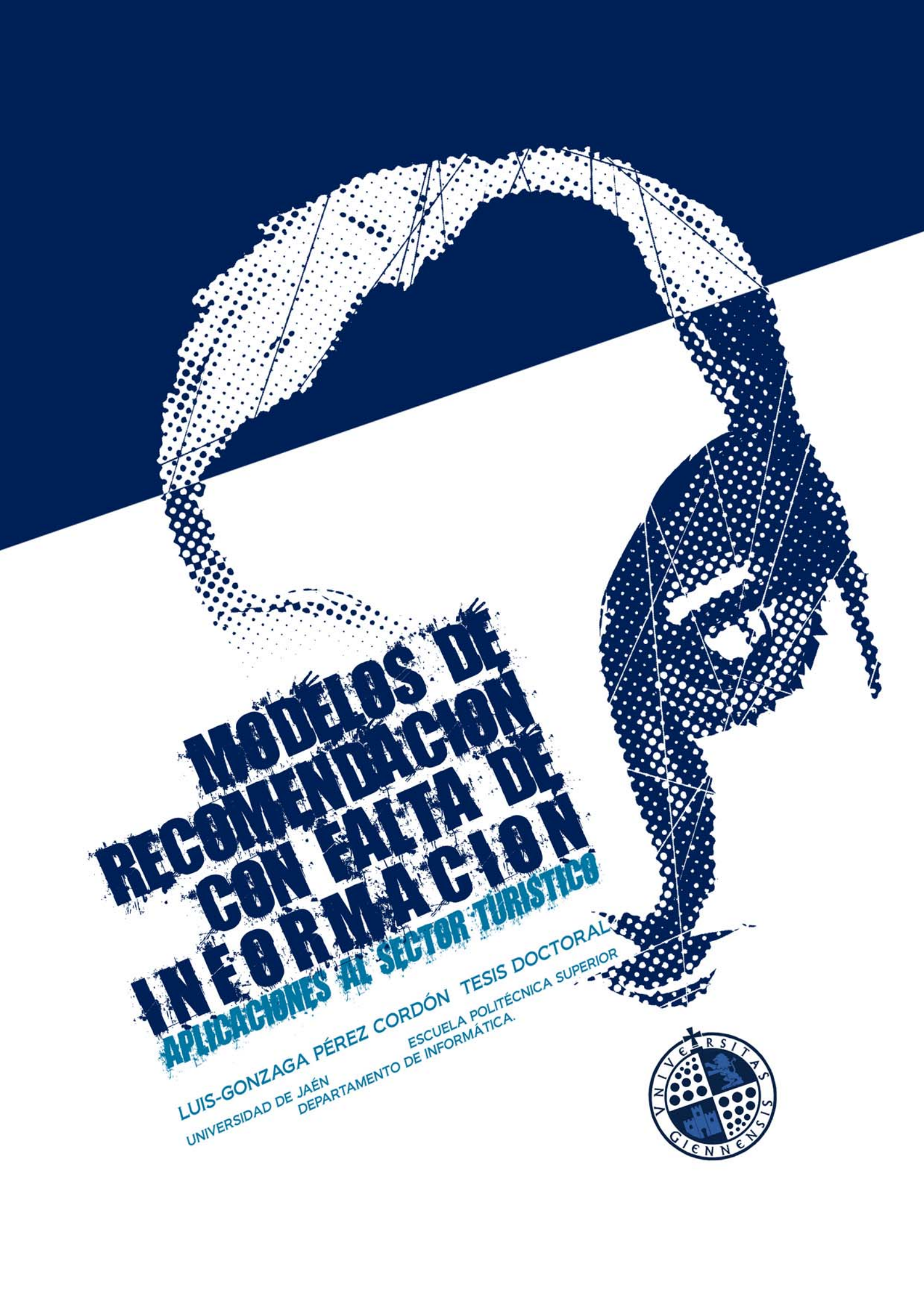


MODELOS DE RECOMENDACIÓN CON FALTA DE INFORMACIÓN

APLICACIONES AL SECTOR TURÍSTICO

LUIS-GONZAGA PÉREZ CORDÓN TESIS DOCTORAL
UNIVERSIDAD DE JAÉN ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INFORMÁTICA.

UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Jaén

Departamento de Informática

Modelos de recomendación con falta de información. Aplicaciones al sector turístico

Memoria de Tesis presentada por

Luis Gonzaga Pérez Cordón

para optar al grado de Doctor en Informática

Director

Dr. Luis Martínez López

Jaén Julio de 2008

La memoria de tesis titulada **Modelos de recomendación con falta de información. Aplicaciones al sector turístico**, que presenta **D. Luis Gonzaga Pérez Cordón** para optar al grado de Doctor en Informática, ha sido realizada en el **Departamento de Informática** de la Universidad de Jaén bajo la dirección del doctor **Luis Martínez López**.

Luis Gonzaga Pérez Cordón Doctorando	Dr. Luis Martínez López Director

Agradecimientos y Dedicatorias

Con mi agradecimiento y aprecio a la gente sin la cual la realización de esta tesis no habría sido posible:

En primer lugar y como no podía ser de otra forma, a mi director de tesis Dr. Luis Martínez López, por su dedicación, esfuerzo y confianza depositada en mí a lo largo de todo este tiempo.

A mis compañeros del Departamento de Informática de la Universidad de Jaén y en especial al grupo con el que comparto temas de trabajo e investigación, Paco, Pedro, Manolo y Macarena, por su ánimo y su amistad.

A mis compañeros y compañeras de la universidad de Jaén y muy encarecidamente a Beatriz Montes.

A Paco Herrera, por la oportunidad que me ofreció de incorporarme a esta apasionante profesión que es la docencia universitaria.

Y ya por último a mi familia y amigos, por el apoyo recibido, y en particular a Ruffy y a mi hermano José María, por haber diseñado la portada y contraportada de esta tesis doctoral.

Gracias a todos ...

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Motivación

En un principio mis primeras investigaciones se centraron en el modelado de preferencias y en la toma de decisiones. En concreto, estudiamos las relaciones de preferencias incompletas, métodos para completarlas, propiedades que debían cumplir, etc. Al mismo tiempo, buscamos campos de utilidad para estos estudios y nos fijamos en Internet.

Internet ha producido un cambio drástico en la sociedad de hoy en día. Gracias a Internet tenemos acceso on-line y de forma gratuita a enciclopedias, diccionarios de todo tipo, podemos consultar las noticias de todo el mundo, comprar cientos de productos o pagar por una serie de servicios. Las posibilidades que nos ofrece Internet son prácticamente infinitas y la mayor limitación que tenemos es el tiempo que podemos dedicar en hallar lo que necesitamos o simplemente en buscar algo que nos guste. Esta limitación en ciertos entornos supone un grave problema.

En Internet podemos encontrar muchos servicios o modelos de negocio que, en principio, se pensaron que tendrían un éxito arrollador, pero que a la hora de la verdad, han supuesto en muchas ocasiones pérdidas importantes de dinero,

cierres de compañías, despidos... Una de las áreas que más se ha visto afectada por esta perspectiva utópica de éxito ha sido la del comercio electrónico, la cual se desarrolló rápidamente cumpliendo mínimamente sus expectativas de éxito. Las compañías involucradas en el comercio electrónico ofrecían una gran variedad de productos, con el objetivo de satisfacer a miles o millones de potenciales clientes. En un principio, para el usuario esto suponía una importante ventaja que no tenía en la tienda tradicional, ya que, podía encontrar una gran variedad de productos e información relacionados con sus necesidades sin salir de casa. Sin embargo, esta variedad de productos a veces supone más que una ventaja, un inconveniente, pues ante una simple consulta de un usuario, era muy probable que esa tienda electrónica devolviera cientos o miles de productos relacionados con dicha consulta. No todos ellos satisfacían las necesidades del usuario, y de entre aquellos productos que podían satisfacer estas necesidades sólo unos pocos eran del gusto del usuario. Encontrar estos productos solía ser una tarea ardua, larga y muchos usuarios desistían de sus búsquedas, y preferían acudir a las tiendas tradicionales donde podían ser atendidos por dependientes, que fácilmente les ofrecían algún producto adecuado a sus necesidades aunque el espacio de búsqueda y la información recibida fuese de inferior calidad, dejando de esta forma de lado la compra de productos mediante tiendas electrónicas.

