) \in [0, 1]$. Si los valores de preferencia están dados en un intervalo distinto de $[0, 1]$ y son mayores, se estaría dando

mayor importancia a la técnica SR_a . Por ello se debe realizar una agregación correcta de las mismas normalizando los intervalos.

También es posible realizar la hibridación por ponderación tomando las recomendaciones como un *ranking* de productos y realizar una combinación de la posición que distintos algoritmos dan a un mismo producto en su lista de recomendaciones.

Otro posible esquema sería tomar la salida de cada SR como un clasificador binario y utilizar las predicciones sobre los productos como votos y agregarlos mediante técnicas de toma de decisiones.

Xinlong Bao y otros [10] utilizan la hibridación de múltiples SR por combinación de características. En su trabajo, proponen la agregación de las salidas de distintos SR utilizando como información adicional medidas de evaluación sobre los algoritmos de recomendación hibridados.

Aunque esta técnica es una forma sencilla de realizar la hibridación, hay que tener en cuenta que el tiempo empleado para calcular las recomendaciones por el sistema hibridado es mayor que la suma de los tiempos de recomendación de cada sistema por separado. Esto puede ser un problema si se hibridan técnicas poco eficientes.

Conmutación (*Switching*)

Una forma de combinar varias técnicas de recomendación es la hibridación por conmutación (ver Figura 2.8). De esta manera, se calculan las recomendaciones utilizando el SR más adecuado según las características del usuario al que se desea recomendar, su entorno o la información que haya disponible sobre el mismo.

Esta técnica de hibridación se ha utilizado para sustituir los SRC en situaciones de arranque en frío en usuarios. R.M. Rodríguez y otros [105] proponen la utilización de SRBCo cuando el perfil del usuario contiene menos de n valoraciones de preferencia sobre los pro-

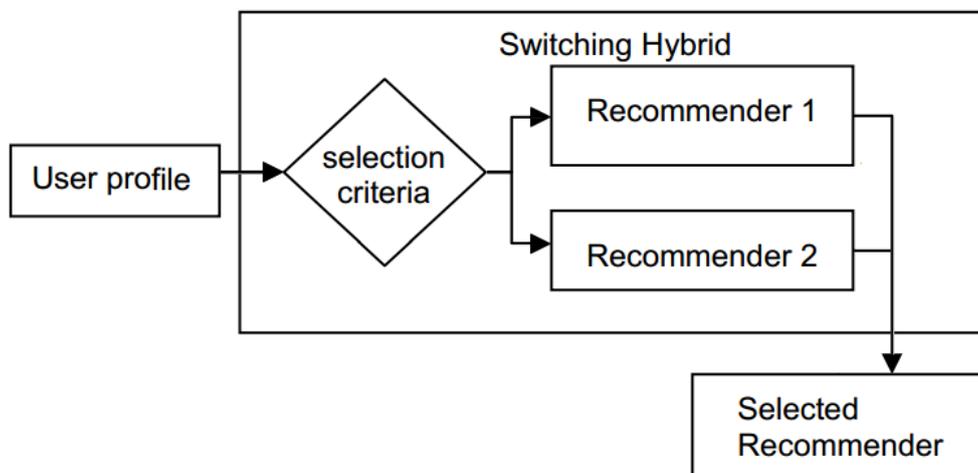


Figura 2.8: Esquema general de hibridación por conmutación

ductos.

M. Ghazanfar y A. Prugel-Bennett [49] utilizan la medida de confianza del clasificador bayesiano, utilizado como SRBCont, para realizar la conmutación entre un SRC con vecinos cercanos basado en productos y éste. También se utiliza el clasificador bayesiano cuando el SRC no encuentra vecinos del producto que se desea predecir.

Ja-Hwung Su y otros [120] realizan también una hibridación por conmutación de SR, cuyo criterio se basa en determinar si el usuario sigue un comportamiento consistente o no.

Esta técnica de hibridación es muy adecuada para ser aplicada en entornos donde la eficiencia sea un factor importante, ya que sólo se calcula la lista de recomendaciones por un sistema de recomendación pero, como inconveniente, se debe determinar el criterio que determine qué algoritmo se utiliza en qué casos.

Múltiples recomendaciones (*Mixed*)

En algunos sistemas puede ser conveniente presentar al usuario varias listas de recomendaciones, cada una calculada con una técnica distinta. Esto deja al usuario la decisión de

qué lista de recomendaciones es más conveniente en cada momento. Por ejemplo, si desea leer un libro parecido a los que ya ha leído se guiará por recomendaciones generadas por un SRBCont y si desea utilizar el sistema para encontrar productos novedosos (recomendación con serendipia) puede utilizar la lista de recomendaciones producida por el SRC.

Si bien esta técnica de recomendación deja la elección del sistema de recomendación al usuario del sistema hay que tener en cuenta que se necesita una alta capacidad de cómputo para que este sistema sea usable. El sistema debe estar configurado para poder calcular recomendaciones de forma paralela y estar preparado para situaciones de alta carga en las horas puntas del sistema.

En Cascada (*Cascade*)

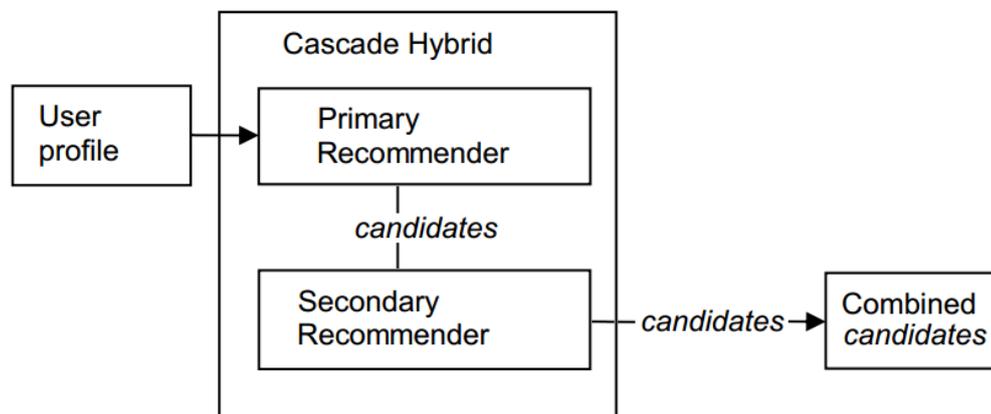


Figura 2.9: Esquema general de hibridación en cascada

Otra manera de hibridar dos SR es utilizar un esquema de hibridación en cascada (ver Figura 2.9). En este esquema, se utiliza un SR para refinar las recomendaciones de otro SR. Se solicita al primer SR (denominada técnica de mayor prioridad) que calcule las recomendaciones para el usuario y se seleccionan los N_1 productos con mayor puntuación. Se solicita al segundo SR (denominada técnica de menor prioridad) que calcule recomendaciones y se seleccionan los N_2 productos más probables. La segunda técnica sólo realiza los cálculos

(predicciones, valor de utilidad, similitud) sobre los N_1 mejores productos que la primera técnica calcula. Para que ésta técnica funcione correctamente N_1 debe ser mayor que N_2 .

En términos de eficiencia, esta técnica resulta muy adecuada, ya que el sistema de recomendación de menor prioridad realiza los cálculos con un subconjunto de los productos del catálogo.

La hibridación en cascada se puede aplicar como un refinamiento de las recomendaciones de un SR. De esta manera, si queremos obtener recomendaciones basadas en conocimiento y utilizar la información colaborativa se aplica primero un SRBCono y sobre los mejores productos de ésta se realiza una reordenación con técnicas colaborativas [23].

Este esquema de hibridación resulta muy útil a la hora de realizar recomendaciones sensibles al contexto. La técnica prioritaria es la encargada de realizar un filtrado que elimine los productos que no tiene sentido recomendar según las condiciones de contexto concretas y la técnica de menor prioridad se encarga de reordenar dichos productos para extraer los que mejor se ajusten a las preferencias del usuario [54, 58].

A. S. Lampropoulos y otros [77] utilizan la hibridación en cascada para realizar recomendaciones de música por género y, tras ello, ajustar el conjunto de productos a la personalidad del usuario.

Combinación de características (*Feature Combination*)

En esta hibridación se combinan las técnicas de recomendación de manera que cada una de las técnicas no es tratada como una caja negra. Esto es porque una hibridación por combinación de características (ver Figura 2.10) utiliza un resultado intermedio de un SR para generar una entrada que sirva a otro SR como características asociadas al perfil de un usuario o producto.

Una hibridación muy utilizada en la literatura es la hibridación de un SRC con un SRB-

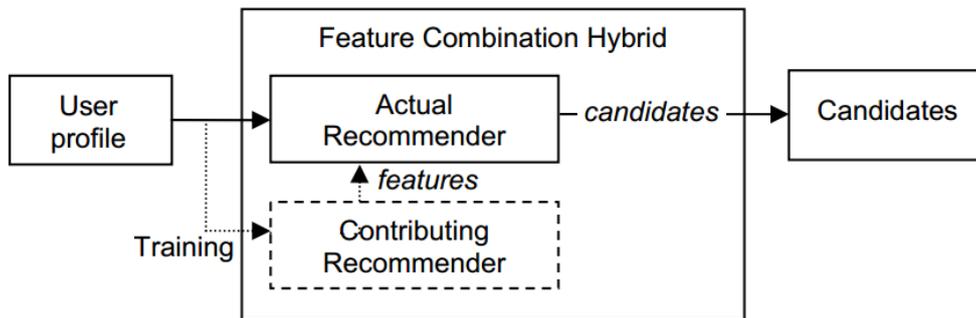


Figura 2.10: Esquema general de hibridación por combinación de características

Cont [9, 37, 106]. Se genera un perfil de usuario con la información de contenido de los productos que han valorado positivamente (SRBCont) y a partir de este se calculan los usuarios vecinos y se realiza la recomendación tal y como se hace en un SRC con vecinos cercanos basado en usuarios.

Multi Nivel (*meta-level*)

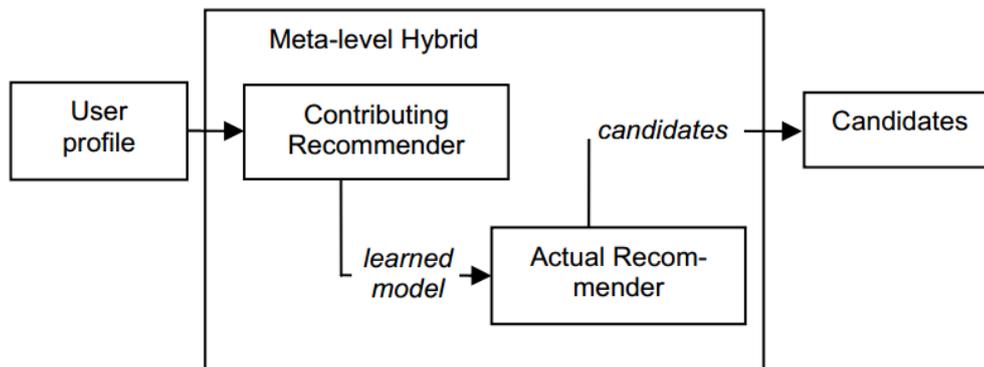


Figura 2.11: Esquema general de hibridación multi nivel

Se puede realizar la hibridación de sistemas de recomendación utilizando como entrada de una técnica el modelo generado por otra. A esta hibridación se denomina Multi Nivel (ver Figura 2.11). Esta técnica se utiliza para generar una representación más rica de los usuarios o productos mediante un SR y mejorar la eficacia de otra técnica de recomendación

más simple o menos precisa [25].

De esta manera, un SR Multi Nivel realiza una hibridación “en serie” es decir, realiza los cálculos necesarios para generar el modelo con el primer SR y después envía los resultados al siguiente para que calcule las recomendaciones.

Un ejemplo de SR Multi Nivel es el propuesto por M.J. Pazzani [97]. En este sistema se utilizan los perfiles generados por un SRBCont con modelado de usuarios en espacios vectoriales (en inglés, *Vectorial Space Modelling*, VSM) como entrada de un algoritmo colaborativo. De esta manera, el SRC utiliza los vectores (perfiles del SRBCont) de los usuarios para calcular las similitudes los mismos y generar los vecindarios.

Un SR Multi Nivel más reciente es el propuesto por M. Zanker [134]. En este trabajo se hibrida un SRC y SRBCono. El primero genera reglas de asociación extraídas de la información proporcionada por los usuarios vecinos al usuario activo y el SRBCono realiza una selección de las reglas que son aplicables a la sesión actual.

Los SR Multi Nivel realizan una hibridación más compleja que el resto de técnicas. Esto es debido a que el segundo SR necesita entender el modelo generado por el primero y aplicarlo en combinación a su esquema de funcionamiento.

2.4. Evaluación de los Sistemas de Recomendación

Cuando se desea implantar un algoritmo de recomendación en un sistema hay que tener en cuenta múltiples factores: tipo de información que el sistema tiene a su disposición, características de los productos que se recomiendan, características de los usuarios a los que se recomienda, etc. Una vez conocidos estos factores, se eligen varias alternativas y se realiza una comparación para determinar cuál de ellas es mas adecuada, dados los valores de dichos factores.

Una herramienta que ayuda en la comparación de distintos algoritmos de recomendación

es el Estudio de Casos. En un Estudio de Casos se pretende evaluar cómo se comportan una serie de alternativas y permite seleccionar cuál de ellas es más adecuada a las necesidades del sistema que se desea implementar.

Para comprobar la validez de los SR se llevan a cabo experimentaciones que comparan qué método de recomendación “funciona mejor” con un conjunto de datos concreto. Este “funciona mejor” puede ser formulado de distintas maneras dependiendo del objetivo que se tenga al elegir un SR. Por ejemplo, la forma más utilizada para evaluar un SRC es medir el error que comete en la predicción. Por lo tanto, la hipótesis a comprobar podría formularse de la siguiente manera: “El Sistema de Recomendación propuesto realiza una mejor predicción de las valoraciones que los SRC con vecinos cercanos”.

Una vez formulada la hipótesis se deben diseñar los experimentos necesarios para comprobarla. Para realizar dicha evaluación se utilizan las experimentaciones, que pueden realizarse de dos formas: *on-line* y *off-line*.

Evaluación *on-line* En la evaluación *on-line* se utiliza un entorno real para la experimentación [71], que en el caso de los SR será el sitio Web donde se utilice. En esta evaluación lo que se pretende medir es si el sistema de recomendación influye en el comportamiento del usuario. Usualmente en este tipo de evaluaciones se utilizan los datos de navegación en el sitio Web y se comprueba si existen diferencias de comportamiento.

Para este tipo de pruebas es recomendable realizar una experimentación de varias alternativas a la vez. Para ello, si queremos comprobar si un SR influye más a los usuarios que un SR propuesto, debemos realizar una experimentación en la que unos usuarios del sistema real reciban recomendaciones de ambos SR. Esto se hace seleccionando aleatoriamente qué usuarios utilizarán cada sistema. Una vez realizada la obtención de los datos se analizan y se comprueba si existen diferencias de comportamiento significativas en cada configuración.

Pero para realizar una evaluación *on-line* correcta se deben realizar experimentos que prueben las distintas alternativas en un entorno real. En un SR en funcionamiento existen

tres fuentes de influencia: las intenciones del usuario, su contexto y la interfaz mediante la cual se presentan las recomendaciones. Por lo tanto, si queremos comprobar qué configuración de un algoritmo funciona mejor, debemos mantener el resto de factores fijos, como la interfaz. Otro problema es la influencia de factores contextuales, como pueden ser el día de la semana o día del mes en los que se realiza el experimento, por lo que hay que diseñar el experimento de manera que los resultados no estén sesgados.

F. Hernández y otros [56] proponen una forma de evaluar los sistemas de recomendación en un entorno *on-line*, utilizando para ello medidas sobre el comportamiento del usuario en la Web. Estas medidas tratan de obtener un valor de cómo de útiles son las recomendaciones del sistema, si el usuario las utiliza, si finalmente compra un producto que el sistema le recomendó, etc.

Los experimentos *on-line* deben ser diseñados con especial cuidado, ya que sus efectos repercuten en el negocio real. Por ejemplo, si en un experimento *on-line* se prueba un algoritmo de recomendación que no funciona correctamente, los usuarios lo notarán y su confianza en las recomendaciones puede verse afectada.

Evaluación *off-line* Con las experimentaciones *off-line* se intenta simular el comportamiento de los usuarios reales. De esta manera, los resultados obtenidos en la experimentación se pueden generalizar al sistema en el entorno real.

Existen conjuntos de datos públicos para la evaluación de SR, como son Netflix, MovieLens, Book-Crossing⁷, Jester⁸. Estos conjuntos de datos contienen las valoraciones de preferencia que los usuarios han hecho sobre los productos, así como información sobre los productos que se están valorando.

Para realizar una evaluación *off-line* correcta, se debe realizar un proceso que refleje correctamente la eficacia del SR, evitando que los resultados estén sesgados de alguna manera.