Para solucionar el problema anterior se desarrollaron varias herramientas en el ámbito del comercio electrónico. La que más éxito ha tenido han sido los sistemas de recomendación [30, 31, 41, 113, 166, 179, 192]. Estos sistemas ayudan a los usuarios en sus procesos de búsqueda en las tiendas electrónicas.

No existe un único modelo de sistema de recomendación, sino más bien una familia de modelos de sistemas de recomendación que comparten el mismo objetivo, guiar al usuario hacia aquellos productos que son de su interés, y la misma

estructura de funcionamiento: (i) parten de unos datos que ya están en el sistema antes de que el proceso de recomendación empiece, (ii) el usuario proporciona sus necesidades o intereses para que el sistema pueda generar las recomendaciones, (iii) se aplica un algoritmo que combina los datos de entrada del usuario, con los datos que ya tenía el sistema con el objetivo de generar las recomendaciones.

Sin embargo, estos sistemas de recomendación presentan una serie de inconvenientes en su utilización:

1. *El uso de escalas precisas:* muchos de los sistemas de recomendación actuales fuerzan a los usuarios a expresar sus preferencias mediante escalas precisas, normalmente numéricas. Aunque esto facilita el manejo de los datos, ya que, los usuarios trabajan en el mismo dominio que trabaja el sistema de recomendación, dificulta su uso. Principalmente porque los datos que estamos requiriendo están relacionados con los gustos, preferencias, percepciones u opiniones de los usuarios que son difíciles de valorar de forma precisa.

2. *Unicidad en la escala de evaluación de preferencias:* trabajan con una única escala independientemente de lo que se esté evaluando en ese momento, o el grado de conocimiento del usuario sobre esa materia o de la naturaleza de lo que estamos evaluando. Muchas de las operaciones internas de los sistemas de recomendación están diseñadas para trabajar con una única escala finita de valores. Es muy habitual que dicha escala sea también utilizada por los usuarios y/o expertos aún cuando grado de conocimiento no sea el mismo.

3. *Información histórica:* para generar las recomendaciones estos sistemas utilizan las valoraciones que han ido aportando los usuarios sobre los productos que han visto, probado o comprado. Si dicha base de datos histórica es lo

suficientemente buena, estos sistemas producen recomendaciones muy cercanas a lo que necesita el usuario, sin embargo, esta situación ideal no siempre se produce, y en ciertos entornos, es habitual que no tengamos información histórica sobre los usuarios, no pudiéndose utilizar técnicas clásicas para generar las recomendaciones.

4. *Dependencia de la densidad de la base de datos:* la densidad de la base datos en muchas ocasiones ha demostrado ser uno de los factores más importante en la calidad de las recomendaciones. Aunque en muchas situaciones, se puede obligar o motivar al usuario a que proporcione valoraciones para aumentar la densidad de la base de datos, en otros casos, esto es imposible o desaconsejable, ya sea por la naturaleza del problema o porque no se espera que el usuario interaccione con el sistema de una forma habitual y duradera.

Dado estos problemas enfocamos nuestra investigación al desarrollo de modelos de recomendación que solucionen o mitiguen los problemas anteriores y sean de aplicación a gran variedad de situaciones.

1.2. Objetivos

Dada la motivación de nuestra investigación, los objetivos de esta memoria son los siguientes:

- Estudiar las alternativas posibles que hay para modelar la información en este tipo de problemas, es decir, qué estructura se pueden utilizar, tipo de información, operaciones... Debemos tener en cuenta que la información que vamos a manejar es subjetiva, pues está relacionada con las opiniones, percepciones y gustos de los usuarios.

- Mejorar los procesos de adquisición de información para la construcción de perfiles de usuario. Buscamos requerir el mínimo de información a los usuarios y encontrar mecanismos para inferir a partir de esta información un perfil de usuario. Para ello utilizaremos relaciones de preferencia incompletas y algoritmos de reconstrucción.

- Diseñar modelos de sistema de recomendación capaces de realizar recomendaciones en situaciones donde la información histórica disponible sobre las valoraciones, opiniones o acciones pasadas del cliente no esté disponible o sea escasa.

- Implementar un prototipo de sistema de recomendación aplicado al sector turístico. Queremos implementar un prototipo funcional que implemente varios modelos, tanto clásicos como algunos de los modelos teóricos expuestos en esta memoria.

1.3. Estructura de la memoria

Esta memoria está estructurada en los siguientes capítulos:

- En el capítulo 2, haremos una revisión en profundidad de los sistemas de recomendación existentes en la actualidad, veremos su funcionamiento, sus ventajas y sus inconvenientes. Además veremos algunas de las técnicas de hibridación existentes entre sistemas de recomendación. En este capítulo haremos especial hincapié en los sistemas de recomendación basados en conocimiento, en contenido y colaborativos, pues, éstos serán los modelos que hemos desarrollado en esta memoria para alcanzar nuestros objetivos.

- En el capítulo 3, realizaremos una revisión del "Enfoque Lingüístico Difuso" [218] y de los modelos lingüísticos de representación de preferencias [76, 81]. Dado que un objetivo de la memoria es mejorar el modelado de las preferencias en los sistemas de recomendación, y dado que la información presente en los sistemas de recomendación es, en su mayoría, vaga e imprecisa, hemos considerado adecuado el uso del enfoque lingüístico difuso ya que, ha demostrado ser útil modelando este tipo de información.
  En primer lugar revisaremos el enfoque lingüístico difuso y finalmente el modelo lingüístico computacional basado en 2-tuplas [81].

- En el capítulo 4, presentaremos un modelo de sistema de recomendación basado en contenido adaptado a entornos donde no hay información histórica sobre los usuarios. Este modelo trabajará en un contexto lingüístico multigranular.

- En el capítulo 5, presentaremos dos modelos de sistema de recomendación basados en conocimiento. El primero es un modelo que ofrece un contexto lingüístico multigranular a los usuarios para expresar sus opiniones y mejora el modelo presentado en el capítulo anterior, desde el punto de vista de que se puede aplicar a un mayor número de productos. El segundo modelo pretende optimizar la recogida de información de las necesidades del usuario y la generación de perfiles de usuario. Para ello, proponemos el uso de relaciones de preferencia incompletas para la construcción de perfiles de usuario.

- En el capítulo 6, se presentará una implementación de un sistema de recomendación aplicada al sector turístico, en concreto a la recomendación de restaurantes de la provincia de Jaén, que utiliza la propuesta presentada en esta memoria y que fue desarrollado para los proyectos de la Consejería de

Comercio, Turismo y Deporte de la Junta de Andalucía obtenidos por el grupo de investigación del que soy miembro.

- Finalmente, se presentan las conclusiones más relevantes obtenidas a lo largo de la memoria. Además, haremos una breve descripción de las lineas futuras de investigación. Y se concluye con una recopilación bibliográfica que recoge las contribuciones más destacadas en la materia estudiada.

Capítulo 2

Sistemas de recomendación

En el mundo real, nos podemos encontrar tiendas adaptadas a las necesidades o características de un barrio o zona concreta. Sin embargo, es imposible que nos encontremos una tienda real que se pueda adaptar a las necesidades, gustos y características de cada uno de los clientes que la visitan. En Internet esto si es posible y ha supuesto un gran cambio tanto en el comercio en sí, como en las técnicas de marketing. Hemos pasado del comercio y de las técnicas de marketing clásicas que buscaban crear productos estándares, con un gran ciclo de vida y que pudieran contentar a grandes mercados homogéneos, a un comercio y a unas técnicas que buscan personalizar los servicios y los productos de forma que puedan contentar expresamente a cada usuario en concreto [98].

Aunque en un principio se pensó que la irrupción del comercio electrónico llevaría a las compañías a diseñar una gran variedad de productos que se pudieran adaptar a los gustos particulares de los usuarios o grupos de usuarios, el cambio más importante no lo han sufrido las empresas sino los usuarios, pues éstos, cuando visitan una tienda electrónica, tienen disponible una mayor gama de productos que si visitaran una tienda tradicional. Así por ejemplo, aunque una librería tradicional puede ofrecer cientos de libros distintos, una librería *on-line* aumenta esta

oferta a miles o millones de libros distintos. Este aumento en las posibles elecciones, también suponen un aumento de la información que el usuario debe procesar antes de escoger el producto que mejor satisface sus necesidades. Muchas veces esto produce una sobrecarga de información para el usuario, que puede sentirse saturado, ya que debe de explorar una extensa gama de productos y no tiene el tiempo suficiente para hacerlo o no quiere dedicarle tanto tiempo en encontrar el producto deseado. Esto supuso un gran inconveniente y fue uno de los principales obstáculos que frenaron el desarrollo del comercio electrónico, ya que lo usuarios muchas veces desistían de las búsquedas y terminaban acudiendo a las tiendas tradicionales. Para solventar este problema de sobrecarga de información, se crearon varias herramientas de personalización que se utilizan no sobre los productos en sí, sino sobre la forma en que son presentados en la tienda *on-line* [99].