⁷<http://www.informatik.uni-freiburg.de/~cziegler/BX/>

⁸<http://goldberg.berkeley.edu/jester-data/>

Por ello, se aplican técnicas de evaluación que determinan qué datos (valoraciones de preferencia) utiliza el SR para la generación del modelo (perfiles de usuarios, perfiles de productos, etc.) y cuáles de ellos debe predecir a partir del modelo generado.

El resto de este apartado se distribuye de la siguiente manera. En el Apartado 2.4.1 se describen las técnicas empleadas para realizar evaluaciones *off-line* de SR. En el Apartado 2.4.2 se describen las medidas de evaluación aplicadas a los SR.

2.4.1. Protocolos de evaluación

En el área del Aprendizaje Automático (en inglés *Machine Learning*) se aplican técnicas de evaluación *off-line* para determinar si un algoritmo es correcto, para comparar su eficacia con otros algoritmos, etc.

Un algoritmo de Aprendizaje Automático se utiliza para la predicción de datos a partir de información conocida. Para ello, un algoritmo consta de dos fases:

- **Construcción del modelo.** El algoritmo utiliza la información conocida para generar un modelo que describa o explique los datos.
- **Fase de explotación.** Una vez construido el modelo se pasa a la fase de explotación. En esta fase se utiliza el modelo para realizar predicciones sobre datos que el algoritmo no conocía en la fase de construcción del modelo.

Para que los datos de evaluación sean fiables, se deben realizar múltiples ejecuciones y calcular la media de las medidas de evaluación. Esto es porque los protocolos de evaluación eligen los datos en el conjunto de entrenamiento y de evaluación de manera aleatoria. Realizando múltiples ejecuciones disminuye la probabilidad de que los resultados del algoritmo sean mejores o peores debido a los datos que se hayan elegido para entrenamiento y evaluación.

Para ello se aplican protocolos de evaluación. Un protocolo de evaluación determina qué datos conoce un algoritmo durante la construcción del modelo (fase de entrenamiento) y qué datos debe calcular a partir del mismo (fase de evaluación). Con estas predicciones se calcula si el algoritmo es eficaz y en qué medida.

Hold out

En el protocolo de evaluación *Hold out* se realiza una división de los datos aleatoria. Para ello, se debe establecer qué porcentaje de los datos se desea que esté en el conjunto de entrenamiento.

Una configuración ampliamente utilizada para Hold Out en la evaluación de algoritmos de SR es 80% entrenamiento, 20% evaluación. Esto quiere decir que el 80% de los datos iniciales se utilizarán para la generación del modelo y el 20% restante se utilizará para comprobar la eficacia del algoritmo.

Validación cruzada

El protocolo de evaluación de Validación Cruzada (en inglés, *Cross fold validation*) se utiliza para generar múltiples conjuntos de datos a partir de un conjunto de datos original. Para ello, se hace una división del conjunto de datos inicial en k particiones de forma aleatoria. Se generarán k pares de conjuntos de entrenamiento y evaluación de manera que cada uno tenga una partición como conjunto de evaluación y el resto de datos como conjunto de entrenamiento (ver Figura 2.12)

Los valores de k más usados en la literatura para la evaluación de SR es $k = 5$. Con esta configuración se realizan 5 particiones del conjunto de datos inicial y se construirán 5 pares de conjuntos de entrenamiento y evaluación. Una vez construidos los conjuntos se realizarán 5 ejecuciones del algoritmo a evaluar.

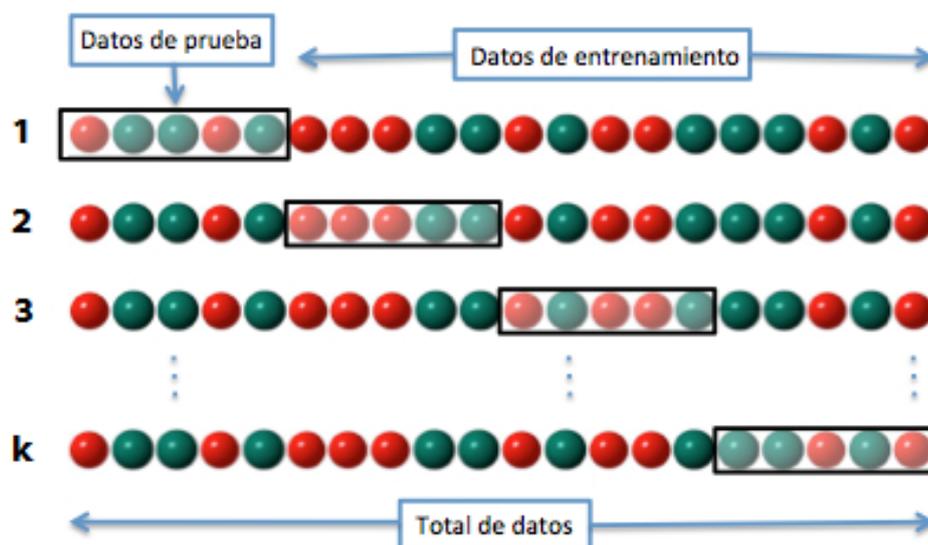


Figura 2.12: Protocolo de evaluación Validación Cruzada

Hay que tener en cuenta, al elegir la configuración de este método de evaluación, que se realizan k ejecuciones del algoritmo. Si el conjunto de datos es grande y el algoritmo demora mucho tiempo en cada ejecución, el experimento tardará k veces más.

Para asegurar que los datos no están sesgados, se deben realizar múltiples ejecuciones de la evaluación. Por lo tanto, si deseamos evaluar un algoritmo con un experimento en el que se realicen 50 ejecuciones de la validación cruzada con $k = 5$, se realizarán 250 ejecuciones concretas⁹ del algoritmo.

Protocolos de predicción en Sistemas de Recomendación Predictivos¹⁰

Una forma diferente de realizar el proceso de evaluación es simulando un comportamiento concreto del usuario mediante la selección de los datos. De esta manera, se puede simular

⁹ Una ejecución concreta es realizar el proceso de entrenamiento y evaluación con un par de conjuntos de entrenamiento-evaluación.

¹⁰Un SR predictivo es aquel que para recomendar productos predice el valor de preferencia que el usuario les daría.

que un usuario haya valorado un número mínimo de productos o que se prediga un número fijo de valoraciones por usuario. Para ello, existen los siguientes protocolos de evaluación:

- **Todos menos N (*All but N*):** Antes de calcular las recomendaciones para un usuario, se extraen aleatoriamente N valoraciones de preferencia, que son las que se predicen a partir del resto. De esta manera se mantiene constante el número de productos ocultos y, por lo tanto, el número de productos que se predicen para cada usuario. Este protocolo de evaluación se utiliza para experimentaciones que revelen los resultados de eficacia del sistema si el usuario valora N productos más.
- **Dadas N valoraciones (*Given N*):** En este protocolo se evalúa el SR de manera que los usuarios tengan un número constante de valoraciones en el momento de la predicción. Para ello, se seleccionan aleatoriamente N valoraciones y se eliminan el resto. Éstas últimas son las valoraciones que el sistema debe predecir. Este protocolo de evaluación se utiliza para conocer qué número mínimo de valoraciones debe tener un usuario para que el SR funcione correctamente.

Otra manera de simular el comportamiento del usuario es utilizando el tiempo. Si se tiene disponible una marca de tiempo de cada valoración de preferencia, se pueden aplicar protocolos que simulen, de una manera más real, el comportamiento del usuario. Por ejemplo, si queremos evaluar el SR desde el registro del usuario, podemos realizar predicciones de productos a un usuario sin conocimiento, con una valoración, con dos valoraciones y así sucesivamente. Otra manera es simular la recomendación al usuario en un punto concreto del tiempo, es decir, elegir una fecha y extraer las valoraciones posteriores, que son las que se predicen.

Para decidir qué protocolo de evaluación se utilizará en un caso de estudio, se debe fijar el propósito del mismo. Por ejemplo, podemos realizar un caso de estudio para evaluar si un SR necesita una mayor cantidad de información para realizar buenas recomendaciones. Además de esto, se debe tener en cuenta también el tipo de SR que se evalúa.

Por ejemplo, podríamos evaluar el error de predicción de un SRC *User-User*¹¹ utilizando la validación cruzada, pero tiene poco sentido aplicar este protocolo, ya que este SR no genera un modelo. En este caso, tendría mas sentido aplicar el protocolo de evaluación *Todos Menos 1*.

Otro caso especial son los SRC *Item-Item*¹². En este tipo de sistemas se puede realizar la evaluación dejando fuera de los datos de entrenamiento todas las valoraciones de un usuario. Esto es posible porque el modelo que genera se basa en buscar similitudes entre productos, por lo que no necesita información del usuario en su momento de creación. Por lo tanto, en esta fase se podría aplicar validación cruzada por usuarios y, en la predicción, aplicar el protocolo de predicción *Dados N*.

2.4.2. Medidas de evaluación

Una vez realizada la recomendación en el proceso de evaluación se analizan los resultados con las medidas de evaluación. Una medida de evaluación asigna una puntuación a los resultados del SR. En función de la medida de evaluación que se aplique, dicha puntuación puede ser el porcentaje de aciertos, el error que comete al predecir o el tiempo que tarda en calcular las recomendaciones.

Las medidas de evaluación se deben elegir teniendo en cuenta la característica del algoritmo se desee comprobar. También hay que tener en cuenta el tipo de algoritmo de recomendación que va a ser evaluado, ya que ciertas medidas de evaluación sólo se pueden aplicar a algoritmos con ciertas características.

En los siguientes apartados se exponen los distintos enfoques para aplicar distintas medidas de evaluación, explicando su funcionamiento y en qué casos se pueden aplicar.

¹¹SRC con vecinos cercanos basado en memoria

¹²SRC con vecinos cercanos basado en modelo

2.4.3. Predicción de valoraciones

Existen SR que realizan una predicción de valoraciones como resultado para el cálculo de las recomendaciones. Una forma de comprobar la eficacia de estos sistemas es calculando una medida del error de predicción. A continuación se exponen las medidas más usadas para la evaluación de la predicción de los SR: el error absoluto medio, el error cuadrático medio y la cobertura.

Error Absoluto Medio (MAE)

El Error Absoluto Medio (en inglés *Mean Absolute Error*, *MAE*) es el error que el SR comete al realizar las predicciones de las valoraciones de preferencia. Por lo tanto, cuanto menor es el MAE de un SR, mejor es su eficacia. Se calcula de la siguiente manera:

$$MAE = \frac{\sum_{i=1}^N |p_i - v_i|}{N}$$

donde p_i es la predicción, v_i es el valor de preferencia real y N es el número de predicciones que se calculan.

Error Cuadrático Medio (MSE)

El Error Cuadrático Medio (en inglés *Mean Squared Error*, *MSE*), es la media del error que el SR comete, al cuadrado, por lo que esta medida penaliza errores de predicción mayores. Al ser una medida del error, cuanto menor sea el MSE, más eficaz es el SR. Se calcula de la siguiente manera:

$$MSE = \frac{\sum_{i=1}^N (p_i - v_i)^2}{N}$$

donde p_i es la predicción, v_i es el valor de preferencia real y N es el número de predicciones que se calculan.

Cobertura

La cobertura es el porcentaje de predicciones con éxito de un SR, es decir, el número de predicciones que puede calcular entre el total de predicciones solicitadas. Por lo tanto, cuanto mayor sea la cobertura, mejor es el SR.

Por ejemplo, cuando intentamos predecir la valoración de un producto mediante el SRC User-User y ningún vecino del usuario activo ha valorado dicho producto, se produce un fallo de cobertura. Intuitivamente, si el número de vecinos elegido en este algoritmo es mayor, la cobertura aumenta.

Esta medida se utiliza como complemento a las medidas de error, ya que puede suceder que un SR prediga las valoraciones muy bien, pero que sólo sea capaz de calcular un porcentaje muy bajo de predicciones.

2.4.4. Clasificación binaria de productos

Los resultados de un SR pueden ser interpretados como un proceso de clasificación binaria de productos. Un SR se encarga de discriminar los productos que pertenecen a la clase positiva¹³ de los que pertenecen a la clase negativa.

En SRC esta discriminación se puede hacer de distintas maneras. Si la información que hay disponible sobre el usuario son los productos que compró, estos pertenecen a la clase positiva y el resto a la clase negativa. Si las valoraciones están definidas en el dominio real, se puede hacer mediante un umbral. Por ejemplo, si la escala de valoración es de 1 (peor valoración) a 5 (mejor valoración) podemos considerar que los valores iguales o mayores a 4 son productos que pertenecerán a la clase positiva para un usuario dado.

Una vez se ha aplicado el criterio de relevancia más adecuado, se debe decidir qué productos clasifica el SR como clase positiva. En SRC se realiza una predicción de la valoración,

¹³En SR, los productos en la clase positiva son los que tienen interés para el usuario

por lo que se puede aplicar el mismo criterio que para la valoración de preferencia.

Pero, en los sistemas de recomendación que no realizan predicción de la valoración, se calcula de una manera distinta. Se toma la lista de todas las recomendaciones como una colección de soluciones de recomendación y el umbral, en este caso, será el número de productos que el SR recomienda. De esta manera, los n primeros productos se considera que pertenecen a la clase positiva y el resto a la clase negativa.

Una vez decidido esto, se pasa a calcular la matriz de confusión (ver Tabla 2.2). Esto se hace calculando el valor de las celdas, según el resultado del sistema y la información en el conjunto de evaluación:

- tp (*True-Positive*) es el número de productos clasificados correctamente en la clase positiva, es decir, que son interesantes para el usuario y el SR los clasifica como tales.
- fp (*False-Positive*) es el número de productos que el SR clasifica mal como clase positiva. Son productos que no deberían haber sido recomendados.
- fn (*False-Negative*) es el número de productos que deberían haber sido recomendados pero que el sistema no recomendó.
- tn (*True-Negative*) es el número de productos clasificados correctamente en la clase negativa, es decir, productos que no han sido recomendados y que no tienen interés para el usuario.

		Valoración	
		Positiva	Negativa
Predicción	Positiva	tp	fp
	Negativa	fn	tn

Tabla 2.2: Matriz de confusión

A continuación se detallan las medidas de evaluación que utilizan la matriz de confusión.

Precisión

La precisión es el porcentaje de aciertos del SR en la clasificación de los productos en la clase positiva. Por lo tanto, cuanto mayor sea la precisión de un SR, mejor es su eficacia. Se calcula de la siguiente manera:

$$precisión = \frac{tp}{tp + fp}$$

El valor de la precisión disminuye cuantos más productos recomiende el SR, ya que aumenta el número de productos que el sistema clasifica como positivos, aumentando la probabilidad de que productos negativos se clasifiquen como positivos.

Recall

El recall es el porcentaje de productos que el usuario recomienda de entre todos los que son útiles al usuario. Por lo tanto, cuanto mayor sea su valor, mejor es la eficacia del SR. Se calcula de la siguiente manera:

$$recall = \frac{tp}{tp + fn}$$

El recall aumenta su valor cuantos más productos recomiende el SR, ya que se reduce el número de productos que el sistema clasifica como negativos.

FPR (*False Positive Rate*)

El FPR es el porcentaje de falsos positivos que el SR comete. Por lo tanto, cuanto menor sea su valor, más eficaz es el SR. Se calcula de la siguiente manera:

$$FPR = \frac{fp}{fp + tn}$$

El FPR aumenta cuantos más productos recomiende el SR, ya que aumenta el número de productos que el sistema clasifica como positivos.

F_β -medida

La precisión y el recall por sí mismos no dan una medida completa de la eficacia de un SR. La F_β -medida es una combinación de la precisión y el recall:

$$F_\beta = (1 + \beta^2) \frac{\text{precisión} \cdot \text{recall}}{\beta^2 \text{precisión} + \text{recall}}$$

donde β es un parámetro libre, que indica la importancia que se da a la precisión sobre el recall. Cuanto mayor es β , mayor importancia se da a la precisión.

Área bajo ROC

Utilizando el número de recomendaciones n como un umbral se puede calcular la curva ROC (en inglés, *Receiver Operating Characteristic*). De esta manera, con distinto k se obtienen matrices de confusión distintas. Si calculamos la matriz de confusión para todos los valores de k (desde uno hasta el número de productos predichos) y sobre éstas calculamos el FPR y el recall, obtenemos una lista de pares¹⁴. Cuando se representa esta lista de pares en una gráfica, representando en el eje X el FPR, se obtiene una curva que da información de cómo se comporta el SR si cambiamos el umbral de recomendación (ver Figura 2.13).

La curva ROC es mejor cuanto más se acerque al punto (0,1), por lo que si calculamos el área bajo esta curva, obtenemos una medida de cuánto se acerca a la curva ROC perfecta, cuya área vale 1. La curva marcada con línea discontinua representa la curva ROC de un SR aleatorio y su área vale 0,5. Por lo tanto, una forma de comprobar la eficacia de un SR es mediante el área bajo ROC.

2.4.5. *Ranking* de productos

La recomendación de productos puede ser entendida como la presentación de una lista ordenada de productos, en orden decreciente, por utilidad para el usuario. En este tipo de

¹⁴ $\langle FPR_k, recall_k \rangle$

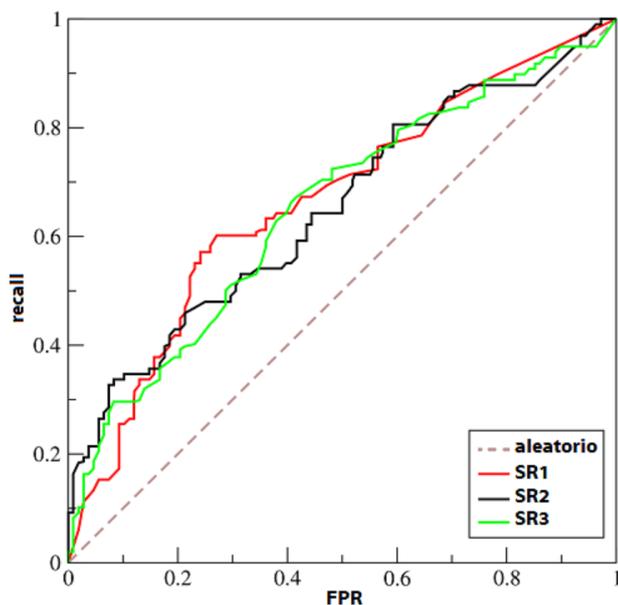


Figura 2.13: Curva ROC de distintos Sistemas de Recomendación

medidas de evaluación se da una puntuación de cómo de correcta es la ordenación que hace un SR.

A continuación se exponen medidas para evaluar los SR teniendo en cuenta la ordenación de recomendaciones que hace.

Normalized Distance-based Performance Measure (NDPM)

La NDPM [133] es una medida de distancia entre ordenaciones que se utiliza cuando se tiene un *ranking* de referencia en el que hay empates. Esto se puede producir cuando el *ranking* se genera mediante la valoración de preferencia del usuario, cuando se da por valores enteros. Por ejemplo, un dominio de valoración ampliamente usado es asignar estrellas, donde 5 estrellas significa que al usuario le gusta mucho el producto y 1 que no le ha gustado nada. Por lo tanto, si realizamos una ordenación de los productos por número de estrellas y hay varios productos valorados con 5 estrellas, se produce un empate. Esta medida tiene como peculiaridad que no penaliza al SR por asignar un orden estricto a los productos.

El valor de NDMP viene dado por la Fórmula 2.8 y vale 0 cuando el SR establece todas las relaciones de preferencia correctamente y 1 cuando contradice las relaciones de preferencia del *ranking* de referencia.

$$NDPM = \frac{C^- + 0,5C^{u0}}{C^u} \quad (2.8)$$

donde C^- es el número de productos en los que los *ranking* no coinciden, C^{u0} es $C^u + C^s$. C^s es el número de empates en el *ranking* de referencia y C^u en el *ranking* del sistema.

Coefficiente τ de Kendall

El coeficiente de τ de Kendall se utiliza en el mismo caso que el NDMP, pero tiene más sentido que este en situaciones donde el usuario solo da valoraciones positiva o negativa. Se calcula de la siguiente manera:

$$\tau = \frac{C^+ - C^-}{\sqrt{C^u} \sqrt{C^s}} \quad (2.9)$$

2.4.6. Medidas de eficiencia

La eficiencia se da cuando se realiza una tarea correctamente utilizando la cantidad mínima de recursos. La eficiencia en SR puede ser medida de dos maneras: memoria y tiempo.

Memoria

Para ejecutar un algoritmo, se necesita ubicar los datos con los que trabaja en un dispositivo de memoria. Podemos medir la memoria que consume un SR para realizar las recomendaciones de dos maneras distintas:

- **Tamaño del modelo.** Se puede tener una medida del consumo en memoria de un SR utilizando el modelo generado. Para realizar las recomendaciones, se debe almacenar en una base de datos, en un fichero, etc.

- **Memoria en ejecución.** Para calcular el coste de implantación de un SR en una Web hay que calcular la memoria RAM que necesita el algoritmo cuando calcula las recomendaciones. Para ello se mide la memoria máxima que el SR utiliza cuando recomienda productos a un usuario.

Las medidas de eficiencia en memoria son críticas en sistemas en los que se dispone de una gran cantidad de datos sobre los usuarios.

Tiempo

Un aspecto interesante de la eficiencia de los SR es el consumo en tiempo. Para que la recomendación sea útil para el usuario, debe poder hacerse en un tiempo que permita la interactividad con el sistema. Por ello, el tiempo medio de recomendación por usuario no debe superar cierto umbral.

También se puede medir la eficiencia en tiempo de un SR mediante el *Throughput*. El *Throughput* mide la cantidad de información que un sistema puede procesar por unidad de tiempo. Aplicado a SR, el *Throughput* se calcula dividiendo el número de recomendaciones que se calculan por el tiempo que se consume al calcularlas.

Otra manera de medir la eficiencia en tiempo de un SR es calculando el tiempo que tarda en construir el modelo.

2.5. Temas emergentes

El área de los Sistemas de Recomendación es un área muy activa en investigación. Actualmente existen múltiples líneas de investigación relacionadas con los SR. F. Ricci y otros [102] hacen una revisión de los temas emergentes y los nuevos retos de los SR:

- **Sistemas de Recomendación Sensibles al Contexto.** Los Sistemas de Recomen-
-

dación Sensibles al Contexto (SRSC) se utilizan en dominios donde no todos los productos son igualmente útiles para el usuario en cualquier situación. Estos sistemas usan información contextual del usuario para ajustar las recomendaciones de una forma más práctica.

Por ejemplo, un usuario que se encuentre en Jaén decide utilizar un SR para buscar un restaurante en el que cenar, el cual tiene un catálogo de restaurantes de Andalucía. Un SR sin información contextual recomendaría el restaurante que más se ajustase a sus gustos, sin importar su ubicación respecto del usuario. Sin embargo, los restaurantes muy lejanos (por ejemplo, que no se encuentren en la misma ciudad) no serían útiles para el usuario.

Para incorporar la información contextual se pueden utilizar tres enfoques distintos [2]:

- **Pre-filtrado.** En este enfoque se realiza una selección de los datos de entrada que el SR utiliza en el proceso de recomendación. Por ejemplo, en un SR de películas podemos utilizar el momento de la semana (día laborable o fin de semana) para realizar recomendaciones más ajustadas al momento en que deseamos ver la película. Para ello, utilizará sólo valoraciones de películas que fueron vistas en el mismo contexto en que se encuentra el usuario.
 - **Post-filtrado.** En este enfoque, la información contextual se aplica después de calcular las recomendaciones. Una vez obtenida la lista, se reajustan los resultados. Para ello se puede realizar una reordenación teniendo en cuenta el contexto, eliminar productos que no tenga sentido recomendar en el contexto concreto, etc.
 - **Modelado de con información contextual.** Se puede utilizar un SRSC en lugar de realizar un ajuste (previo o posterior) de un SR sin información contextual. De esta manera, se utiliza la información contextual en el algoritmo que calcula las recomendaciones, es decir, como un dato más del perfil de usuario.
- **Visualización de las Recomendaciones.** Tradicionalmente, los SR presentan las recomendaciones en forma de una lista ordenada de productos. Sin embargo este tipo
-

de presentación de las recomendaciones no permite mostrar si dos productos recomendados son parecidos entre sí o no. Esto se puede realizar por medio de un diagrama bidimensional, en el que se representarán cerca los productos recomendados que sean más similares entre sí (ver Figura 2.14).



Figura 2.14: Visualización de las recomendaciones

M. Kagie y otros [68] analizan distintas técnicas de representación de objetos complejos en diagramas bidimensionales, como los *self organizing maps* [72], los mapas jerárquicos (*tree maps*), el escalado multidimensional o el análisis de componentes principales.

- **Recuperación de información personalizada.** Cada vez más, los buscadores recogen información de uso, información sobre los usuarios que los utilizan, sobre las consultas más frecuentes, etc. Esta información puede ser utilizada para mejorar los procesos de recuperación de información, realizando una personalización de los resultados de las búsquedas.
 - **Sistemas de Recomendación con redes de confianza (*Trust-based*).** Estos
-

sistemas utilizan información de fuentes externas para elaborar una red de confianza que permita mejorar el proceso de selección de vecinos en SRC.

Una fuente de información para construir las redes de confianza de estos SR son las redes sociales. En ellas, pueden existir relaciones entre usuarios unidireccionales (seguidor) o bidireccionales (amigo, compañero de trabajo). Una vez construida la red de confianza, se pretende predecir la confianza entre dos usuarios que no están directamente relacionados a través de confianza.

Por ejemplo, se pueden utilizar las redes de confianza para mejorar la formación del vecindario de los usuarios en SRC [127].

- **Sistemas de Recomendación con Etiquetado Social.** El etiquetado social es un tema con gran crecimiento debido al uso de la Web 2.0. Consiste en la posibilidad de que los usuarios asignen etiquetas (palabras clave) a los productos de un catálogo. Esto puede producir una sobrecarga de información, ya que en un sistema de etiquetado social participan muchos usuarios. Por ello, se utilizan los SR para ayudar al usuario a encontrar productos relevantes, utilizando las etiquetas.

El modelo que soporta el etiquetado social son las *folksonomías*. Una *folksonomía* está definida por un conjunto de usuarios U , un conjunto de etiquetas E , un conjunto de productos P y el conjunto de las etiquetas que cada usuario ha asignado a cada producto $Y \subset U \times E \times P$. Este modelo es más complejo que el utilizado por los SR tradicionales, por lo que los algoritmos utilizados para SR con etiquetado social tienen mayor complejidad.

L.B.Marinho y otros [89] realizan una revisión de los métodos que se aplican en SR con etiquetado social. Exponen los distintos modos de funcionamiento que puede tener un sistema con etiquetado social:

- **Recomendación de usuarios:** en este modo, el SR recomienda usuarios parecidos al usuario activo. De esta manera, el usuario activo puede encontrar personas con intereses afines a los suyos.
-

- Recomendación de productos: este es el modo tradicional de funcionamiento de los SR, pero en sistemas con etiquetado social se puede aprovechar esta información para mejorar la eficacia del proceso.
- Recomendación de etiquetas: este modo es útil para los usuarios, ya que los ayuda a completar la relación de etiquetas asociadas a un producto.

■ **Recomendación a Grupos.**

Los SR para Grupos, en lugar de recomendar a un usuario, realizan recomendaciones a grupos de usuarios. Existen situaciones en las que esto es útil, por ejemplo, cuando un grupo de amigos decide ver una película juntos. El sistema puede ayudarlos a decidir qué película es más adecuada, ya que tiene en cuenta las preferencias de cada individuo.

Recomendar productos a grupos de usuarios es más complejo que recomendar a un individuo. Aunque el SR conozca perfectamente las preferencias de cada uno, existen distintas formas de agregarlas para construir un perfil asociado al grupo [91]. Además del problema comentado anteriormente, un SR para Grupos se debe enfrentar a problemas como recoger las preferencias de cada individuo, detectar qué usuarios están presentes o explicar por qué recomienda ciertos productos.

- **Sistemas de Recomendación Multi-criterio.** En los SR se tiene en cuenta un valor de preferencia general para realizar las recomendaciones, pero los usuarios pueden asignar las valoraciones utilizando criterios distintos. Los SR Multi-criterio (SRMC) permiten al usuario valorar los productos en distintos aspectos¹⁵. De esta manera, la utilidad de un producto para un usuario se determina mediante múltiples criterios.

Para ello, se pueden aplicar técnicas de Toma de Decisión Multi Criterio (TDMC) como la desagregación de preferencias. Esta técnica permite obtener valoraciones de los productos en los distintos criterios a partir de la información que se tiene sobre los usuarios.

- **Seguridad en Sistemas de Recomendación.** Esta línea de investigación trata de

¹⁵Por ejemplo, en el caso de una película: guión, efectos especiales, interpretación, etc.

detectar usos fraudulentos de los SR. En [27] se define la *robustez* de un SR como la capacidad de realizar buenas recomendaciones en presencia de datos erróneos. También realiza un análisis del impacto en la eficacia de distintos ataques en un SR¹⁶. Existen dos tipos de ataques a SR:

- **Ataques al SR:** Este tipo de ataques tienen como objetivo reducir la efectividad general del SR. Para ello, se realiza una inserción de perfiles de usuario falsos en la base de datos que el SR utiliza.
- **Ataques a productos concretos:** con este ataque se pretende que un producto concreto sea recomendado con mayor probabilidad o que sea clasificado como un producto no recomendable, según el objetivo del ataque. Para ello, se insertan perfiles de usuario falsos en la base de datos con valoraciones altas (o bajas) en el producto objetivo y valoraciones generadas de alguna manera (aleatoriamente, valoración media) en el resto de productos.

¹⁶Se utiliza un SRC con vecinos cercanos basado en memoria

Sistemas de Recomendación Basados en Contenido Con Modelado en Espacios Vectoriales

Como se vió en el Apartado 2.3.2 existen distintas técnicas para realizar el proceso de recomendación basada en contenido. Un conjunto de técnicas de interés para este trabajo son los SRBCont con Modelado en Espacios Vectoriales, ya que es la técnica utilizada en nuestra propuesta. Nos interesan especialmente los SRBCont que trabajan con contenidos multivaluados, es decir, contenidos que se definen por un conjunto de características, cada una de las cuales puede tomar distintos valores dentro de un dominio; ya que nuestra propuesta trabaja en dicho contexto. Por ello, se describen los sistemas clásicos para contenidos multivaluados.

En este capítulo se explica el funcionamiento de las técnicas básicas de recomendación basada en contenido con Modelado en Espacios Vectoriales. Primero, se explica el filtrado basado en contenido con modelado booleano básico. Tras esto, se explica una mejora al mismo mediante el filtrado basado en contenido con ponderación de características basada en TF-IDF (*Term Frequency - Inverse Document Frequency*) [122].

3.1. Modelo Booleano Básico

En este apartado se explica el funcionamiento del SRBCont con Modelado en Espacios Vectoriales Booleano Básico (MEV_BB). Aplicado a la recomendación basada en contenido, el Modelado en Espacios Vectoriales se aplica calculando un vector que describe el perfil de cada usuario (P_u) y de cada producto (a_i). Una vez calculados los vectores, se realizan operaciones sobre los mismos para dar una medida que estima qué producto (o productos) es más adecuado para un usuario.

En la Tabla 3.1 se muestra la información que este sistema de recomendación utiliza y calcula:

- $a_{x_{u,1}}, \dots, a_{x_{u,nu}}$ son los productos que el usuario u ha valorado. $x_{u,1}$ denota el primer producto que el usuario u valoró.
- r_1^u, \dots, r_{nu}^u son los respectivos valores de preferencia que el usuario ha dado.
- c_1, \dots, c_m son las características que tienen dichos productos.
- $v_1^{x_{u,1}}, \dots, v_m^{x_{u,1}}$ son los valores de las características para el producto $x_{u,1}$
- $P_u = p_1^u, \dots, p_m^u$ denota los valores del perfil del usuario u para cada característica.

	c_1	\dots	c_m	R_u
$a_{x_{u,1}}$	$v_1^{x_{u,1}}$	\dots	$v_m^{x_{u,1}}$	r_1^u
\dots	\dots	\dots	\dots	\dots
$a_{x_{u,nu}}$	$v_1^{x_{u,nu}}$	\dots	$v_m^{x_{u,nu}}$	r_{nu}^u
P_u	p_1^u	\dots	p_m^u	

Tabla 3.1: Información que utiliza un Sistema de Recomendación Basado en Contenido con Modelado Booleano Básico

El algoritmo MEV_BB proviene del área de la Recuperación de Información [122]. En la recuperación de información, el contenido de los productos es una descripción textual de los mismos, por lo que las características de los productos son las apariciones de un término concreto.

Para aplicar MEV_BB a productos con características multivaluadas, es necesario realizar una transformación de los datos. Para ilustrar el funcionamiento de esta transformación, se utilizará una base de datos de contenido (ver Tabla 3.2). Esta base de datos describe las características de cada producto (en este caso, canciones), como son la categoría, el idioma y el año; y valores para las mismas: para la categoría Jazz o Country; para el idioma Inglés, Español o Francés; y para el año, valores entre 1960 y 1962.

	Categoría	Idioma	Año
a_1	Jazz	Inglés	1962
a_2	Country	Español	1960
a_3	Country	Francés	1962
a_4	Jazz	Inglés	1962
a_5	Jazz	Inglés	1960
a_6	Country	Español	1962

Tabla 3.2: Contenido de los productos

Sobre los datos iniciales (ver Tabla 3.2) se realiza una transformación que convierte las características originales, cuyo tipo puede ser numérico, nominal, etc., a características booleanas. Esto se hace utilizando como nueva característica el par (característica,valor). Por ejemplo, si tenemos una canción cuya categoría es *Jazz*, la nueva característica será *Categoría.Jazz* = 1. El resto de características del ejemplo de la Tabla 3.2 se transformaría de la misma manera, cuyo resultado se expone en la Tabla 3.3.