La "personalización" es un mecanismo muy utilizado en otras áreas cuyo objetivo es adaptar o personalizar un servicio a los gustos o necesidades particulares de un usuario. Cuando hablamos de personalización web normalmente nos referimos a la personalización de las páginas web [117]. Esta personalización, ofrece contenidos adaptados y una presentación de las páginas adaptadas basándose en las preferencias particulares de cada usuario, en las visitas pasadas o en los intereses futuros. La idea de la personalización es ofrecer la información o los productos adecuados, de manera precisa, a la gente que los necesita. Una de las herramientas más utilizadas y que mejores resultados ha proporcionado en este campo han sido los sistemas de recomendación [30, 31, 128, 146, 126, 183].

Los sistemas de recomendación fueron definidos en un principio como aquellos sistemas que tenían como entrada de datos recomendaciones proporcionadas por sus usuarios, agregaba estas recomendaciones y las mostraba a los sujetos apropiados [166]. Actualmente, el significado del término "sistema de recomen-

dación" es más amplio, refiriéndose a cualquier sistema que produce recomendaciones individuales como salida, o que tiene el efecto de guiar al usuario de un modo personalizado a objetos útiles y/o interesantes dentro de un gran espacio de posibles opciones. Estos sistemas son muy atractivos en situaciones donde la cantidad de información que se ofrece al usuario supera ampliamente cualquier capacidad individual de exploración. Los sistemas de recomendación en la actualidad están integrados en muchas páginas web de comercio electrónico tales como Amazom.com o CDNow [179]. Los adjetivos "individualizado" e "interesante y útiles" que hemos usado en la definición de sistemas de recomendación son las características que los separan de los sistemas de recuperación de información o de los sistemas de soporte a la decisión.

En este capítulo, en la sección 2.1, revisaremos los modelos de sistema de recomendación más comunes y nos detendremos con mayor detalle en los utilizados en esta memoria. A continuación, en la sección 2.2, revisaremos varios sistemas de recomendación utilizados en la actualidad: Amazon.com, CDNOW, Drugstore.com, eBay, MovieFinder.com Reel.com o Filmaffinity.

2.1. Modelos de sistemas de recomendación

Los sistemas de recomendación son usados en el área del comercio electrónico para sugerir productos a los clientes o para proporcionarles información que les ayude a decidir qué productos puede comprar. Estos productos pueden ser recomendados basándose en las ventas más importantes, en información demográfica del cliente y/o en el análisis histórico de las compras pasadas del cliente para predecir futuras compras. Existen varias formas de recomendación, en éstas podemos incluir sugerencias de productos a los clientes, proporcionar información persona-

lizada sobre el producto, resumir las opiniones de una comunidad de usuarios o proporcionar las críticas de dicha comunidad [180]. Estas técnicas de recomendación son una parte fundamental en la personalización de una tienda electrónica porque ayuda al sitio a adaptarse por si mismo a cada cliente. Esta personalización puede verse como una adaptación de la tienda *on-line* a las necesidades particulares de cada usuario [98]. Desde este punto de vista, uno podría definir los sistemas de recomendación como una herramienta que permite la creación de una tienda virtual personalmente diseñada para cada cliente [180].

Los sistemas de recomendación mejoran las ventas de las tiendas electrónicas de tres formas distintas [180]:

1. **Convierte a los visitantes en compradores:** muchos visitantes de tiendas electrónicas suelen visitar las páginas con el único objetivo de echar un vistazo a los productos ofertados pero, sin ninguna intención inicial de compra. Los sistemas de recomendación pueden ayudar a que estos usuarios encuentren los productos que desearían comprar.

2. **Incrementa las ventas cruzadas:** los sistemas de recomendación aumentan las ventas cruzadas al sugerir productos adicionales a los clientes para que los compre. Si las recomendaciones son buenas, la media de pedidos realizados en esa tienda debería incrementarse. Por ejemplo, una tienda electrónica podría recomendar productos adicionales en el proceso de finalización de la compra (en la caja virtual), basándose en los productos que ya tiene en el carrito de la compra.

3. **Mejora la fidelidad de los clientes:** en un mundo virtual como el de Internet donde, las tiendas electrónicas de la competencia están sólo a uno o dos clicks de ratón, la fidelidad de los clientes se convierte en una parte

fundamental del éxito de cualquier tienda *on-line* [163, 164]. Los sistemas de recomendación mejoran esta fidelidad al crear un valor añadido en la relación de la tienda electrónica con el cliente. Las tiendas electrónicas utilizan los sistemas de recomendación para aprender las costumbres y gustos de sus clientes, y presentar recomendaciones personalizadas a las necesidades de cada uno de ellos. Si este proceso de personalización se ha realizado con éxito, existen muchas posibilidades de que estos clientes vuelvan a visitar el sitio web, y ésto hará a su vez, que se proporcione más información sobre sus gustos, que mejoren las recomendaciones y que a la larga tengamos unos clientes fieles a dicha tienda electrónica. Incluso si un competidor desarrollara un software similar a esta tienda, los clientes tendrían que dedicar el mismo tiempo que dedicaron en el primer caso para que la competencia conozca con la misma precisión sus gustos y necesidades. Algo que normalmente no hará mientras la tienda actual sea capaz de cubrir sus necesidades [100].

En esta sección, presentaremos una clasificación de los sistemas de recomendación [166, 179, 192] basada en las fuentes de información utilizadas para calcular sus recomendaciones. Para entender dicha clasificación, es importante señalar que los sistemas de recomendación utilizan:

1. *Datos de contexto:* la información que el sistema tiene antes de que el proceso de recomendación empiece

2. *Datos de entrada:* la información que el usuario comunica al sistema para generar una recomendación.

3. *Algoritmo de recomendación:* un algoritmo que combine los datos de contexto con los datos de entrada para generar las recomendaciones.

Los datos de entrada están relacionados con los gustos, necesidades y preferencias de los usuarios. Dependiendo de como se recoja esta información se puede clasificar en:

- *Información explícita* [2, 39, 53, 73, 75, 155, 162, 180]*:* información proporcionada por el usuario, en la mayoría de los casos a petición del sistema, aunque también puede ser introducida por terceras personas en base a cuestionarios, datos aportados, información ya disponible, etc., pero siempre contemplando información directamente proporcionada por el usuario. De esta forma, es dicho usuario el responsable de la veracidad de la información aportada.

- *Información implícita* [2, 39, 53, 73, 75, 152, 155, 162, 180]*:* información recogida automáticamente por el propio sistema en función del comportamiento del usuario. Pueden ser por ejemplo, datos referidos al historial de navegación del usuario, productos adquiridos en una página web, acceso a ciertas páginas, etc.

Partiendo de esta información y de los algoritmos utilizados para generar las recomendaciones podemos distinguir cinco técnicas de recomendación:

1. *Sistemas de recomendación colaborativos*[19, 28, 41, 64, 67, 72, 159, 165, 176, 182]*:* son probablemente los más extendidos en el mercado. Se han utilizado con éxito en múltiples ocasiones y es el modelo más asentado en la actualidad. Los sistemas de recomendación colaborativos agregan las valoraciones o recomendaciones de los objetos, identifican los gustos comunes de los usuarios basándose en sus valoraciones y generan una nueva recomendación teniendo en cuenta las comparaciones entre usuarios. Un perfil típico

de usuario en un sistema colaborativo consiste en un vector de objetos y sus valoraciones. La mayor ventaja de las técnicas colaborativas es que son completamente independientes de la representación interna de los productos que se pueden recomendar. En la sección 2.1.1 se explicarán con más detalle.

2. *Sistemas de recomendación basados en contenido*[7, 17, 128, 146, 126, 154, 183]*:* provienen de la investigación del filtrado y recuperación de información [16]. En un sistema basado en contenido, los objetos de interés están definidos por sus características asociadas. Un sistema de recomendación basado en contenido aprende un perfil de intereses de los usuarios basándose en las características presentes en los objetos que el usuario ha seleccionado. En la sección 2.1.2 los veremos más detalladamente.