Una vez transformada la información de contenido, se pasa a calcular los perfiles de usuario. Los perfiles de usuario se calculan agregando los perfiles de los productos que han

	Categoría		Idioma			Año	
	Jazz	Country	Inglés	Español	Francés	1960	1962
a_1	1	0	1	0	0	0	1
a_2	0	1	0	1	0	1	0
a_3	0	1	0	0	1	0	1
a_4	1	0	1	0	0	0	1
a_5	1	0	1	0	0	1	0
a_6	0	1	0	1	0	0	1

Tabla 3.3: Transformación del contenido de los productos a características booleanas

	a_1	a_2	a_3	a_4	a_5	a_6
u_1	-	4	-	-	5	-
u_2	-	3	-	4	-	-
u_3	-	-	-	-	-	4
u_4	5	-	3	-	-	-

Tabla 3.4: Valoraciones de preferencia

valorado (ver Tabla 3.4). En MEV_BB, los valores de $P_u = \{p_{u,1}, \dots, p_{u,j}, \dots, p_{u,m}\}$ coordenadas del perfil del usuario son la suma de las coordenadas de los perfiles de los productos que ha valorado (ver Fórmula 3.1):

$$P_j^u = \sum_{i=1}^{nu} v_j^{x_{u,i}} \quad (3.1)$$

donde P_j^u es el valor de la característica c_j en el perfil del usuario u , nu es el número de productos valorados por el usuario u y $v_j^{x_{u,i}}$ denota el valor de la característica c_j para el producto $a_{x_{u,i}}$.

Por ejemplo, el usuario U_1 ha valorado los productos a_2 e a_5 cuyos perfiles son $P(a_2) =$

P_{u_i}	Categoría		Idioma			Año	
	Jazz	Country	Inglés	Español	Francés	1960	1962
P_{u_1}	1	1	1	1	0	2	0
P_{u_2}	1	1	1	1	0	1	1
P_{u_3}	0	1	0	1	0	0	1
P_{u_4}	1	1	1	0	1	0	2
P_{u_5}	0	3	0	2	1	1	2

Tabla 3.5: Perfiles de usuarios según el Modelo Booleano Básico

$(0, 1, 0, 1, 0, 1, 0)$ y $P(a_5) = (1, 0, 1, 0, 0, 1, 0)$ respectivamente. Agregando los perfiles de los productos, su perfil de usuario es $P(U_1) = (1, 1, 1, 1, 0, 2, 0)$.

Los perfiles de usuario resultantes se muestran en la Tabla 3.5 (se han puesto las características por filas por comodidad en la representación) y corresponden al modelo en este método de recomendación basada en contenido. Para realizar la recomendación de productos, se calcula una medida de similitud entre el perfil del usuario activo y los perfiles de los productos que no ha valorado aún. Los n productos cuya similitud con el perfil del usuario sea más alta, serán los productos que este método recomienda al usuario.

$$\cos(u, v) = \frac{\sum_{i=1}^N u_i v_i}{\sqrt{\sum_{i=1}^N (u_i)^2} \cdot \sqrt{\sum_{i=1}^N (v_i)^2}} \quad (3.2)$$

Por ejemplo, si se desea recomendar productos al usuario u_3 , se calcula la similitud de su perfil con los perfiles de los productos a_1, a_2, a_3, a_4 y a_5 (los productos que no ha valorado), utilizando la medida del coseno (ver Fórmula 3.2). Las similitudes son $\{0.33, 0.66, 0.66, 0.33, 0\}$ respectivamente, por lo que la lista de recomendaciones es $\{a_2, a_3, a_1, a_4, a_5\}$.

3.2. Modelo Booleano con Ponderación Basada en TF-IDF

Una peculiaridad de los usuarios, que MEV_BB no tiene en cuenta, es que pueden dar mayor importancia a unas características de los productos frente a otras. Por ejemplo, un comprador que esté comparando dos coches puede considerar que el consumo del motor es más importante que el color de la carrocería.

La ponderación de características es un problema que ha sido tratado en otras áreas, con la Recuperación de Información [4, 104] y el Aprendizaje Automático [57, 65].

Para reflejar esto en los SRBCont con modelado en espacios vectoriales, P. Symeonidis [122] añade una ponderación asociada a cada una de las características de los productos. Propone utilizar la ponderación basada en TF-IDF (*Term Frequency - Inverse Document Frequency*) [4, 104], cuya ponderación penaliza los términos que son poco distintivos en un documento y da más peso a aquellos términos que distinguen un documento o pequeño número de documentos.

	c_1	\dots	c_m	R_u
$a_{x_{u1}}$	$v_1^{x_{u1}}$	\dots	$v_m^{x_{u1}}$	r_1^u
\dots	\dots	\dots	\dots	\dots
$a_{x_{u,nu}}$	$v_1^{x_{u,nu}}$	\dots	$v_m^{x_{u,nu}}$	r_{nu}^u
P_u	p_1^u	\dots	p_m^u	
W_u	w_1^u	\dots	w_m^u	

Tabla 3.6: Información que utiliza un Sistema de Recomendación Basado en Contenido con Ponderación de Características

El SRBCont con Modelado en Espacios Vectoriales y ponderación Basada en TF-IDF (MEV-TF-IDF), calcula un vector de ponderaciones (W_u), que se utiliza para ponderar la

similitud de los vectores de perfil de usuario y de perfil de producto. Los valores del perfil de usuario (P_u) y del perfil de producto se calculan de la misma manera que en MEV_BB (ver Apartado 3.1).

Dicho vector de ponderaciones W_u se calcula combinando el valor de similitud intra-usuario ($FF(u, c_j)$, Feature Frequency) y el valor de disimilitud inter-usuario ($IUF(c_j)$, Inverse User Frequency).

$$W(u, c_j) = FF(u, c_j) * IUF(c_j) \quad (3.3)$$

donde $FF(u, c_j)$ es el valor de la similitud intra-usuario de la característica c_j para el usuario u y $IUF(c_j)$ es la disimilitud inter-usuario de la característica c_j .

En MEV_TF-IDF, la similitud intra-usuario coincide con los valores del perfil de usuario en MEV_BB ($F(u, c_j) = P_j^u$, ver Fórmula 3.1). Con la similitud intra-usuario se pretende dar mayor peso a aquellas características que sean frecuentes en el perfil de usuario, es decir, a las características que posean los productos que el usuario u ha valorado positivamente.

La disimilitud inter-usuario penaliza aquellas características que no son distintivas. Para ello realiza una penalización de las características que son valoradas por todos los usuarios de la siguiente manera:

$$IUF(c_j) = \log \frac{|U|}{UF(c_j)} \quad (3.4)$$

donde $|U|$ es el número de usuarios registrados en el sistema y $UF(c_j)$ es el número de usuarios que valoraron positivamente algún producto con la característica c_j .

De esta manera, si una característica aparece en productos valorados por la mayoría de los usuarios del sistema, tendrá un valor de $IUF(c_j)$ bajo, ya que su valor será próximo a cero. Esto resulta en una penalización de la característica.

Con los datos del ejemplo, la disimilitud intra-usuarios de las características se muestra

en la Tabla 3.7.

	Categoría		Idioma			Año	
	Jazz	Country	Inglés	Español	Francés	1960	1962
$IUF(c_j)$	0,22	0	0,22	0,09	0,39	0,22	0,09

Tabla 3.7: Disimilitud inter-usuario según el SRBCont con ponderación TF-IDF

Una propiedad que deben cumplir las ponderaciones de características es que la suma sea 1, por lo que una vez calculado el vector de ponderaciones de características hay que normalizarlo:

$$w_i = \frac{w_i}{\sum_{j=1}^N w_j} \quad (3.5)$$

Con los datos del ejemplo, la ponderación de cada usuario para las características se muestra en la Tabla 3.8.

W_{u_i}	Categoría		Idioma			Año	
	Jazz	Country	Inglés	Español	Francés	1960	1962
W_{u_1}	0,22	0	0,22	0,09	0	0,45	0
W_{u_2}	0,25	0	0,25	0,11	0	0,25	0,11
W_{u_3}	0	0	0	0,5	0	0	0,5
W_{u_4}	0,21	0	0,21	0	0,38	0	0,18
W_{u_5}	0	0	0	0,19	0,39	0,22	0,19

Tabla 3.8: Vectores de ponderación de características según el SRBCont con ponderación TF-IDF

En este modelo el perfil de usuario u se compone de dos vectores: el vector de características P_u y la ponderación de las mismas W_u .

En la fase de recomendación, se calcula la similitud de P_u ponderada con el vector W_u con los perfiles de los productos que aún no ha valorado el usuario u . Finalmente se recomiendan los N productos con mayor similitud con el usuario.

$$\cos(u, v, w) = \frac{\sum_{i=1}^N u_i v_i w_i}{\sqrt{\sum_{i=1}^N (u_i)^2} \cdot \sqrt{\sum_{i=1}^N (v_i)^2}} \quad (3.6)$$

Por ejemplo, el usuario u_4 no ha valorado los productos a_2 , a_4 y a_5 ; por lo que se calcula el coseno ponderado con cada uno. En este caso, el producto que se recomienda es a_4 .

	a_2	a_4	a_5
u_4	0,204	0,816	0,408

Modelo Ponderado con Medidas de Contingencia y Entropía

En el capítulo anterior se presentaron las técnicas básicas de SRBCont con Modelado en Espacios Vectoriales, el modelado Booleano Básico (MEV_BB) y el modelo con ponderación de características basada en TF-IDF (MEV_TF-IDF).

Dichas técnicas presentan problemas en el tratamiento de la información de contenido. En este capítulo se realiza un análisis de estas técnicas y de los problemas que presentan, proponiendo un método de recomendación basado en contenido que los solucione. De esta manera se pretende mejorar la eficacia de los SRBCont presentados en el Capítulo 3.

Este capítulo se estructura de la siguiente manera. En primer lugar, se analizan los problemas de las técnicas básicas de recomendación basada en contenido con Modelado en Espacios Vectoriales. Tras esto, se analizan las posibles soluciones a estos problemas, presentando las herramientas necesarias para el cálculo de la ponderación de características. Finalmente se presenta el modelo, exponiendo un ejemplo que clarifique su funcionamiento.

4.1. Problemas del modelado Booleano

Las técnicas básicas de recomendación basada en contenido con Modelado en Espacios Vectoriales funcionan correctamente cuando la información de contenido viene determinada por términos extraídos de una descripción textual de los productos. Pero, a la hora de utilizar información de contenido definida por una serie de características que pueden tomar distintos valores, se observan ciertos problemas.

Por ejemplo, en el dominio de las películas se tienen características numéricas (año, duración) y características nominales (actores, directores, género). Los SRBCont previos convierten dichas características a características booleanas, por lo que se pierde información.

Los SRBCont con Modelado en Espacios Vectoriales básicos no aprovechan la información de ponderación de características que el usuario proporciona implícitamente cuando valora los productos. En MEV_TF-IDF se utiliza como medida de similitud intra-usuario el número de productos valorados positivamente que tienen esa característica. Una forma más avanzada de extraer la ponderación es calcular si las valoraciones están relacionadas con el valor de las características, es decir, si conociendo el valor de la característica se reduce la incertidumbre sobre la valoración.

Otro problema que presentan viene determinado por la conversión a características booleanas del contenido de los productos. Cuando se tiene una característica definida en un dominio real (por ejemplo, la duración de las películas), la conversión a características booleanas convierte esa característica en múltiples características. Por lo tanto, si en el dominio original tenemos la duración de las películas y en el conjunto de productos existen 100 duraciones distintas, el preprocesamiento convertiría una dimensión en 100.

El problema de la alta dimensionalidad también se presenta en características nominales. Por ejemplo, el conjunto de datos de MovieLens se compone de 9773 películas, que tienen definidas distintas características. Si nos centramos en el director de la película, existen

aproximadamente 4000 directores distintos, por lo que la dimensionalidad de los perfiles de usuarios y productos se dispara. Esto repercute en la eficiencia del cálculo del modelo en MEV_TF-IDF, ya que para cada característica se necesita calcular la frecuencia inversa (ver Fórmula 3.4) de cada una.

Otro problema asociado a la transformación en características booleanas, es que se tiene un valor de ponderación para cada valor distinto de las características.

Además de la alta dimensionalidad de los perfiles, la transformación a características booleanas elimina cierta información aportada por las características numéricas. Por ejemplo, tenemos una película cuyo año de lanzamiento es 2011 y el usuario activo sólo ha valorado películas de los años 2010 y 2012. El modelado booleano devuelve un valor de similitud incorrecto, ya que si calculamos la similitud del usuario con la película, diría que son completamente distintos, debido a que no existe coincidencia exacta en las características.

A continuación se propone un modelo de recomendación basada en contenido con medidas de entropía y contingencia para incorporar dicha información en el proceso y mejorar los resultados de recomendación.

4.2. Modelo Ponderado con medidas de Contingencia y Entropía

Con este modelo se pretende aplicar un método de ponderación de características en SRB-Cont, basado en [11], que aproveche la información que aportan las características nominales y numéricas. Por ello, en este modelo no se realiza un preprocesamiento de la información de contenido para transformar las características de los productos a valores booleanos.

Para el cálculo de la ponderación se utiliza el esquema descrito por P. Symeonidis [122], que utiliza dos tipos de medidas sobre las características: (i) la medida de disimilitud inter-usuario y (ii) la medida de similitud intra-usuario (ver Figura 4.1).

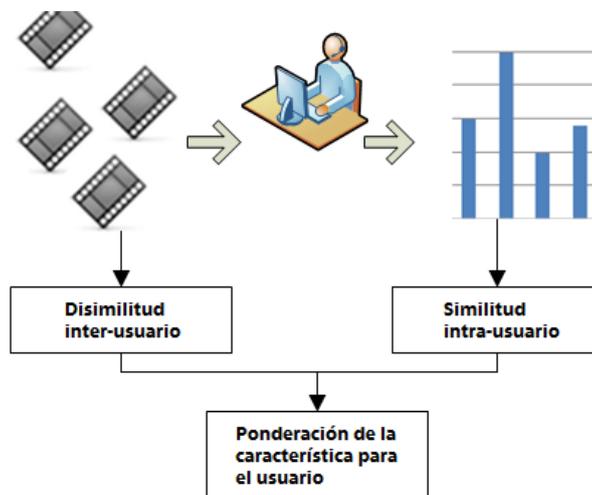


Figura 4.1: Ponderación de características basada en la entropía y medidas de contingencia

- **Disimilitud inter-usuario:** Esta medida refleja qué características son más informativas. Una característica informativa es aquella que, conociendo su valor, aporta una mayor cantidad de información sobre el producto que la posee.

- **Similitud intra-usuario:** Con esta medida se pretende dar mayor importancia a aquellas características que el usuario valora positivamente. Para ello, se aplica una medida que indica si el valor de una característica está relacionado con las valoraciones del usuario.

El método propuesto utiliza la misma información que MEV_TF-IDF (ver Tabla 3.6).

En los siguientes apartados se presentan las herramientas disponibles para el cálculo de la ponderación de las características, el modelo propuesto y un ejemplo de funcionamiento del mismo.

4.2.1. Herramientas para el cálculo de la ponderación

4.2.1.1. Medida de disimilitud inter-usuario

Nuestra propuesta se basa en la idea de que características más diversas son más relevantes para los usuarios, ya que son características que permiten discriminar los gustos de un usuario de una forma más adecuada. La medida de disimilitud inter-usuario trata de reflejar esta idea en la construcción de los perfiles de usuario para la recomendación basada en contenido.

Un tipo de medidas que se pueden aplicar para ello son las medidas de diversidad. A continuación se presentan distintas medidas que podemos aplicar a las características de los productos, tomando cada característica como una variable aleatoria.

Entropía de Shannon. La entropía o cantidad de información [32, 114] mide el número mínimo de bits necesarios para representar la información que una fuente de información transmite. Aplicada al cálculo de la cantidad de información que una variable aleatoria posee, la entropía es la cantidad media de información, medida en bits, que contiene una variable aleatoria. La entropía se define de la siguiente manera:

$$H(x) = - \sum_i p(x_i) \log_2(p(x_i)) \quad (4.1)$$

donde x es la variable aleatoria que viene dada por una función de distribución de probabilidad y $p(x_i)$ es la probabilidad de que la variable aleatoria tome el valor i . La entropía toma su valor máximo cuando $p(x_i) = \frac{1}{N}$ (N número de muestras) y es igual a $\log_2(N)$. Esta medida cambia mucho su valor cuando se añaden nuevos valores de la variable x , por lo que da mayor importancia a características distintivas.

Índice de Simpson. Fue propuesto por el británico Edward H. Simpson en 1949. Esta medida procede del área de la ecología e indica el índice de diversidad de las especies en un ecosistema. El índice de Simpson, aplicado a las variables aleatorias, mide la probabilidad de que dos muestras al azar tengan el mismo valor. El índice de Simpson se formula de la

siguiente manera:

$$D' = \frac{1}{\sum \frac{n_v}{N}} \quad (4.2)$$

donde n_v denota el número de muestras para los que la variable X toma el valor v . Para utilizar el índice de Simpson como medidor de la diversidad, se puede utilizar el índice de diversidad de Simpson $(1 - D')$ o el índice recíproco de Simpson $1/D'$. Una característica del índice de Simpson es que da poco peso a los valores poco frecuentes, es decir, a aquellos valores de características que aparezcan en un número reducido de productos. El índice de Simpson está influido en mayor medida por la uniformidad, es decir, un cambio en las distribuciones de mayor magnitud afecta en mayor medida a su valor.

4.2.1.2. Medidas de similitud intra-usuario

La medida de similitud intra-usuario intenta revelar si existe relación entre las valoraciones de preferencia de un usuario y los valores de una característica. Para ello se pueden aplicar medidas que revelen la existencia o no de dicha relación y su magnitud.

Este modelo supone que, si existe una relación entre las valoraciones de preferencia y los valores de una característica, entonces el usuario está teniendo en cuenta ésta para dar su valoración, es decir, está dando importancia a esta característica.

La información de contenido que un SR maneja, define una serie de características referentes a los productos. Por ejemplo, una película tiene como características el género, año de lanzamiento, la duración, el actor principal, los directores, etc. Para una utilización correcta de esta información, la naturaleza de las característica debe ser tomada en cuenta en el cálculo de la similitud intra-usuario. Por ello, se propone la aplicación de medidas distintas en función del tipo de característica.

A continuación se presentan (i) las medidas aplicables a características nominales y (ii) las medidas aplicables a características numéricas, en el cálculo de la similitud intra-usuario.

Características nominales

Una característica de un producto puede ser de tipo nominal (o categórica). Las características nominales son variables que pueden tomar un valor concreto dentro de un conjunto finito de valores previamente delimitado. Una peculiaridad de dichos valores es que no admiten un criterio de orden.

Un ejemplo de característica nominal es el género de una película. Esta pertenece a un género, que puede ser *comedia*, *drama*, *terror*, etc. Pero entre dichos géneros no puede establecerse una relación de orden intuitiva. La única relación que se puede utilizar para comparar los géneros de una película es la igualdad/desigualdad.

Para tener una medida de la relación entre dos variables (al menos una nominal) se utilizan los coeficientes de contingencia. Un coeficiente de contingencia revela si existe relación entre dos variables, es decir, si conociendo el valor de una de ellas se reduce la incertidumbre sobre el valor de la otra. Los coeficientes de contingencia se aplican a la tabla de contingencia, que es una representación de distribuciones de los individuos en relación al valor de las variables a comparar:

	B_1	B_2	\dots	B_s	Total
A_1	x_{11}	x_{12}	\dots	x_{1s}	$n_{1.}$
A_2	x_{21}	x_{22}	\dots	x_{2s}	$n_{2.}$
\dots	\dots	\dots	\dots	\dots	\dots
A_r	x_{r1}	x_{r2}	\dots	x_{rs}	$n_{r.}$
Total	$n_{.1}$	$n_{.2}$	\dots	$n_{.s}$	n

Tabla 4.1: Ejemplo de tabla de contingencia

donde:

- A es una variable que puede tomar los valores A_1, A_2, \dots, A_r

- B es una variable que puede tomar los valores B_1, B_2, \dots, B_s .

Una tabla de contingencia se dice que es de dimensión $r \times s$. Sobre los datos de esta, se calcula el estadístico χ^2 (ver Fórmula 4.3) que sirve de soporte para el cálculo de distintos coeficientes de contingencia.

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^s \frac{(x_{ij} - q_{ij})^2}{q_{ij}} \quad \text{donde} \quad q_{ij} = \frac{n_{i.} * n_{.j}}{n} \quad (4.3)$$

Para el estudio de la relación entre una variable nominal y las valoraciones de preferencia de los usuarios se propone el uso de un coeficiente de contingencia. A continuación se presentan distintos coeficiente de contingencia que se considerarán para el cálculo de la medida de similitud intra-usuario para las características nominales.

Coefficiente de contingencia de Karl Pearson. Esta medida de asociación no se ve afectada por el tamaño de la muestra y está acotada ($0 < C < 1$). Un problema de esta medida es que, aunque la relación entre las variables sea perfecta, no siempre alcanza el valor 1.

$$C = \sqrt{\frac{\chi^2}{\chi^2 + n}} \quad (4.4)$$

Coefficiente V de Cramer. [15]. Es uno de los ratios de contingencia más usados para medir la dependencia entre dos variables aleatorias, X e Y, donde al menos una de las dos es nominal. La fórmula para obtener dicho coeficiente sobre nuestros datos es la siguiente:

$$V' = \sqrt{\frac{\chi^2}{nk}} \quad \text{donde} \quad k = \text{mín}(r, s) - 1 \quad (4.5)$$

Características numéricas

Las características de los productos también pueden ser de tipo numérico. Las características numéricas son variables que pueden tomar un valor finito o no finito dentro de un rango de valores. Una peculiaridad de las variables numéricas es que sus valores tienen un orden natural.

Un ejemplo de característica numérica es el año de lanzamiento de una película, que puede tomar un valor cualquiera entre la invención del cinematógrafo hasta la actualidad.

Debido a que las valoraciones de preferencia también poseen una relación de orden preestablecida, se pueden calcular medidas que determinen si una característica numérica de los productos y las valoraciones de preferencia de un usuario están relacionadas y en qué medida.

Estas medidas se denominan coeficientes de correlación. A continuación se presentan algunos coeficientes de correlación sobre variables numéricas.

Coefficiente de correlación de Pearson. El coeficiente de correlación de Pearson [15] es un índice estadístico que mide la relación lineal entre dos variables dando una medida independiente de la escala empleada. Se obtiene dividiendo la covarianza por el producto de las desviaciones estándar de ambas variables:

$$PCC(X, Y) = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \quad (4.6)$$

$$s.t. \begin{cases} \sigma_{XY} = \frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x} \bar{y} \\ \sigma_X = \sqrt{\sum_{i=1}^n \frac{x_i^2}{n} - \bar{x}^2} \\ \sigma_Y = \sqrt{\sum_{i=1}^n \frac{y_i^2}{n} - \bar{y}^2} \end{cases}$$

El coeficiente de correlación de Pearson, además de revelar relaciones lineales entre los valores de dos variables aleatorias, permite también conocer la dirección de la relación, es decir, si es una relación directa o inversa. Como inconveniente, esta medida no es útil para revelar relaciones no lineales.

Coefficiente de correlación de Spearman El Coeficiente de correlación de Spearman [69] se basa en el orden de los valores de las variables. Al igual que el coeficiente de correlación de Pearson, es independiente de la magnitud de los valores. Para calcular este coeficiente, los valores que pueden tomar las variables aleatorias X e Y se ordenan y posteriormente se sustituyen los valores por su orden. En la Tabla 4.2 se muestra un ejemplo de este cálculo en el que, en caso de empate, se asigna el valor medio de los ordenes correspondientes a los valores empatados.

A_i	B_i	$orden(A_i)$	$orden(B_i)$
1	0	1	1
2	1	2	2
3	3	3	3
4	4	4	4
5	6	5.5	7
5	6	5.5	7
8	6	7	7
9	7	8	8
10	9	9	9

Tabla 4.2: Cálculo del orden de los datos para el coeficiente de contingencia de Spearman

Una vez se han cambiado los valores a sus correspondientes órdenes, se calcula el coeficiente mediante la siguiente ecuación:

$$\rho = 1 - \frac{6 \sum D_i^2}{n(n^2 - 1)} \quad \text{donde} \quad D_i = orden(A_i) - orden(B_i) \quad (4.7)$$

Este coeficiente de correlación permite revelar relaciones no lineales entre las dos variables. Como inconveniente, este coeficiente no revela relaciones inversas entre las variables. Otra característica de esta medida es que si una variable es monótona y creciente respecto de los valores de la otra, devuelve un valor de 1.

4.2.2. Configuración del sistema

Una vez revisadas las diferentes medidas que se pueden aplicar en cada parte, se pasa a realizar la elección de estos parámetros.

- Medida de disimilitud inter-usuario** La medida de entropía es más adecuada para este cálculo, ya que tiene en cuenta que existan mayor cantidad de valores distintos.

El índice de Simpson da mayor importancia a la uniformidad de los valores, es decir, que todos los valores tengan la misma frecuencia relativa.

- **Medida de similitud intra-usuario** Como se expone en el Apartado 4.2.1.2, la medida de similitud intra-usuario que se aplique dependerá de la naturaleza de la característica para la que se calcula su ponderación:
 - **Características nominales** Para las características de tipo nominal se aplican los coeficientes de contingencia. El coeficiente de contingencia que se elige es el coeficiente V de Cramer. Se ha tomado esta decisión porque el coeficiente de contingencia de Pearson no alcanza el valor 1, aunque la contingencia sea perfecta.
 - **Características numéricas** Para las características de tipo numérico se aplican los coeficientes de correlación. El coeficiente de correlación elegido es el coeficiente de correlación de Pearson. Se ha tomado esta decisión porque el coeficiente de Spearman, aunque revela relaciones no lineales entre variables, no permite revelar relaciones inversas. Consideramos que esta es una característica importante, ya que los usuarios pueden valorar positivamente que una característica tenga un valor bajo. Por ejemplo un usuario puede considerar bueno que el peso de un teléfono móvil sea bajo, que un producto sea barato, etc. Por esto, como medida de similitud intra-usuario se utilizará el valor absoluto del coeficiente de correlación de Pearson.

De esta manera, el esquema general de funcionamiento de la propuesta queda descrito en la Figura 4.2.

4.2.3. Descripción detallada del algoritmo

En primer lugar, el algoritmo de recomendación calcula los perfiles de usuario, que reflejan los gustos y necesidades de los usuarios. Para ello, se siguen los siguientes pasos:

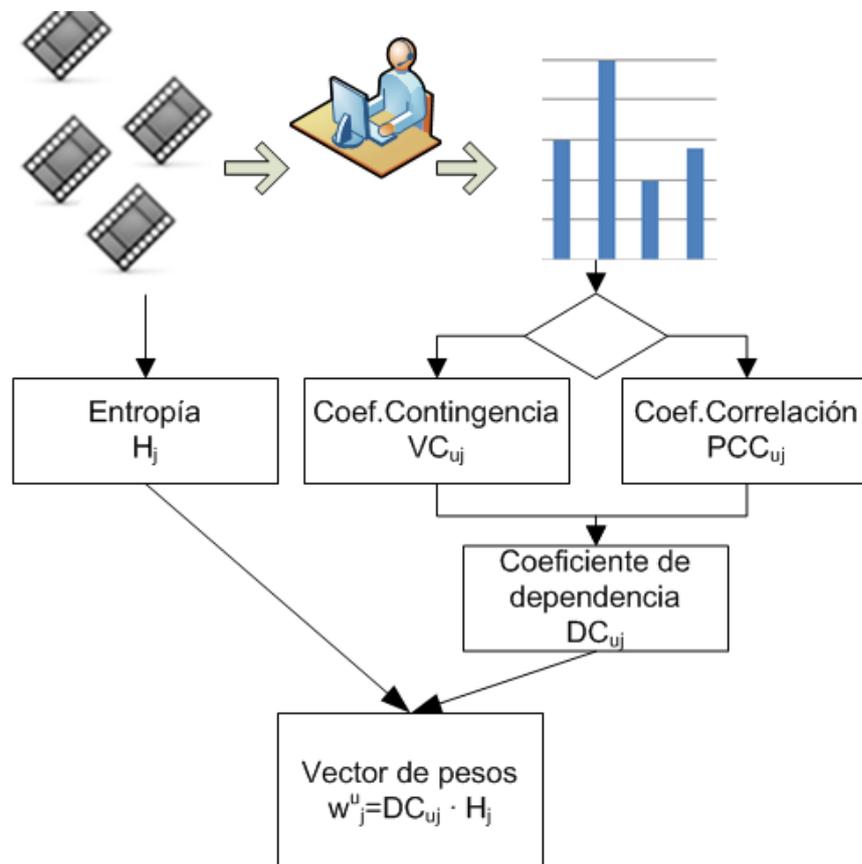


Figura 4.2: Ponderación de características basada en la entropía y medidas de contingencia

En la Figura 4.3 se muestra el algoritmo que implementa el diseño descrito en el apartado anterior. A continuación se explica detalladamente su funcionamiento, utilizando un ejemplo para ello (información de contenido: Tabla 4.3, información del usuario: Tabla 4.4):

Primero, se realiza el cálculo de la entropía de cada característica de los productos (Pasos 1 a 3, ver Fórmula 4.1). Esto se hace en primer lugar, ya que es independiente de los usuarios y de sus valoraciones. Sólo tiene en cuenta los valores distintos de las características y el número de productos que tiene cada valor, es decir, la información de contenido. Utilizando los datos de la Tabla 4.3, las entropías son las siguientes:

$$H_j = - \sum_{k_j} (f_{k_j}/n) \text{Log}_2(f_{k_j}/n) \quad (4.8)$$

siendo $\{k_j\}$ el conjunto de valores que la característica c_j puede tomar, f_{k_j} la frecuencia del

```

1. Para cada c en todasCaracteristicasProductos() hacer:
2.     Calcular y almacenar entropia(c);
3. Fin_para;
4. Normalizar_entropias();
5. Para cada u en todosUsuarios() hacer:
6.     perfil = InicializarPerfil();
7.     Para cada p en u.productosValorados() hacer:
8.         Si ( u.getValoracion(p) > umbral )
9.             perfilUsuario.añadeProducto(p);
10.        Fin_si;
11.    Fin_para;
12.    Para cada c en perfil.caracteristicasValoradas() hacer:
13.        Si c es de tipo nominal hacer:
14.            intraU = VCramer(u.paresValorCaracterística_Valoración(c));
15.        Fin_si;
16.        Si c es de tipo numérico hacer:
17.            intraU = abs(Pearson(u.paresValorCaracterística_Valoración(c)));
18.        Fin_si;
19.        perfil.establecerPeso(c,intraU*entropia(c));
20.    Fin_para;
21.    perfil.normalizar_pesos();
22. Fin_para;

```

Figura 4.3: Algoritmo de la propuesta de recomendación basada en contenido

valor k_j en todo el conjunto de productos A y n el cardinal de A . Este cálculo considera $\log 0 = 0$, de modo que los valores cuya frecuencia sea 0 no afecten el resultado.

$$H = \{H_j\} = \{1,86, 2,13, 3,92, 4,12, 1,86\}$$

Posteriormente, se realiza la normalización de las entropías (Paso 4). Para su utilización como pesos, las entropías deben cumplir la propiedad $\sum w_j = 1$, aplicando la Fórmula 3.5. Las entropías de las características quedan como sigue:

	c_1	c_2	c_3	c_4	c_5
a_1	1	2	2	5	A
a_2	3	6	5	20	C
a_3	2	2	12	10	B
a_4	2	3	14	11	B
a_5	4	4	8	25	D
a_6	2	4	16	13	B
a_7	1	2	4	5	A
a_8	1	6	4	2	A
a_9	4	5	15	28	D
a_{10}	2	4	14	12	B
a_{11}	2	3	18	9	B
a_{12}	1	6	17	4	A
a_{13}	1	2	10	6	A
a_{14}	3	4	13	18	C
a_{15}	4	5	17	27	D
a_{16}	1	2	1	3	A
a_{17}	2	2	1	10	B
a_{18}	4	4	19	29	D
a_{19}	1	2	9	1	A
a_{20}	2	4	11	14	B

Tabla 4.3: Conjunto de datos de contenido de los productos

$$H^* = \{H_j^*\} = \{0,13, 0,15, 0,28, 0,30, 0,13\}$$

Una vez calculadas las medidas de disimilitud inter-usuario, se pasa a la generación de los perfiles de usuario (Pasos 5-11). Para ello, se inicializa el perfil de usuario (Paso 6). Esto se hace generando un vector en el que, para cada valor de cada característica nominal, se inicializa un valor a cero. Para las características numéricas se tiene un único valor, que

almacena el valor medio de la característica para el usuario. Una vez hecho esto, se recorre el conjunto de productos que el usuario ha valorado (Pasos 7 a 11). Si la valoración que ha dado es positiva (ver Tabla 4.4), el producto se añade al perfil del usuario (Paso 9). En el ejemplo, se consideran positivas las valoraciones mayores o iguales que 4, por lo que se agregan los perfiles de los productos $\{a_2, a_5, a_9\}$. Esto se hace realizando la agregación de los perfiles de los productos como se expone en la Tabla 4.5.

Para las características de tipo nominal (en el ejemplo, c_5), se almacena un valor de perfil de usuario por cada valor distinto de la misma (en el ejemplo: A,B,C y D). En el cálculo de las recomendaciones, se utilizará como valor de la característica c_5 para el perfil del usuario, el valor correspondiente a la característica c_5 del producto con el que se compare.