3. *Sistemas de recomendación demográficos*[148, 155]*:* tienen como objetivo clasificar al usuario según sus atributos personales y hacer las recomendaciones basándose en sus clases demográficas. Uno de los primeros ejemplos de este tipo de sistema fue Grundy [167] que recomendaba libros basándose en la información personal recogida mediante un dialogo interactivo. Las respuestas del usuario eran comparadas con una librería construida manualmente de estereotipos de usuarios. Algunos sistemas de recomendación más recientes también han utilizado este modo de trabajo. Por ejemplo en [114] se utilizan grupos demográficos de estudios de marketing para sugerir una gama de productos y servicios. En otros sistemas, se utilizaron algoritmos de aprendizaje para conseguir un clasificador basado en datos demográficos [155]. La representación demográfica de la información en un modelo de usuario puede variar enormemente. El sistema de Rich [167] utilizaba atributos establecidos manualmente con valores de confianza numéricos. El modelo

de Pazzani [155] usa Winnow para extraer las características de las páginas de inicio de los usuarios que son las que predecían el gusto por ciertos restaurantes. Uno de los principales beneficios del enfoque demográfico es que no requiere información histórica, como ocurre con las técnicas colaborativas y basadas en contenido, para realizar las recomendaciones. Sin embargo, tiene como inconveniente que requiere información demográfica sobre el usuario, ésta muchas veces puede ser de carácter personal y muchos usuarios pueden sentir que se está violando su privacidad o que se le está requiriendo información de carácter privado.

4. *Sistemas de recomendación basados en conocimiento*[30, 31, 32, 33, 201]*:* los sistemas de recomendación basados en conocimiento intentan sugerir objetos haciendo inferencias sobre las necesidades de un usuario y sus preferencias. Se puede decir que todas las técnicas de recomendación hacen algún tipo de inferencia. El enfoque basado en conocimiento se distingue en el sentido que usan conocimiento funcional: tienen conocimiento sobre cómo un objeto en particular puede satisfacer las necesidades del usuario, y por lo tanto pueden razonar sobre la relación entre una necesidad y una posible recomendación. El perfil del usuario es una estructura de conocimiento que apoya esta inferencia. En la sección 2.1.3 veremos detalladamente el funcionamiento de este tipo de sistemas de recomendación.

5. *Sistemas de recomendación basados en utilidad [68]:* realizan sugerencias basándose en el cálculo de una utilidad de cada objeto para el usuario. Por supuesto, el problema central, es como crear una función de utilidad para cada usuario. Por ejemplo, TêTE-à-Tête y PersonalLogic tienen distintas técnicas para conseguir una función de utilidad específica para cada usuario

y aplicarla a los objetos en consideración [68]. El perfil de usuario por tanto, es una función de utilidad que el sistema tiene que obtener del usuario y el sistema emplea técnicas de satisfacción de restricciones para localizar el producto que mejor concuerda con lo que quiere el usuario. Las ventajas de las recomendaciones basadas en utilidad es que puede trabajar con atributos no relacionados directamente con los productos, tales como la fiabilidad del vendedor y la disponibilidad del producto.

6. *Sistemas de recomendación híbridos* [13, 31, 40, 155]: individualmente todas las técnicas presentan alguna limitación o problema. Para solucionar éstas limitaciones se planteó la hibridación de distintas técnicas de recomendación. Hablamos de hibridación cuando combinamos dos o más técnicas de recomendación con el objetivo de obtener un mejor rendimiento que si hubieramos utilizado estas técnicas de forma independiente. En la sección 2.1.5 las veremos más detalladamente.

En la tabla 2.1 podemos ver un resumen de las características principales de los sistemas de recomendación que hemos visto anteriormente. Suponemos que I es el conjunto de objetos sobre los cuales podemos hacer recomendaciones, U es el conjunto de usuarios cuyas preferencias son conocidas, u es el usuario para el que queremos generar recomendaciones, e i es un objeto para el que nos gustaría predecir las preferencias de u, $i \in I'$ con $I' \subseteq I$.

En las siguientes secciones revisaremos el funcionamiento de los sistemas de recomendación utilizados en el desarrollo de esta memoria ya sea para su implementación o para el desarrollo de nuevos modelos de recomendación a partir de ellos. En concreto estudiaremos por un lado los sistemas de recomendación colaborativos (sección 2.1.1) y los basados en contenido (sección 2.1.2). Por otro lado,

Tabla 2.1: Técnicas de recomendación

Técnicas	Datos de contexto	Datos de entrada	Algoritmo de recomendación
Colaborativo	Valoración por parte de U de los objetos de I	Valoraciones por parte de u de los objetos de I	Identifica usuarios en U similares a u y predice la valoración de i a partir de las valoraciones de esos usuarios
Basado en contenido	Características de los objetos de I	Valoraciones de u de objetos de I	Genera un clasificador que ajusta el comportamiento de las valoraciones de u y lo usa en i
Demográfico	Información demográfica sobre U y la valoración de los objetos de I	Información demográfica sobre u	Identifica a los usuarios que son similares demográficamente a u y a partir de las valoraciones de estos usuarios, predice la valoración de i
Basado en utilidad	Características de los objetos de I	Una función de utilidad sobre los objetos en I que describen las preferencias de u	Aplica una función a los objetos y determina la ordenación de i
Basado en conocimiento	Características de los objetos de I. Conocimiento de como estos objetos satisfacen las necesidades del usuario.	Una descripción de las necesidades o intereses de u	Infiere una relación entre i y las necesidades de u

también revisaremos los sistemas de recomendación basados en conocimiento (ver sección 2.1.3), dado que en esta memoria presentamos nuevos modelos basados en esta técnica. En la sección 2.1.4 haremos un resumen de las ventajas e inconvenientes que tienen cada una de las técnicas de recomendación vistas en este capítulo, y por último, en la sección 2.1.5 veremos las técnicas de hibridación más conocidas, pues han sido utilizadas en el software presentado en el capítulo 6.

2.1.1. Sistemas de recomendación colaborativos

Para revisar estos sistemas, estudiaremos cuáles fueron sus orígenes y en qué técnicas se basa su funcionamiento. Después, veremos una clasificación de los mismos y por último estudiaremos en profundidad el funcionamiento de un modelo simple de sistema de recomendación colaborativo.

Orígenes

Cuando las empresas pudieron almacenar los datos de las transacciones comerciales, éstas empezaron a analizar dicha información para comprender el comportamiento de sus clientes. El término *minería de datos* se utiliza para describir un conjunto de técnicas de análisis utilizado para inferir reglas o construir modelos a partir de un gran conjunto de datos [39, 53].

Los primeros sistemas de recomendación colaborativos basaban su esquema de funcionamiento en las ideas provenientes de la minería de datos. Así por ejemplo, se diseñaron sistemas de recomendación que tiene una fase *off-line* durante la cual aprenden el modelo, como ocurre en la minería de datos, y una fase *on-line* durante la cual aplican este modelo a situaciones reales generando recomendación para los clientes del sistema. Sin embargo, lo más habitual es utilizar un modelo

de aprendizaje relajado, en el cual se construye y se actualiza el modelo durante el funcionamiento del sistema, ya que, las bases de datos sobre clientes evolucionan dinámicamente conforme los usuarios interaccionan con el sistema. En las primeras tiendas de comercio electrónico, las recomendaciones se generaban a partir de simples consultas a una base de datos, muchas veces diseñadas por expertos y obtenidas mediante minería de datos.

Clasificación

Los sistemas de recomendación colaborativos pueden ser agrupados en dos clases generales [2, 28, 48, 102, 177, 180, 204, 222]:

- *Sistemas de recomendación que usan algoritmos basados en memoria* [165, 185]*:* estos sistemas de recomendación, esencialmente, realizan las recomendaciones basándose en la colección completa de ítems valorados previamente por el usuario. Es decir, el valor de una puntuación no conocida para un usuario en concreto sobre un producto en particular se calcula como un agregado de las valoraciones de otros usuarios (generalmente, los K más parecidos) sobre dicho producto. Para estos cálculos se utiliza el algoritmo de los vecinos más cercanos (K-NN) y como agregación se utiliza una media ponderada de los vecinos más cercanos que hayan valorado ese producto. Estas opiniones deberían ser escaladas para ajustar las diferencias en las tendencias de las valoraciones entre los usuarios [72]. Los algoritmos basados en el K-NN tienen la ventaja de ser capaces de utilizar rápidamente la información actual del sistema sobre las preferencias de los usuarios. En situaciones reales tienen un gran inconveniente: conforme la base de usuarios del sistema va creciendo, el tiempo y los recursos necesarios para generar

las recomendaciones crece. Por lo que, puede llegar a ser excesivo y no generar recomendaciones en tiempos aceptables. Además la calidad de éstas dependen de la densidad de la base de datos, si hay muchos usuarios con pocas recomendaciones es muy probable que no se generen recomendaciones adecuadas.

Para solucionar este problema se suelen emplear heurísticas para buscar buenos vecinos dentro de una gran población de clientes.

- *Sistemas de recomendación que usan algoritmos basados en modelo* [28, 41, 66, 205]*:* las aproximaciones basadas en modelos utilizan una colección de valoraciones para aprender un modelo, el cual será utilizado para generar las recomendaciones. Los técnicas más utilizadas para la construcción de estos modelos han sido:

  1. *Redes Bayesianas* [28, 41]*:* las redes bayesianas crean un modelo a partir de un conjunto de entrenamiento con un árbol de decisión, en donde cada nodo y cada lado representa información de los clientes. El modelo puede ser construido *off-line* y puede tardar horas o días en construirse. El modelo resultante es muy pequeño, rápido y en algunos casos tan preciso como los modelos basados en los vecinos más cercanos [28]. Las redes bayesianas pueden ser prácticas en situaciones en donde el conocimiento sobre las preferencias del usuario, cambia muy lentamente con respecto al tiempo que se tarda en construir modelo. No es útil o no es adecuado en situaciones en donde los modelos de preferencias de los clientes deben ser actualizados rápida y frecuentemente.
  2. *Sistemas de Recomendación basados en técnicas de clustering:* las técnicas de clustering identifican grupos de clientes que parecen tener pre-

ferencias similares. Una vez se han creado los cluster, las predicciones para un individuo pueden hacerse agregando las opiniones de otros clientes pertenecientes a ese cluster. Algunas técnicas de clustering tienen en cuenta que un cliente o usuario puede pertenecer parcialmente a varios clusters. En estos casos la predicción se obtiene mediante una agregación entre los cluster participantes, y ponderada por el grado de pertenencia. Las técnicas de clustering normalmente producen recomendaciones menos personales que otros métodos y en algunos casos, los clusters pueden tener peor precisión que los algoritmos del vecino más cercano [28]. Una vez hemos obtenido todos los clusters, el rendimiento del sistema es muy bueno, ya que, el tamaño del grupo que debe ser analizado es mucho más pequeño.

3. *Modelos basados en ítems* [176]*:* otro modelo propuesto son los algoritmos basados en ítems los cuales, para hacer una predicción de qué valoración le daría el usuario a un producto dado, llevan a cabo los siguientes pasos: primero tienen que explorar el conjunto de ítems o productos que el usuario ha valorado, calculan lo similar que son al producto del cual queremos predecir la puntuación que le daría el usuario, seleccionan los k productos más cercanos de un modelo ya calculado, y por último, predicen la valoración global como una media ponderada de las valoraciones del usuario de los productos similares.

4. *Sistemas de Recomendación basados en clasificadores:* los clasificadores son modelos computacionales para asignar una categoría a una entrada. Las entradas pueden ser vectores de categorías para los productos que han sido clasificados o datos sobre las relaciones entre los produc-

tos. Una forma de construir un sistema de recomendación utilizando un clasificador es utilizar la información sobre un producto o un cliente como entrada, y obtener una categoría de salida que represente la confianza con que se recomienda un producto al cliente.

Los clasificadores han tenido bastante éxito en una variedad de dominios, desde la identificación de riesgos de créditos y fraude en transacciones financieras, hasta diagnóstico médico o detección de intrusos. En [14] podemos encontrarnos con un sistema de recomendación híbrido que mezcla el filtrado colaborativo y el basado en contenido usando un clasificador de aprendizaje mediante inducción. En [66] se implementó un clasificador de películas, basado en vectores de características con aprendizaje por inducción y compara la clasificación con las recomendaciones basadas en el vecino más cercano. Este estudio demostró que el rendimiento de los clasificadores es ligeramente peor que el de el vecino más cercano, pero que combinándolos, pueden obtener mejores resultados que el del vecino más cercano por si sólo.

5. *Sistemas de Recomendación basados en reglas de asociación:* las reglas de asociación han sido utilizadas en procesos de negocio, para analizar tanto los patrones de preferencia entre los productos, y para recomendar productos a clientes basados en los otros productos que han escogido. Una regla de asociación expresa la relación de un producto que es habitualmente comprado junto con otros productos. Las reglas de asociación pueden formar una representación compacta de datos de preferencia que pueden mejorar la eficiencia del almacenamiento tanto como la del rendimiento. No aporta buenos resultados cuando

el conocimiento de las preferencias cambia rápidamente. Las reglas de asociación han tenido un gran éxito en otras aplicaciones tales como las que organizan las estanterías en las tiendas al detalle. Por el contrario, los sistemas de recomendación basados en técnicas del vecino más cercano son más fáciles de implementar para recomendaciones personales en un dominio donde se aportan frecuentemente opiniones, tales como, la venta al detalle *on-line*.

6. *Sistemas de Recomendación basados en Horting: horting* es una técnica basada en grafos en donde los nodos son clientes y los arcos indican el grado de similitud entre dos clientes [205]. Las predicciones son producidas desplazándose entre los nodos cercanos y combinando las opiniones de los clientes cercanos. Horting difiere del vecino más cercano en el sentido que se puede explorar en el grafo, a clientes que aún no han evaluado el producto en cuestión y, de esta forma, explorar relaciones transitivas que el vecino más cercano nunca consideraría. En [205] tenemos un estudio, utilizando datos sintéticos, en donde Horting produce mejores recomendaciones que un algoritmo del vecino más cercano.

Esquema de funcionamiento

En esta sección explicaremos brevemente el funcionamiento básico de los sistemas de recomendación colaborativos clásico (basado en memoria). Estos sistemas de recomendación basan su funcionamiento en la idea de que la gente que estaba de acuerdo en las valoraciones pasadas, también estará de acuerdo en las futuras. En el filtrado colaborativo [64, 72] las recomendaciones se realizan encontrando las correlaciones entre los usuarios del sistema de recomendación (ver figura 2.1). Con

Figura 2.1: Esquema de funcionamiento de un sistema de recomendación colaborativo

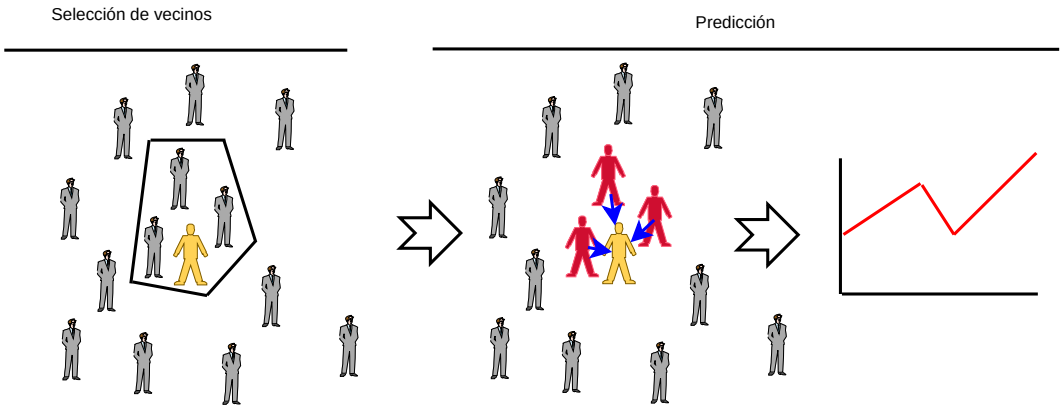


Tabla 2.2: Valoración de 5 usuarios sobre 5 restaurantes

	Marta	Luis	Carmen	Pedro	Paco
Kitima	-	+	+	+	-
Marco Polo	+	+	+	+	+
Spiga	+	-	+	-	+
Thai Touch	-	+	-	+	-
Dolce	+	-	+	-	?

esta metodología podemos encontrar fácilmente productos de interés (productos que no conoce el usuario actual pero que han sido valorado por otros usuarios afines a él) y predecir la valoración que el usuario actual le daría a los productos. Para entender su funcionamiento desarrollaremos el ejemplo que puede verse en la tabla 2.2. En esta tabla tenemos las valoraciones de 5 usuarios sobre 5 restaurantes. El símbolo "+" indica que al usuario le ha gustado la descripción del restaurante y el símbolo "-" indica que al usuario no lo ha gustado el restaurante