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
1	5	2	3	4	3	1	1	5	2

Tabla 4.4: Valoraciones del usuario utilizado como ejemplo

	c_1	c_2	c_3	c_4	c_5			
					A	B	C	D
P_{u_1}	-	-	-	-	0	0	0	0
$+a_2$	3	6	5	20	0	0	1	0
$+a_5$	$\frac{7}{2}$	$\frac{10}{2}$	$\frac{13}{2}$	$\frac{45}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$
$+a_9$	$\frac{11}{3}$	$\frac{15}{3}$	$\frac{28}{3}$	$\frac{73}{3}$	0	0	$\frac{1}{3}$	$\frac{2}{3}$

Tabla 4.5: Generación iterativa del perfil de usuario

Una vez que se han calculado los valores del perfil, se realiza el cálculo de las similitudes intra-usuario (Pasos 12 a 20). Para ello se recorre cada característica calculando el coeficiente de contingencia V de Cramer (Pasos 13 a 15, ver Fórmula 4.9) si es de tipo nominal o el coeficiente de correlación de Pearson (Pasos 16 a 18, ver Fórmula 4.10), si es de tipo numérico. La función $u.paresValorCaracterística_Valoración(c)$ construye la lista de pares

$\langle \text{valorCaracterística}, \text{valoración} \rangle$ que se utilizará para calcular cada uno de estos coeficientes. El coeficiente de correlación de Pearson está limitado en el intervalo $[-1,1]$ dando información del grado de dependencia y también del tipo de dependencia, directa o inversa. Para nuestros propósitos, lo importante no es el tipo de dependencia, sino la intensidad. Por lo tanto, tomamos el valor absoluto, con lo cual, el resultado estará siempre en el intervalo $[0,1]$.

$$VC_{uj} = \sqrt{\frac{\sum_{k_u} \sum_{k_j} \frac{\left(f_{k_u, k_j} - \frac{f_{k_u} f_{k_j}}{n_u}\right)^2}{\frac{f_{k_u} f_{k_j}}{n_u}}}{n_u \min(|D_u|, |D_j|)}} \quad (4.9)$$

donde k_u y k_j son índices en los conjuntos de datos R_u y V_j^u respectivamente, f_{k_u} , f_{k_j} son las frecuencias de los valores indexados por k_u y k_j respectivamente, y f_{k_u, k_j} es la frecuencia de la ocurrencia simultánea de los dos valores indexados por k_u y k_j .

$$PCC_{uj} = \frac{\sum_i r_i^u v_j^{x_{ui}} - \frac{\sum_i r_i^u \sum_i v_j^{x_{ui}}}{n_u}}{\sqrt{\left(\sum_i (r_i^u)^2 - \frac{(\sum_i r_i^u)^2}{n_u}\right)} \sqrt{\left(\sum_i (v_j^{x_{ui}})^2 - \frac{(\sum_i v_j^{x_{ui}})^2}{n_u}\right)}} \quad (4.10)$$

En el ejemplo se calcula el coeficiente de correlación de Pearson para las características c_1, c_2, c_3, c_4 y el coeficiente V de Cramer para la característica c_5 . Para el usuario u_1 , el valor de similitud intra-usuario de cada característica es de 0,91,0,52,0,40,0,93,0,73.

Una vez se ha calculado esto, se tienen las dos componentes de la ponderación de características final. Para obtener la ponderación, se multiplican la medida de disimilitud inter-usuario y de similitud intra-usuario (ver Fórmula 4.11), para obtener el peso de la característica c_j para el usuario u .

$$w_j^u = H_j^* \cdot DC_{uj} \quad (4.11)$$

$$DC_{uj} = \begin{cases} |PCC_{uj}| & \text{si } c_j \text{ es numérico} \\ VC_{uj} & \text{si } c_j \text{ es nominal} \end{cases}$$

siendo VC_{uj} el coeficiente de contingencia V de Cramer correspondiente a R_u y c_j (nominales); PCC_{uj} el coeficiente de correlación de Pearson correspondiente a las variables R_u y c_j (numéricas).

Una vez calculados todos los pesos de las características para el usuario u se deben normalizar (Paso 21, ver Fórmula 3.5). Por lo tanto, los pesos de las características para el usuario u son:

$$w_{u_1} = \{0,12, 0,08, 0,11, 0,28, 0,10\} \longrightarrow w_{u_1}^* = \{0,18, 0,12, 0,16, 0,40, 0,14\}$$

Cuando finaliza la construcción del modelo, para cada usuario se tendrán dos vectores: el vector de valores del perfil (p^u) y el vector de ponderaciones de las características (w^u). En el ejemplo, para el usuario u_1 , estos vectores son:

	c_1	c_2	c_3	c_4	c_5			
					A	B	C	D
p^{u_1}	3,66	5	9,33	24,33	0	0	0,33	0,66
w^{u_1}	0,18	0,12	0,16	0,40	0,14			

El perfil y la ponderación de características del usuario se utiliza para realizar las recomendaciones. Estas se calculan mediante el coseno del ángulo entre los vectores del perfil de usuario y del perfil de producto (ver Fórmula 4.12).

$$\cos(u, v, w) = \frac{\sum_{i=1}^N u_i v_i w_i}{\sqrt{\sum_{i=1}^N (u_i)^2} \cdot \sqrt{\sum_{i=1}^N (v_i)^2}} \quad (4.12)$$

Para evitar que las características numéricas tengan mas peso en la similitud que las características nominales, se debe modificar su dominio a $[0, 1]$. De esta manera, los perfiles de los productos a_{11} a a_{20} y del usuario u_1 quedan como sigue:

	c_1	c_2	c_3	c_4	c_5			
					A	B	C	D
$p^{a_{11}}$	0.33	0.25	0.94	0.28	-	1	-	-
$p^{a_{12}}$	0	1	0.88	0.10	1	-	-	-
$p^{a_{13}}$	0	0	0.5	0.17	1	-	-	-
$p^{a_{14}}$	0.66	0.5	0.66	0.60	-	-	1	-
$p^{a_{15}}$	1	0.75	0.88	0.92	-	-	-	1
$p^{a_{16}}$	0	0	0	0.07	1	-	-	-
$p^{a_{17}}$	0.33	0	0	0.32	-	1	-	-
$p^{a_{18}}$	1	0.5	1	1	-	-	-	1
$p^{a_{19}}$	0	0	0.44	0	1	-	-	-
$p^{a_{20}}$	0.33	0.5	0.55	0.46	-	1	-	-
p^{u_1}	0.88	0.75	0.46	0.83	0	0	0.33	0.66

Por ejemplo, el usuario u_1 no ha valorado los productos a_{11} , a_{12} , a_{13} , a_{14} , a_{15} , a_{16} , a_{17} , a_{18} , a_{19} y a_{20} ; por lo que se calcula el coeficiente del coseno ponderado con cada uno. Como se comentó anteriormente, cuando se realizan las recomendaciones, el perfil del usuario en la característica c_5 toma el valor correspondiente al valor del producto para la característica c_5 . El producto en la característica c_5 toma el valor 1.

	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}
cos	0,109	0,076	0,056	0,184	0,206	0,015	0,096	0,208	0,020	0,142

El producto más adecuado para el usuario u_1 será el que tenga mayor valor para el coseno del ángulo entre los vectores. Según este modelo, es el producto a_{18} .

Una vez explicado el funcionamiento del método propuesto, en el siguiente capítulo, se lleva a cabo un caso de estudio para determinar su validez.

Caso de Estudio

En este capítulo se describe el proceso realizado para validar el Modelo Ponderado con Medidas de Contingencia y Entropía. El objetivo planteado es comprobar si el modelo propuesto obtiene mejores resultados en términos de eficacia que los modelos booleano y TF-IDF. Para ello se realiza una evaluación *off-line*, comparando el modelo propuesto con los algoritmos básicos de SRBCont con Modelado en Espacios Vectoriales, presentados en el Capítulo 3.

El resto del capítulo se estructura de la siguiente manera. En primer lugar, se describe el proceso de experimentación que se utilizará en este caso de estudio, describiendo los parámetros del mismo. A continuación se realiza una descripción del conjunto de datos utilizado. Finalmente, se exponen y analizan los resultados de la experimentación.

5.1. Exposición del proceso de experimentación.

Este caso de estudio tiene como objetivo determinar si el método propuesto aporta una mejora en eficacia respecto de los algoritmos básicos de SRBCont con Modelado en Espacios Vectoriales. Por ello, se realiza una evaluación de los tres métodos que se comparan:

- SRBCont con modelado booleano (MEV_BB).
- SRBCont con ponderación TF-IDF (MEV_TF-IDF).

- Modelo Ponderado con Medidas de Contingencia y Entropía (MEV_MCE).

Para ello, existen dos opciones: evaluación *off-line* y evaluación *on-line*. En nuestro caso, realizamos una evaluación *off-line*, ya que el coste de su realización es menor y existen multitud de conjuntos de datos públicos, con información de los gustos o necesidades de los usuarios, que se pueden utilizar para este objetivo.

Los modelos que se pretende comparar tienen en común la generación de un modelo para describir los gustos y necesidades de los usuarios, mediante un perfil asociado a cada usuario (fase de entrenamiento). Este perfil generado puede ser visto como un clasificador de productos, ya que su objetivo es discriminar los productos que son relevantes para el usuario de los que no lo son. Por ello, se pueden aplicar técnicas de evaluación de algoritmos de Aprendizaje Automático, como los protocolos *Hold Out* o Validación Cruzada (ver Apartado 2.4.1).

Para la experimentación realizada en este caso de estudio, se elige el protocolo de evaluación Validación Cruzada. Se elige este método en lugar de *Hold Out* porque garantiza que todos los datos se predicen al menos una vez en cada ejecución. Esto es así porque realiza una partición metódica del conjunto de datos inicial, mientras que *Hold Out* realiza la división de los datos de forma aleatoria.

5.1.1. Descripción de los conjuntos de datos

Para realizar la experimentación de este caso de estudio se necesita utilizar un conjunto de datos. Estos algoritmos utilizan como conjunto de datos una relación de las valoraciones de preferencia que ciertos usuarios asignaron a distintos productos.

Existen distintos conjuntos de datos públicos para este propósito. En ellos, se recoge información de los usuarios (que ha sido procesada para garantizar el anonimato) y de su interacción con un sistema de valoración de productos.

Algunos conjuntos de datos públicos son los siguientes:

- **Jester**¹: Recoge valoraciones sobre chistes.
- **Book Crossing**²: Recoge valoraciones sobre libros.
- **MovieLens**³: Recoge valoraciones sobre películas.
- **Netflix**⁴: Recoge valoraciones sobre películas.

En este caso se utiliza el conjunto MovieLens. Este conjunto de datos fue recopilado durante el proyecto *GroupLens Research Project*, realizado en la Universidad de Minnesota.

Para utilizar este conjunto en recomendación basada en contenido, ha sido necesario completar la información de contenido. Para ello, se ha recopilado información sobre el director, género, año de lanzamiento y país de las películas, utilizando la base de datos de películas *IMDB*⁵.

Tras completar la información de contenido, se realiza un filtrado de los usuarios, que asegura que todos tienen un mínimo de 15 productos valorados. La caracterización del conjunto de datos resultante es la siguiente:

- 9.773 productos.
- 69.878 usuarios.
- 9.464.734 valoraciones. Por lo tanto, el conjunto tiene un 98,6 % de las valoraciones no definidas (*sparsity*).

¹<http://goldberg.berkeley.edu/jester-data/>

²<http://www.informatik.uni-freiburg.de/~chiegler/BX/>

³<http://www.grouplens.org/node/73#attachments>

⁴<http://www.netflixprize.com/>

⁵<http://www.imdb.es/>

- Las valoraciones de preferencia son *estrellas*, por lo que serán valores enteros entre 1 y 5. Una valoración de 1 significa que al usuario no le ha gustado y 5 que le ha parecido perfecta.
- El contenido que se tiene de los productos son las siguientes características:
 - **Género:** 25 valores distintos.
 - **Director:** 3.999 valores distintos.
 - **Año de lanzamiento:** Valores enteros entre 1915 y 2008, por lo que esta característica tiene 94 valores distintos.
 - **País:** 70 valores distintos.

5.1.2. Configuración del experimento

A continuación se describe la configuración elegida para la realización de la experimentación necesaria para alcanzar los objetivos descritos en este caso de estudio.

El **protocolo de evaluación** elegido para la realización del experimento es la **Validación Cruzada**. Este protocolo tiene como parámetro k , que es el número de particiones que se realizan. En este caso, se ha elegido **$k=5$** . Para garantizar que los resultados del experimento no estén sesgados, se realizan múltiples ejecuciones, en este caso se realizan **50 ejecuciones**.

Las **medidas de evaluación** se utilizan para dar una puntuación a cada algoritmo que refleje su comportamiento en las ejecuciones. En este caso, utilizamos la **Precisión** y el **Recall**. Estas medias de evaluación necesitan que se defina un **criterio de relevancia**, que se utiliza para decidir si producto es relevante o no. En este caso, consideramos que los productos con una valoración mayor o igual a 4 son relevantes y menor o igual a 3 son no relevantes.

La **implementación** de los algoritmos se ha realizado utilizando el lenguaje Java. Se

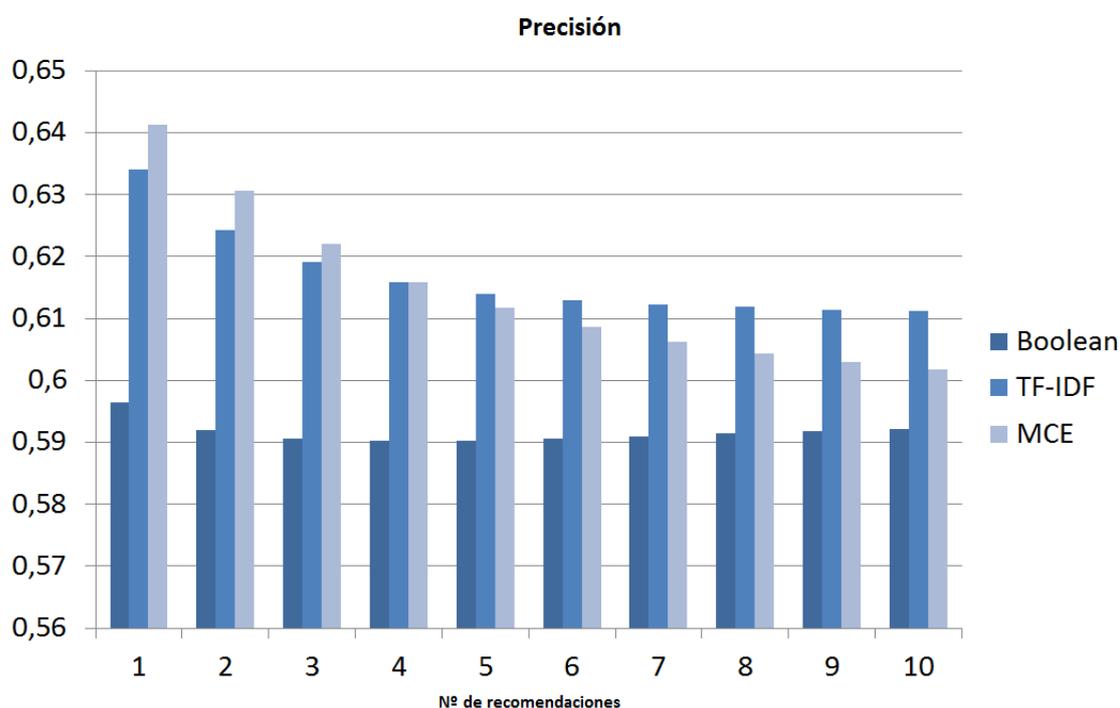


Figura 5.1: Precisión de los algoritmos en el experimento

ha utilizado este lenguaje por su capacidad de funcionar en distintas plataformas y por mi experiencia en desarrollo de aplicaciones en Java. Para su ejecución, se ha utilizado un ordenador con CPU Core i3-2600@3.40GHz y 8GB de memoria RAM.

5.2. Resultados

Se ha realizado un experimento con la configuración descrita en el Apartado 5.1.2 . En este apartado se exponen y analizan los resultados del mismo. Los resultados de precisión, recall y F1-medida se analizan en función del número de productos que se considera que cada algoritmo recomienda al usuario activo. Como el número de productos recomendados no afecta al tiempo de entrenamiento (construcción del modelo) y evaluación (recomendación), se muestra el tiempo medio de cada algoritmo.

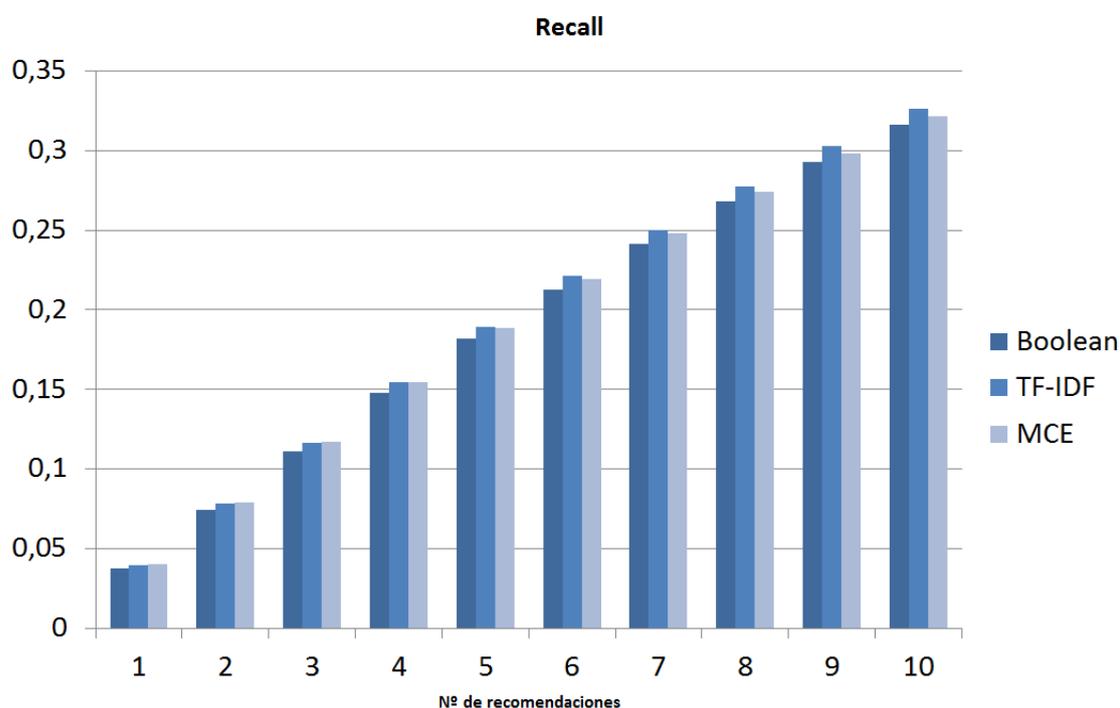


Figura 5.2: Recall de los algoritmos en el experimento

Como se muestra en las Figuras 5.1, 5.2 y 5.3, los SRBCont con ponderación de características, MEV_TF-IDF y MEV_MCE, mejoran en precisión, recall y F1-medida al MEV_BB, que no realiza ponderación de las características. Esto indica que la ponderación supone una mejora, lo que confirma la hipótesis de que los usuarios, al realizar las valoraciones de preferencia, tienen en cuenta las características de las películas en diferente medida.

Asimismo, el algoritmo MEV_MCE mejora a MEV_TF-IDF en precisión para 4 o menos recomendaciones, lo que indica su mayor capacidad de recomendar productos correctamente. Por lo tanto, el algoritmo MEV_MCE es útil para ser utilizado en aquellos dominios en los que el usuario reciba una lista de recomendaciones pequeña, por ejemplo, cuando el sistema ofrece recomendaciones al usuario sin que hayan sido solicitadas previamente.

En cuanto a los resultados de eficiencia en tiempo de construcción del modelo (ver Figura 5.4), el algoritmo MEV_MCE mejora a MEV_TF-IDF. Esto es porque el algoritmo MEV_TF-IDF tiene que calcular la frecuencia inversa utilizando los perfiles de usuario ya construidos,

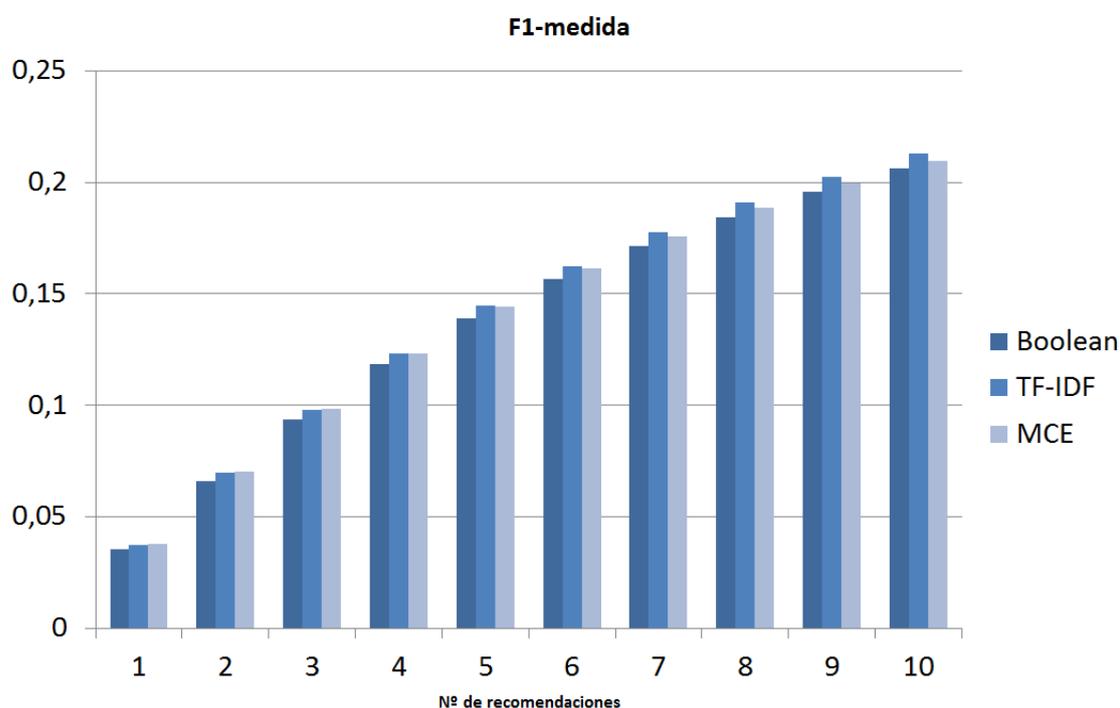


Figura 5.3: F1-medida de los algoritmos en el experimento

por lo que hay que añadir un bucle de ejecución adicional de orden de eficiencia $O(n \cdot m)$, donde n es el número de características, una vez convertidas a características booleanas; y m es el número de usuarios totales en el sistema. Los dos algoritmos tardan más tiempo que el algoritmo MEV_BB ya que este no calcula ponderación de características.

El tiempo que ejecución que más importa en un SR es el tiempo medio de recomendación por usuario, ya que será el tiempo de respuesta del sistema. Como se muestra en la Figura 5.5 el algoritmo MEV_MCE mejora a los algoritmos MEV_BB y MEV_TF-IDF. Esto es así porque éstos últimos producen perfiles de usuario y producto más complejos, por la conversión a características booleanas. Un vector de perfil (ya sea de usuario o de producto) en MEV_MCE tendrá 4 dimensiones (una por cada característica), mientras que en MEV_BB y MEV_TF-IDF tendrá 4.188 dimensiones (una por cada valor distinto de cada característica).

Esta evaluación demuestra que MEV_MCE presenta sensibles mejoras de eficacia sobre los modelos clásicos de recomendación basada en contenido con Modelado en Espacios

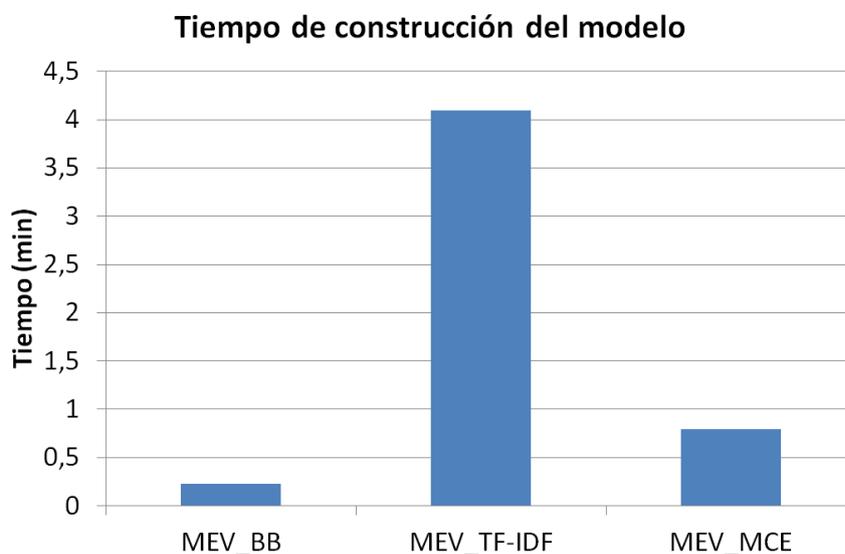


Figura 5.4: Tiempos de construcción del modelo de los métodos

Vectoriales. En cuanto a la eficacia, su mejora es más acentuada, ya que comparado con MEV_TF-IDF, proporciona recomendaciones de eficacia similar y mejora el tiempo de ejecución en construcción del modelo un % y en tiempo de recomendación medio un % .

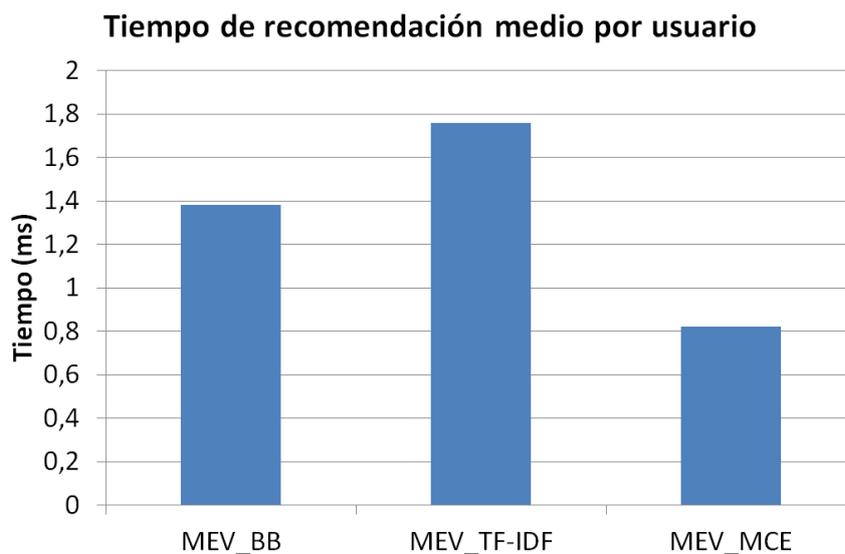


Figura 5.5: Tiempo de respuesta de los métodos

Conclusiones

En este capítulo se exponen las conclusiones extraídas de la realización de este trabajo y de los resultados obtenidos. También se comentan los trabajos futuros que se ha planteado abordar a partir del mismo.

Los Sistemas de Recomendación son una potente herramienta de personalización en dominios en los que la cantidad de productos disponibles desborda la capacidad de un usuario para evaluar cual de ellos le puede resultar útil o interesante. Estos sistemas ayudan al usuario, filtrando los productos existentes en el sistema y presentando una lista de productos acorde con sus gustos o necesidades.

Realizando una revisión bibliográfica del área de los Sistemas de Recomendación, se detectó que no existían Sistemas de Recomendación Basados en Contenido con Modelado en Espacios Vectoriales que realizaran un tratamiento correcto del contenido de los productos, cuando viene determinado por una serie de características numéricas o nominales que pueden tomar distintos valores. Los sistemas existentes están diseñados para funcionar correctamente cuando el contenido viene dado por una descripción textual del producto.

Analizando estos métodos, aplicados a contenido mediante características multivaluadas, se detectaron una serie de problemas debidos a la transformación de la información de contenido en características booleanas. Algunos de ellos son la alta dimensionalidad de los perfiles y la pérdida de información en características numéricas.

Para abordar estos problemas se ha propuesto un modelo que ofrece mejoras en eficacia y eficiencia con respecto a modelos clásicos, en situaciones en las que los productos están descritos por un conjunto de características, cada una de las cuales puede tomar diferentes valores en un dominio dado.

El modelo propuesto trata la información sin realizar la transformación a características booleanas. Haciendo un mejor uso de la información disponible y aplicando ciertas medidas de diversidad, correlación y contingencia, el sistema realiza una ponderación de las características descriptivas de los productos, que aplicada al proceso de filtrado, proporciona recomendaciones más acordes con los gustos o necesidades del usuario.

En el caso de estudio se demuestra la validez del modelo propuesto, revelando que, para un número de recomendaciones menor o igual a 4, obtiene mejores resultados que los modelos básicos. En términos de eficiencia, el modelo propuesto mejora en tiempo de construcción del Sistema de Recomendación Basado en Contenido con ponderación de características basado en TF-IDF y en tiempo de recomendación medio a este y al Sistema de Recomendación Basado en Contenido con Modelado en Espacios Vectoriales Booleano Básico.

Derivado del conocimiento adquirido en la realización de esta investigación y la memoria, como trabajo futuro, se estudiarán posibles mejoras a este método:

- Estudio de otros métodos de ponderación de características, que permitan revelar interacciones entre múltiples características y la valoración que otorga el usuario a los productos.
 - Hibridación de este sistema con otros métodos de recomendación clásicos, con el objetivo de mejorar las recomendaciones que el usuario recibe.
 - Ampliación del tratamiento de las características para permitir que una características de un producto pueda soportar múltiples valores. Por ejemplo, una película puede tener varios protagonistas a la vez, pertenecer a varios géneros, etc.
-

Asimismo, teniendo en cuenta la importancia actual de los Sistemas de Recomendación, así como las múltiples líneas de investigación existentes en este área, nuestros trabajos futuros pretenden abordar los siguientes temas:

- Estudio de los Sistemas de Recomendación para grupos y la posible adaptación, para su funcionamiento en esta línea, del método propuesto en esta investigación.
 - Ampliación del sistema actual para su aplicación en recomendación sensible al contexto y en recomendación multi-criterio.
-

Bibliografía

- [1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, 23(1):103–145, 2005.
- [2] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Recommender Systems Handbook*, chapter 3, pages 217–253. Springer US, 2011.
- [3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
- [4] Akiko Aizawa. An information-theoretic perspective of tf-idf measures. *Information Processing and Management*, 39(1):45–65, 2003.
- [5] Slah Alsaleh, Richi Nayak, Yue Xu, and Lin Chen. Improving matching process in social network using implicit and explicit user information. In *Proceedings of the 13th Asia-Pacific web conference on Web technologies and applications*, pages 313–320, Berlin, Heidelberg, 2011. Springer-Verlag.
- [6] Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, and Nuria Oliver. The wisdom of the few: a collaborative filtering approach based on expert opinions from the

-
- web. In *Proceedings of the 32nd International ACM SIGIR Conference*, pages 532–539, New York, NY, USA, 2009. ACM.
- [7] Amazon.com. Amazon.com: online shopping for electronics, apparel, etc., 2006.
- [8] Sarabjot Singh Anand, Patricia Kearney, and Mary Shapcott. Generating semantically enriched user profiles for web personalization. *ACM Transactions on Internet Technology*, 7(4), 2007.
- [9] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, March 1997.
- [10] Xinlong Bao, Lawrence Bergman, and Rich Thompson. Stacking recommendation engines with additional meta-features. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 109–116, New York, NY, USA, 2009. ACM.
- [11] Manuel J. Barranco and Luís Martínez. A method for weighting multi-valued features in content-based filtering. In *Proceedings of the 23rd international conference on Industrial engineering and other applications of applied intelligent systems - Volume Part III*, IEA/AIE'10, pages 409–418, Berlin, Heidelberg, 2010. Springer-Verlag.
- [12] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1-2):115–153, 2001.
- [13] Daniel Billsus, Clifford A. Brunk, Craig Evans, Brian Gladish, and Michael Pazzani. Adaptive interfaces for ubiquitous web access. *Interacting with Computers of the ACM - The Adaptive Web*, 45(5):34–38, 2002.
- [14] Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98*, pages 46–54, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [15] Yvonne M. M. Bishop, Stephen E. Fienberg, and Paul W. Holl. Discrete multivariate analysis: Theory and practice, 1995.
-

-
- [16] Yolanda Blanco-Fernandez, Martin Lopez-Nores, Alberto Gil-Solla, Manuel Ramos-Cabrer, and Jose J. Pazos-Arias. Exploring synergies between content-based filtering and Spreading Activation techniques in knowledge-based recommender systems. *Information Sciences*, 181(21):4823–4846, 2011.
- [17] Yolanda Blanco-Fernández, José J. Pazos-Arias, Martín López-Nores, Alberto Gil-Solla, and Manuel Ramos-Cabrer. Avatar: an improved solution for personalized tv based on semantic inference. *IEEE Transactions on Consumer Electronics*, 52(1):223–231, 2006.
- [18] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [19] Derek Bridge and Alex Ferguson. An expressive query language for product recommender systems. *Artificial Intelligence Review*, 18(3-4):269–307, 2002.
- [20] Robin Burke. Integrating Knowledge-based and Collaborative-filtering Recommender Systems. In *Workshop on AI and Electronic Commerce*, 1999.
- [21] Robin Burke. The wasabi personal shopper: a case-based recommender system. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 844–849, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.
- [22] Robin Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69(32):175–186, 2000.
- [23] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modelling and User-Adapted Interaction*, 12(4):331–370, 2002.
- [24] Robin Burke. Interactive critiquing for catalog navigation in E-commerce. *Artificial Intelligence Review*, 18(3-4):245–267, 2002.
-