Para predecir la valoración que Paco le daría al restaurante Dolce, buscaremos usuarios que tenga un patrón de valoraciones similares a Paco. En este caso, Marta y Paco tienen idénticos gustos y uno podría suponer que a Paco le gustará Dolce porque a Marta le gusta. Un solución más general sería encontrar el grado de correlación entre Paco y los demás usuarios, en vez de usar solamente aquellos

usuarios que tiene el gusto más parecido a los de Paco, y usar un media ponderada de sus valoraciones para generar las recomendaciones. El peso dado a cada usuario será el grado de correlación entre los dos usuarios. En el caso más general, la valoración podría ser un número continuo en vez de solo +1 o -1. Se podría utilizar por ejemplo el coeficiente de correlación de Pearson r. Sea $R_{i,j}$ la valoración del usuario i sobre el producto j, entonces la correlación entre el usuario x y el usuario y se puede obtener de la siguiente forma:

$$r\left(x,y\right)=\frac{\sum_{d\in Productos}\left(R_{x,d}-\bar{R}_x\right)\left(R_{y,d}-\bar{R}_y\right)}{\sqrt{\sum_{d\in Productos}\left(R_{x,d}-\bar{R}_x\right)^2\left(R_{y,d}-\bar{R}_y\right)^2}}$$

Donde $\bar{R}_x$ es el valor medio de las valoraciones hechas por el usuario x.

En el ejemplo que estamos comentando, la correlación entre Paco y Marta es 1.0, entre Paco y Luis es 0.577, entre Paco y Carmen es 0.577, y entre Paco y Pedro es -0.577. Si calculamos la media ponderada de la valoración de cada usuario sobre el restaurante Dolce por su correlación con Paco obtendremos un valor de 0.682. Un algoritmo colaborativo predeciría que a Paco le gustaría el restaurante Dolce basándose en las recomendaciones de los otros usuarios. Podemos destacar que en esta recomendación se ha tenido en cuenta el hecho de que Paco y Pedro tienen prácticamente gustos opuestos y que a Pedro no le gusta el restaurante Dolce.

Aunque el filtrado colaborativo es comúnmente utilizado para encontrar correlaciones entre las valoraciones de los usuarios, también puede ser utilizado para encontrar "colaboraciones" entre los objetos valorados. Por ejemplo, hay una correlación perfecta entre las valoraciones de Dolce y Spiga en la tabla 2.2. Basándose en este hecho, podríamos predecir que a Paco le gustaría el restaurante Dolce ya que a Paco le gusta el Restaurante Spiga. De forma análoga, se puede calcular la correlación entre restaurantes, usando otra vez el coeficiente de Pearson r, y

hacer predicciones basadas en la media ponderada de las valoraciones de los otros restaurantes. De nuevo, haciendo la media ponderada de todos los restaurantes de la tabla 2.2 obtendríamos el resultado de que a Paco le gustaría el restaurante Dolce.

2.1.2. Sistemas de recomendación basados en contenido

En esta sección veremos cuáles fueron sus orígenes y en qué técnicas se basaron para la construcción de estos modelos de sistemas de recomendación. Por último, estudiaremos en detalle el funcionamiento de un modelo simple de sistema de recomendación basado en contenido.

Orígenes

Muchos de los productos que se pueden encontrar en Internet se describen con gran detalle mediante textos, descripciones, resúmenes... Los sistemas de recomendación basados en contenido explotan esta información mediante el uso de técnicas previamente utilizadas en la recuperación y filtrado de información

Como sabemos, uno de los usos más habituales de Internet es como fuente de información. La cantidad de información que podemos encontrar sobre un tema o relacionado con una consulta es enorme, sin embargo, no toda esta información es relevante o coincide con lo que realmente buscamos. Aunque, la mayoría de veces los usuarios pueden obtener lo que quieren utilizando los motores de búsqueda tradicionales, en ocasiones estos devuelven muchos resultados que no son de interés real para el usuario. El problema de encontrar la información relevante es conocido como sobrecarga de información. Los factores que causan este problema son los siguientes:

- La gran cantidad de información existente y creciente en Internet.
- La información mostrada en Internet no suele estar estructurada y la información nueva se añade sin ningún orden ni estructura.
- La información existente en Internet tiene una naturaleza dinámica. Puede ser cambiada, modificada o borrada aunque ya haya sido publicada.

El hecho de que se hayan convertido en una de las fuentes de información más importantes y más utilizadas ha hecho que en muchas páginas web aparezcan herramientas para:

- *Recuperación de información* [10, 174] *:* el objetivo de la recuperación de información es devolver un documento, o varios, a un usuario final que contenga la información deseada por este usuario. Esta área ha evolucionado desde el análisis de letras y palabras, para obtener el tema central del documento, hasta la integración de propiedades intrínsecas del documento tales como citas, hiperenlaces o datos del uso del documento. Los enfoques basados en contenido, tales como, los estadísticos o técnicas de lenguaje natural proporcionan resultados que contienen un conjunto específico de palabras o significados, pero no diferencian qué documentos en una colección son los que de verdad interesan leer. Salton [174], un pionero en recuperación de información desde 1970, introdujo el modelo espacial vectorial que se ha convertido en la actualidad en la base para la representación de documentos en los sistemas de recuperación de información actuales y los motores de búsqueda web.
- *Filtrado de información*[16] *:* las técnicas de filtrado de información son un complemento de los motores de búsqueda más que un modelo alternativo. El

> concepto de "filtrado" tiene que ver con la decisión de considerar, a priori, si un documento es relevante o no, eliminándolo del conjunto de elementos que se pueden recuperar en caso contrario. Estas técnicas combinan, normalmente, sistemas de auto aprendizaje y sistemas de recuperación de información y han sido utilizadas en la construcción de motores de búsqueda especializados [37].

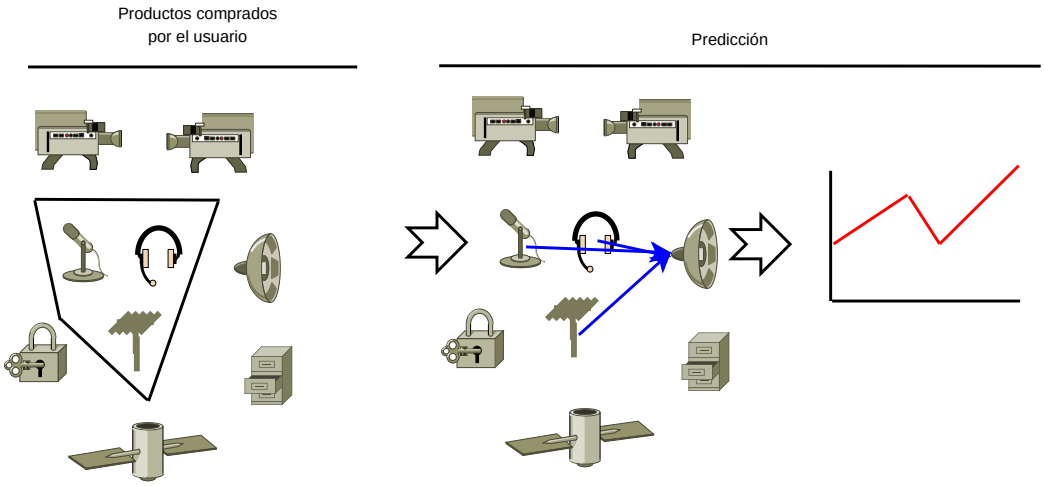
Debido a los grandes avances que se hicieron en estos campos, muchos de los sistemas de recomendación basados en contenido actuales se centran en recomendar productos o ítems que contienen información textual tales como documentos, páginas web (URLs), o mensajes de noticias de Usenet. Las mejoras sobre los enfoques de recuperación de información tradicionales provienen del uso de perfiles de usuario que contienen información sobre los gustos, preferencias o necesidades de estos. La información almacenada en estos perfiles se puede obtener de muy diversas formas: de forma explícita, mediante, por ejemplo, cuestionarios que el usuario debe rellenar, o de forma implícita, obteniendo información estudiando su comportamiento (que ítems o productos lee, etc.).