-
- [25] Robin Burke. Hybrid web recommender systems. In *The adaptive web*, pages 377–408. 2007.
- [26] Robin Burke, Kristian J. Hammond, and Benjamin C. Young. The findme approach to assisted browsing. *IEEE Expert: Intelligent Systems and Their Applications*, 12(4):32–40, 1997.
- [27] Robin Burke, Michael P. O’Mahony, and Neil J. Hurley. Robust collaborative recommendation. In *Recommender Systems Handbook*, chapter 25, pages 805–835. Springer US, 2011.
- [28] Li Chen and Pearl Pu. Interaction design guidelines on critiquing-based recommender systems. *User Modelling and User-Adapted Interaction*, 19(3):167–206, 2009.
- [29] Christina Christakou and Andreas Stafylopatis. A hybrid movie recommender system based on neural networks. In *Intelligent Systems Design and Applications, 2005. ISDA ’05. Proceedings. 5th International Conference on*, pages 500 – 505, 2005.
- [30] Sergio Cleger-Tamayo, Juan M. Fernandez-Luna, and Juan F. Huete. Top-N news recommendations in digital newspapers. *Knowledge-Based Systems*, 27:180–189, 2012.
- [31] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions*, 13(1):21–27, 1967.
- [32] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.
- [33] Byron Leite Dantas Bezerra and Francisco de Assis Tenorio de Carvalho. A symbolic approach for content-based information filtering. *Information Processing Letters*, 92(1):45–52, 2004.
- [34] Byron Leite Dantas Bezerra and Francisco de Assis Tenorio de Carvalho. Symbolic data analysis tools for recommendation systems. *Knowledge and Information Systems*, 26(3):385–418, 2011.
-

-
- [35] Luís M. de Campos, Juan M. Fernandez-Luna, and Juan F. Huete. A collaborative recommender system based on probabilistic inference from fuzzy observations. *Fuzzy Sets and Systems*, 159(12):1554–1576, 2008.
- [36] Luís M. de Campos, Juan M. Fernandez-Luna, Juan F. Huete, and Miguel A. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*, 51(7):785–799, 2010.
- [37] Marco Degenmis, Pasquale Lops, and Giovanni Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17(3):217–255, 2007.
- [38] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.
- [39] Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, chapter 4, pages 107–144. Springer US, 2011.
- [40] Pedro Domingos and Michael Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.
- [41] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Zanker. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 11(2):11–34, 2006.
- [42] Alexander Felfernig, Monika Mandl, Juha Tiihonen, Monika Schubert, and Gerhard Leitner. Personalized user interfaces for product configuration. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 317–320, New York, NY, USA, 2010. ACM.
-

-
- [43] Daniel M. Fleder and Kartik Hosanagar. Recommender systems and their impact on sales diversity. In *Proceedings of the 8th ACM Conference on Electronic Commerce*, pages 192–199, New York, NY, USA, 2007. ACM.
- [44] Peter W. Foltz and Susan T. Dumais. Personalized information delivery: an analysis of information filtering methods. *Interacting with Computers of the ACM*, 35(12):51–60, 1992.
- [45] Nikolaus Franke, Martin Schreier, and Ulrike Kaiser. The i designed it myself; effect in mass customization. *Management Sciences*, 56(1):125–140, 2010.
- [46] Simon Funk. Netflix update: Try this at home, 2006.
- [47] Min Gao, Kecheng Liu, and Zhongfu Wu. Personalisation in web computing and informatics: Theories, techniques, applications, and future research. *Information Systems Frontiers*, 12(5):607–629, 2010.
- [48] Fatih Gedikli and Dietmar Jannach. Neighborhood-restricted mining and weighted application of association rules for recommenders. In *Proceedings of the 11th International Conference on Web Information Systems Engineering*, pages 157–165, Berlin, Heidelberg, 2010. Springer-Verlag.
- [49] Mustansar Ghazanfar and Adam Prugel-Bennett. An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering. In *The 2010 IAENG International Conference on Data Mining and Applications*, 2010.
- [50] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Interacting with Computers of the ACM*, 35:61–70, 1992.
- [51] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.
-

-
- [52] Anna Goy, Liliana Ardissono, and Giovanna Petrone. The adaptive web. chapter Personalization in e-commerce applications, pages 485–520. Springer-Verlag, Berlin, Heidelberg, 2007.
- [53] Steven Guan, Chon Seng Ngoo, and Fangming Zhu. Handy broker: an intelligent product-brokering agent for m-commerce applications with user preference tracking. *Electronic Commerce Research and Applications*, 1(3-4):314–330, 2002.
- [54] Byeong-Jun Han, Seungmin Rho, Sanghoon Jun, and Eenjun Hwang. Music emotion classification and context-based music recommendation. *Multimedia Tools and Applications*, 47(3):433–460, 2010.
- [55] John A. Hartigan. *Clustering Algorithms (Probability & Mathematical Statistics)*. John Wiley & Sons Inc, 1975.
- [56] Félix Hernández del Olmo and Elena Gaudioso. Evaluation of recommender systems: A new approach. *Expert Systems with Applications*, 35(3):790–804, 2008.
- [57] Tzung-Pei Hong and Jyh-Bin Chen. Finding relevant attributes and membership functions. *Fuzzy Sets and Systems*, 103(3):389 – 404, 1999.
- [58] M. Anwar Hossain, Jorge Parra, Pradeep K. Atrey, and Abdulmotaleb Saddik. A framework for human-centered provisioning of ambient media services. *Multimedia Tools and Applications*, 44(3):407–431, 2009.
- [59] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24, 1933.
- [60] Chein-Shung Hwang and Pei-Jung Tsai. A collaborative recommender system based on user association clusters. In *Proceedings of the 6th International Conference on Web Information Systems Engineering*, pages 463–469, Berlin, Heidelberg, 2005. Springer-Verlag.
-

- [61] Dietmar Jannach. Fast computation of query relaxations for knowledge-based recommenders. *AI Interacting with Computers*, 22(4):235–248, 2009.
- [62] Dietmar Jannach and Johannes Liegl. Conflict-directed relaxation of constraints in content-based recommender systems. In *Proceedings of the 19th International Conference on Advances in Applied Artificial Intelligence: Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 819–829, Berlin, Heidelberg, 2006. Springer-Verlag.
- [63] Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Information Science and Statistics. Springer, corrected edition, 2002.
- [64] Buhwan Jeong, Jaewook Lee, and Hyunbo Cho. An iterative semi-explicit rating method for building collaborative recommender systems. *Expert Systems with Applications*, 36(3):6181–6186, 2009.
- [65] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant Features and the Subset Selection Problem. In *International Conference on Machine Learning*, pages 121–129, 1994.
- [66] Ian T. Jolliffe. *Principal Component Analysis*. Springer, second edition, 2002.
- [67] Carme Julia, Angel D. Sappa, Felipe Lumbreras, Joan Serrat, and Antonio Lopez. Predicting Missing Ratings in Recommender Systems: Adapted Factorization Approach. *International Journal of Electronic Commerce*, 14(2):89–108, 2009.
- [68] Martijn Kagie, Michiel Wezel, and Patrick J.F. Groenen. Map based visualization of product catalogs. In *Recommender Systems Handbook*, chapter 17, pages 547–576. Springer US, 2011.
- [69] M.G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.
-

-
- [70] Jae Kyeong Kim and Yoon Ho Cho. Using web usage mining and SVD to improve E-commerce recommendation quality. In *Intelligent Agents and Multi-Agent Systems*, volume 2891 of *Lecture Notes in Artificial Intelligence*, pages 86–97. Springer-Verlag Berlin, 2003.
- [71] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.
- [72] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [73] Janet L. Kolodner. An introduction to case-based reasoning. *Artificial Intelligence Review*, 6(1):3–34, 1992.
- [74] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
- [75] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, chapter 5, pages 145–186. Springer US, 2011.
- [76] Bruce Krulwich. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 1997.
- [77] Aristomenis S. Lampropoulos, Paraskevi S. Lampropoulou, and George A. Tsihrin-tzis. A cascade-hybrid music recommender system for mobile services based on musical genre classification and personality diagnosis. *Multimedia Tools and Applications*, 59(1):241–258, 2012.
- [78] Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.
-

-
- [79] Seok Kee Lee, Yoon Ho Cho, and Soung Hie Kim. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, 180(11):2142–2155, 2010.
- [80] Tong Queue Lee, Young Park, and Yong-Tae Park. A time-based approach to effective recommender systems using implicit feedback. *Expert Systems with Applications*, 34(4):3055–3062, 2008.
- [81] Wei-Po Lee. Towards agent-based decision making in the electronic marketplace: interactive recommendation and automated negotiation. *Expert Systems with Applications*, 27(4):665 – 679, 2004.
- [82] Cane Wing-ki Leung, Stephen Chi-fai Chan, and Fu-lai Chung. A collaborative filtering framework based on fuzzy association rules and multiple-level similarity. *Knowledge and Information Systems*, 10(3):357–381, 2006.
- [83] Cane Wing-ki Leung, Stephen Chi-fai Chan, and Fu-lai Chung. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, 21(7):515–529, 2008.
- [84] Weiyang Lin, Sergio A. Alvarez, and Carolina Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6(1):83–105, 2002.
- [85] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, 1988.
- [86] Stanley Loh, Fabiana Lorenzi, Ramiro Saldaña, and Daniel Licthnow. A tourism recommender system based on collaboration and text analysis. *Information Technology and Tourism*, 6, 2004.
- [87] Pasquale Lops, Marco Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, chapter 3, pages 73–105. Springer US, 2011.
-

-
- [88] Fabiana Lorenzi and Francesco Ricci. Case-based recommender systems: A unifying view. *Intelligent Techniques for Web Personalization*, pages 89–113, 2005.
- [89] Leandro Balby Marinho, Alexandros Nanopoulos, Lars Schmidt-Thieme, Robert Jäschke, Andreas Hotho, Gerd Stumme, and Panagiotis Symeonidis. Social tagging recommender systems. In *Recommender Systems Handbook*, chapter 19, pages 615–644. Springer US, 2011.
- [90] Luís Martínez, Manuel J. Barranco, Luís G. Pérez, and Macarena Espinilla. A knowledge based recommender system with multigranular linguistic information. *International Journal of Computational Intelligence Systems*, 1(3):225–236, 2008.
- [91] Judith Masthoff. Group recommender systems: Combining individual models. In *Recommender Systems Handbook*, chapter 21, pages 677–702. Springer US, 2011.
- [92] José Ramón San Cristóbal Mateo. Multi-attribute utility theory. In *Multi Criteria Analysis in the Renewable Energy Industry*, Green Energy and Technology, pages 63–72. Springer London, 2012.
- [93] Donald Michie, D. J. Spiegelhalter, C. C. Taylor, and John Campbell, editors. *Machine learning, neural and statistical classification*. Ellis Horwood, Upper Saddle River, NJ, USA, 1994.
- [94] Koji Miyahara and Michael J. Pazzani. Collaborative filtering with the simple bayesian classifier. In *Proceedings of the 6th Pacific Rim international conference on Artificial intelligence*, PRICAI'00, pages 679–689, Berlin, Heidelberg, 2000. Springer-Verlag.
- [95] D. Nichols. Implicit rating and filtering. In *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36. ERCIM, 1998.
- [96] Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup Workshop at SIGKDD'07, 13th ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 39–42, 2007.
-

- [97] Michael J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.
- [98] Witold Pedrycz, Petr Ekel, and Roberta Parreiras. *Fuzzy Multicriteria Decision-Making*. John Wiley & Sons, Ltd, 2010.
- [99] Pearl Pu, Li Chen, and Pratyush Kumar. Evaluating product search and recommender systems for E-commerce environments. *Electronic Commerce Research*, 8(1-2):1–27, 2008.
- [100] James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Dynamic critiquing. In *Advances in Case-Based Reasoning*, volume 3155, pages 763–777. Springer-Verlag Berlin, 2004.
- [101] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 175–186, New York, USA, 1994. ACM.
- [102] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
- [103] Elaine Rich. User modeling via stereotypes. In *Cognitive Science*, volume 3, pages 329–354. 1979.
- [104] J. Rocchio. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, 1971.
- [105] Rosa M. Rodríguez, Macarena Espinilla, Pedro J. Sánchez, and Luís Martínez-Lopez. Using linguistic incomplete preference relations to cold start recommendations. *Internet Research*, 20(3):296–315, 2010.
-

-
- [106] Ekkawut Rojsattarat and Nuanwan Soonthornphisaj. Hybrid recommendation: Combining content-based prediction and collaborative filtering. In *Intelligent Data Engineering and Automated Learning*, volume 2690 of *Lecture Notes in Computer Science*, pages 337–344. Springer-Verlag Berlin, 2003.
- [107] Edward Rolando Núñez-Valdéz, Juan Manuel Cueva Lovelle, Oscar Sanjuán Martínez, Vicente García-Díaz, Patricia Ordoñez de Pablos, and Carlos Enrique Montenegro Marín. Implicit feedback techniques on recommender systems applied to electronic books. *Computers in Human Behavior*, 28(4):1186–1193, 2012.
- [108] Giovanni M. Sacco. Dynamic taxonomies: A model for large information bases. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):468–479, 2000.
- [109] Badrul Sarwar, George Karypis, Joseph Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.
- [110] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems. In *Fifth International Conference on Computer and Information Science*, pages 27–28, 2002.
- [111] Silvia Schiaffino and Analia Amandi. Building an expert travel agent as a software agent. *Expert Systems with Applications*, 36(2, Part 1):1291–1299, 2009.
- [112] C. Schmitt, D. Dengler, and M. Bauer. The maut machine: An adaptive recommender system. In *In Proceedings of the ABIS Workshop*, 2002.
- [113] Michael Scholz. Implications of Consumer Information Behaviour to Construct Utility-based Recommender Systems A Prototypical Study. In *BLED 2010 Proceedings*, pages 314–328, 2010.
- [114] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, pages 623–656, 1948.
-

-
- [115] Upendra Shardanand and Pattie Maes. Social information filtering: algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pages 210–217, New York, NY, USA, 1995. ACM Press.
- [116] Gavin Shaw, Yue Xu, and Shlomo Geva. Using association rules to solve the cold-start problem in recommender systems. In *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 340–347. Springer-Verlag, 2010.
- [117] Beerud D. Sheth. A Learning Approach to Personalized Information Filtering. Master’s thesis, January 1994.
- [118] Steffen Staab and Rudi Studer. *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2nd edition, 2009.
- [119] Markus Stolze and Michael Ströbel. Dealing with learning in ecommerce product navigation and decision support: the teaching salesman problem. In *Proceedings of the Second Interdisciplinary World Congress on Mass Customization and Personalization*, 2003.
- [120] Ja-Hwung Su, Bo-Wen Wang, Chin-Yuan Hsiao, and Vincent S. Tseng. Personalized rough-set-based recommendation by integrating multiple contents and collaborative information. *Information Sciences*, 180(1):113–131, 2010.
- [121] Xiaoyuan Su and Taghi M. Khoshgoftaar. Collaborative filtering for multi-class data using Bayesian networks. *International Journal on Artificial Intelligence Tools*, 17(1):71–85, 2008.
- [122] Panagiotis Symeonidis, Alexandros Nanopoulos, and Yannis Manolopoulos. Feature-weighted user model for recommender systems. In *Proceedings of the 11th international conference on User Modeling*, pages 97–106. Springer-Verlag, 2007.
- [123] Paul Timmers. Business models for electronic markets. *Electronic Markets*, 8(2):3–8, 1998.
-

-
- [124] E. Triantaphyllou. *Multi-Criteria Decision Making Methodologies: A Comparative Study*, volume 44 of *Applied Optimization*. Dordrecht, 2000.
- [125] Paolo Viappiani, Boi Faltings, and Pearl Pu. Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research*, 27:465–503, 2006.
- [126] Paolo Viappiani, Pearl Pu, and Boi Faltings. Preference-based search with adaptive recommendations. *AI Interacting with Computers*, 21(2-3):155–175, 2008.
- [127] Patricia Victor, Martine Cock, and Chris Cornelis. Trust and recommendations. In *Recommender Systems Handbook*, chapter 20, pages 645–675. Springer US, 2011.
- [128] Patricia Victor, Martine De Cock, Chris Cornelis, and A Teredesai. Getting cold start users connected in a recommender system’s trust network. In *World Scientific Proceedings Series on Computer Engineering and Information Science*, volume 1, pages 877–882. World Scientific, 2008.
- [129] M. G. Vozalis and K. G. Margaritis. Using SVD and demographic data for the enhancement of generalized Collaborative Filtering. *Information Sciences*, 177(15):3017–3037, 2007.
- [130] Fu Lee Wang. Improvements to collaborative filtering systems. In *Computational and Information Science*, volume 3314 of *Lecture Notes in Computer Science*, pages 975–981. Springer-Verlag Berlin, 2004.
- [131] Michael J. Weiss. *The clustering of America*. Harper&Row, 1952.
- [132] Michael White, Robert A. J. Clark, and Johanna D. Moore. Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, 36(2):159–201.
- [133] Y. Y. Yao. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science and Technology*, 46(2):133–145, March 1995.
-

- [134] Markus Zanker. A collaborative constraint-based meta-level recommender. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 139–146, New York, NY, USA, 2008. ACM.
- [135] Markus Zanker and Markus Jessenitschnig. Case-studies on exploiting explicit customer requirements in recommender systems. *User Modelling and User-Adapted Interaction*, 19(1-2):133–166, 2009.
- [136] Markus Zanker, Markus Jessenitschnig, and Wolfgang Schmid. Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15(4):574–595, 2010.
- [137] Chun Zeng, Chun-Xiao Xing, and Li-Zhu Zhou. Similarity measure and instance selection for collaborative filtering. *International Journal of Electronic Commerce*, 8(4):115–129, 2004.
- [138] Lu Zhen, George Q. Huang, and Zuhua Jiang. An inner-enterprise knowledge recommender system. *Expert Systems with Applications*, 37(2):1703–1712, 2010.
-