Esquema de funcionamiento

En esta sección explicaremos brevemente el funcionamiento básico de los sistemas de recomendación basados en contenido. Estos sistemas, analizan las descripciones de los productos que han sido valorados por el usuario y predicen qué productos le pueden gustar (ver figura 2.2). El funcionamiento de estos sistemas dependerá en gran medida del tipo de información de descripción que se utilice en la construcción del perfil de usuario. Esta información se puede dividir en:

- *Conjunto de características:* asociado a cada producto puede aparecer un

Figura 2.2: Esquema de funcionamiento de un sistema de recomendación basado en contenido

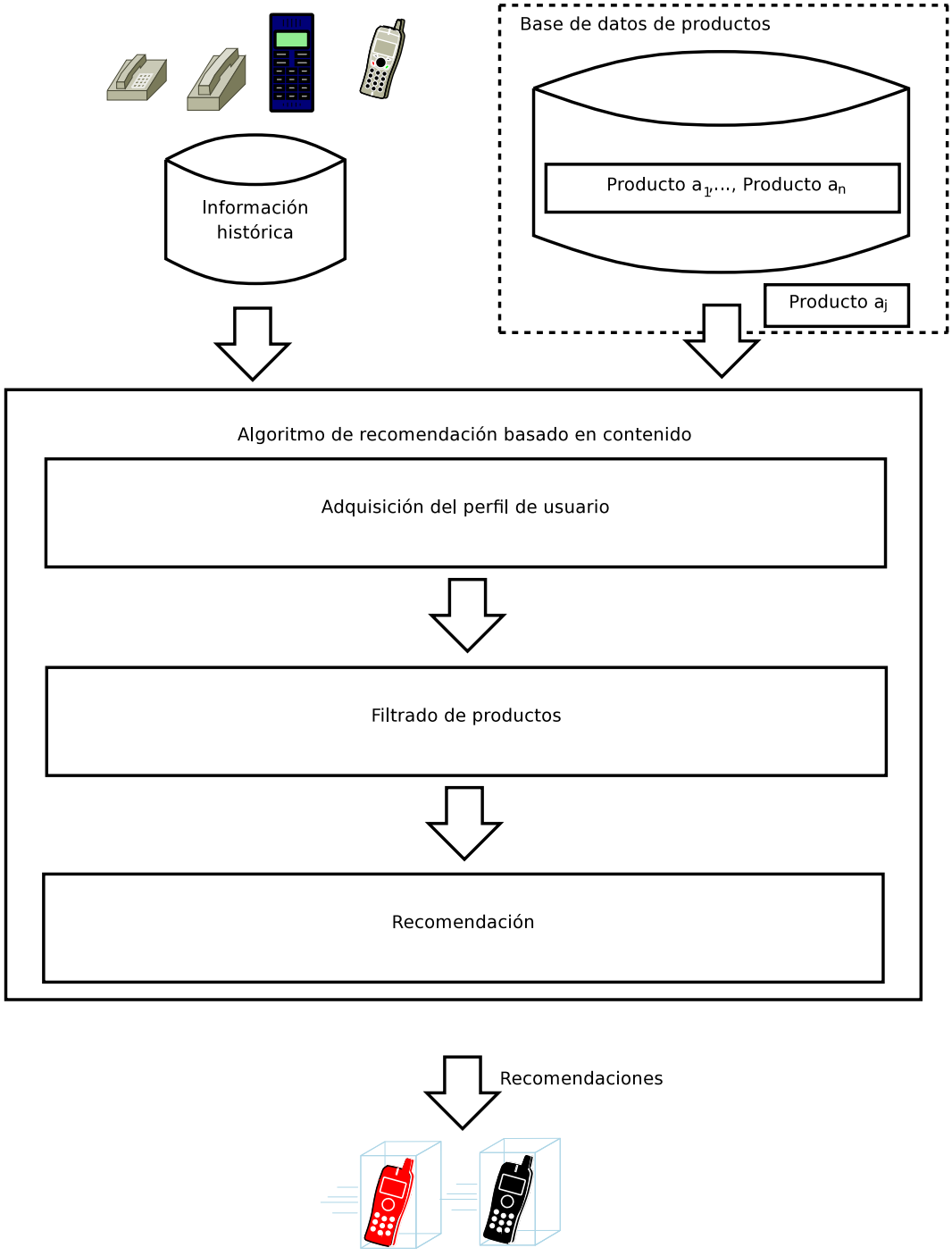


conjunto de características que lo describe, como por ejemplo el autor de un libro, serie a la que pertenece, actores de una película, consumo medio si estamos hablando de vehículos...

- *Información textual sobre el producto:* es un documento que describe dicho producto (un resumen de un libro, el argumento de una película,...). La principal diferencia con respecto al conjunto de características, es que está información no está estructurada.

Existen sistemas basados en contenido que solo trabajan con los conjuntos de características asociados a cada producto. Éstos tienen un esquema de funcionamiento similar al que podemos ver en la figura 2.3. Analizan las características de los productos comprados o valorados positivamente por el usuario. Construyen un perfil de usuario y por último, usan este perfil de usuario para buscar nuevos productos que puedan satisfacer al usuario y los recomienda.

Figura 2.3: Funcionamiento sistema de recomendación basado en contenido



Por ejemplo, en [17], se presenta un sistema de recomendación basado en contenido de películas. En este sistema, el perfil de usuario está compuesto, mayorita-

riamente, por el conjunto de características deseables en una película. Una de las características utilizadas en este sistema son los actores preferidos por el usuario. Así por ejemplo, supongamos que tenemos un usuario que ha visto la película "La milla verde" y "Soñador", y que ha puntuado dichas películas con un 4 (buena) la primera y con un 5 (muy buena) la segunda (ver tabla 2.3).

Tabla 2.3: Descripción de dos películas

Característica	La milla verde	Soñador
Reparto	T. Hanks, D. Morse	K. Russell, D.Morse
...	...	...
Puntuación	4	5

Para generar las recomendaciones para este usuario, el sistema construirá el perfil de usuario, partiendo de las características de las películas valoradas positivamente por él. Si nos centramos unicamente en la característica "Reparto" veremos que el perfil obtenido (ver tabla 2.4) contiene los actores y actrices de las películas valoradas positivamente por él. Además, podemos ver que que cada uno de ellos tiene asociado un peso que mide el grado de preferencia del usuario por dicho actor o actriz. A partir de este perfil, el sistema buscará aquellas películas que cumplan en mayor grado con las características almacenadas en él y las recomendará.

Tabla 2.4: Perfil de usuario basado en contenido

Característica	Perfil de usuario
Reparto	$(\{T.Hanks, D.Morse, K.Russell\}, \{0.25, 0.50, 0.25\})$
...	

Sin embargo, la mayoría de las técnicas clásicas de recomendación basadas en contenido analizan únicamente la información textual del producto. El objetivo de este análisis es encontrar patrones en el contenido de dichos textos para realizar las recomendaciones. Muchos de estos sistemas de recomendación son versiones especializadas de clasificadores entrenables donde su objetivo es aprender una función que predice a qué clase pertenece un documento (si el documento le gusta o no le gusta). Otros algoritmos tratan este problema como un problema de regresión en donde el objetivo es aprender una función que predice un valor numérico (la valoración del producto). Existen dos problemas importantes en el diseño de un sistema de filtrado basado en contenido. El primero es encontrar una representación de documentos. El segundo es crear un perfil de usuario que nos permita recomendar productos al usuario que aún no haya visto.

En este tipo de sistemas de recomendación los productos se representan mediante documentos y éstos se modelan por medio de un vector de palabras "importantes". Por ejemplo, [13] representa los documentos en términos de las 100 palabras con los pesos más altos en el TF-IDF [174], o lo que es lo mismo, con las palabras que aparecen más frecuentemente en estos documentos de lo que hacen en media. En [154] se representa los documentos con las 128 palabras más informativas, palabras que están más asociadas con un documento que con otro. En la tabla 2.5 podemos ver un ejemplo con 5 restaurantes y 5 palabras que aparecen en las descripciones de los restaurantes. Las valoraciones de Paco sobre estos restaurantes también las podemos encontrar en la misma tabla.

Una vez que hemos definido una representación para los documentos, podemos entrenar un algoritmo de clasificación para que distinga los documentos que tendrían una valoración alta de los otros. Fab [13] utiliza el algoritmo de Rocchio [170] para aprender un vector TF-IDF que es la media de los documentos que

Tabla 2.5: Palabras contenidas en la descripción de 5 restaurantes junto con las valoraciones de un usuarios para estos restaurantes

	fideos	langostino	albahaca	exótico	salmón	Paco
Kitima	Y	Y	Y	Y	Y	-
Marco Polo		Y	Y			+
Spiga	Y		Y			+
Thai Touch	Y	Y		Y		-
Dolce		Y	Y		Y	?

tienen una valoración alta. Syskill & Weber [154] utiliza un clasificador Bayesiano para estimar la probabilidad de un documento sea interesante para el usuario. Ambos sistemas de recomendación tienen el inconveniente de que hay que definir a priori los términos que se van a utilizar en el perfil. En [155] utilizan el algoritmo Winnow [20, 124] para identificar las características relevantes de entre todas las posibles.

Para resolver este ejemplo con Winnow cada una de las palabras, x_i, asociadas a cada restaurante deberá ser tratada como una variable booleana (valdrá 1 si esta palabra está en la descripción del restaurante y 0 si no está). Winnow aprende los pesos w_i que están asociados con cada palabra de acuerdo con la inecuación:

$$\sum w_i x_i > \tau$$

donde τ es un umbral. Al principio todos los pesos son inicializados a 1. A continuación empieza una fase de aprendizaje cuyo objetivo es aprender los pesos asociados a cada palabra. Para ello, dividiremos el conjunto de restaurantes valorados por el usuario en un conjunto de entrenamiento y en un conjunto de prueba. A continuación ajustaremos los pesos de forma que la suma de los pesos de las palabras contenidas en la descripción de un restaurante supere el umbral, si ese restaurante le gusta al usuario, y se quede por debajo del umbral en caso contrario. Para hacer este ajuste, usaremos el conjunto de entrenamiento. Por cada

ejemplo de este conjunto se obtiene la suma de los pesos de las palabras presentes en su descripción. Si la suma está por encima del umbral y al usuario no le gustó, el peso asociado con cada palabra de este restaurante es dividido por 2. Si la suma está por debajo del umbral, y le gustaba el restaurante, el peso asociado a cada palabra es multiplicado por 2. En otro caso, el restaurante fue clasificado correctamente y no se realiza ningún cambio en los pesos.

Este proceso se repetirá continuamente ajustando los pesos hasta que todos los ejemplos del conjunto de prueba sean correctamente clasificados (o hasta que se hayan pasado n iteraciones sin ningún cambio en la precisión de las recomendaciones del conjunto de prueba).

Una vez hecho este proceso de aprendizaje, podremos conocer si a Paco le gustará o no el restaurante Dolce. Así por ejemplo, si después del entrenamiento obtenemos que los pesos asociados a las palabras fideos, langostino, albahaca, exótico, salmón son $\{0.5, 0.9, 0.8, 0.2, 0.9\}$ y el umbral seleccionado es 1.5 el valor de la inecuación será:

$$0.9 + 0.8 + 0.9 = 2.6 > 1.5$$

Y por lo tanto el restaurante Dolce será recomendado.

2.1.3. Sistemas de recomendación basados en conocimiento

Al igual que en los casos anteriores, en primer lugar estudiaremos cuáles fueron sus orígenes y en qué técnicas se basaron. Por último, estudiaremos el funcionamiento de los sistemas de recomendación basados en conocimiento.

Orígenes

La mayor parte de los sistemas de recomendación basados en conocimiento emplean técnicas de razonamiento basados en casos para inferir las recomendaciones. El razonamiento basado en casos fue formalizado en cuatro pasos:

1. *Recordar:* dado un determinado problema, el sistema debe recuperar aquellos casos relevantes (similares) a este problema. Un caso consiste en un problema, una solución y anotaciones de cómo la solución se llevo a cabo.

2. *Reutilizar:* adaptar la solución del problema anterior a este nuevo.

3. *Revisar:* probar que la solución puede ser aplicada al mundo real o al problema en concreto

4. *Retener:* si esta solución ha sido satisfactoria almacenarla como un nuevo caso para futuros problemas.

El razonamiento basado en casos tiene sus orígenes en los trabajos de Roger Shank [181], Janet Kolobner [110] y Michael Lebowitz [119]. El primer sistema razonador basado en casos fue CLAVIER utilizado para diseñar las piezas compuestas que se cocerán en un horno industrial de convección [93].

Esquema de funcionamiento

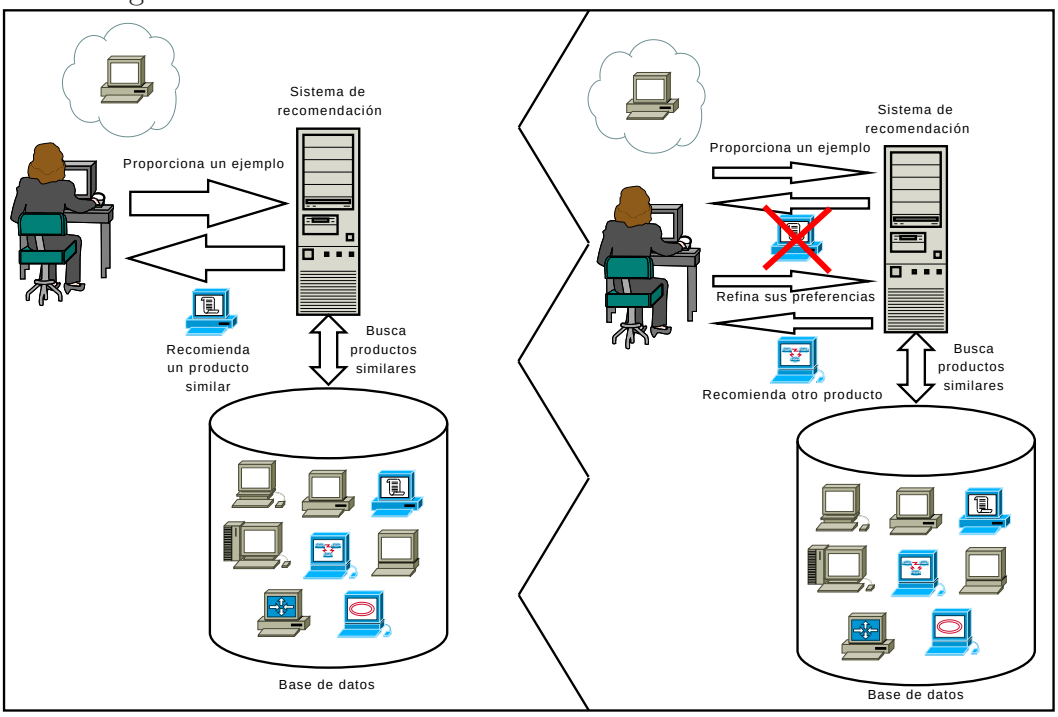
Este tipo de sistema de recomendación utiliza el conocimiento que tiene sobre los usuarios y los productos para, mediante un enfoque basado en dicho conocimiento, generar una recomendación razonando sobre qué productos satisfacen las necesidades del usuario. Existen varias alternativas para implementar este tipo de sistemas, por ejemplo, el sistema de recomendación PersonalLogic interacciona

con el usuario mediante diálogos que le permiten explorar un árbol de discriminación de las características de los productos. Otros han adaptado herramientas de soporte a la decisión con características cuantitativas para esta tarea [18]. En esta sección introduciremos los sistemas de recomendación que se basan en el razonamiento basado en casos [69, 111, 168] para realizar las recomendaciones. El sistema de recomendación de restaurantes Entree [30, 31, 32, 33] realiza las recomendaciones encontrando restaurantes en una nueva ciudad similares a los que el usuario conocía o le gustaban. Los usuarios que interaccionan con el sistema, declaran sus preferencias con respecto a un restaurante dando un ejemplo de sus necesidades. Posteriormente, si hace falta, éstos pueden refinar sus criterios de búsqueda.

Estos sistemas de recomendación presentan algunas ventajas sobre los sistemas de recomendación clásicos, los colaborativos y los basados en contenido. Por ejemplo, los sistemas de recomendación colaborativos necesitan una gran cantidad de valoraciones de los usuarios sobre un conjunto de productos para poder realizar recomendaciones precisas sobre dichos productos. Este problema también se presenta en aquellas técnicas de recomendación que emplean algoritmos de aprendizaje, ya que, para que estos puedan aprender o generar un modelo que represente las preferencias del usuario, se necesita una cierta cantidad de información sobre acciones pasadas o valoradas de los usuarios. Los sistemas de recomendación basados en conocimiento no presentan estos inconvenientes ya que sus recomendaciones no están basadas en las valoraciones de un grupo de usuarios. Esto hace que los sistemas de recomendación basados en conocimiento no sólo sean útiles por si mismos, sino también un importante complemento para otros tipos de sistemas de recomendación.

En la siguiente figura (figura 2.4) podemos ver un ejemplo de cómo funciona

Figura 2.4: Sistema de recomendación basado en conocimiento



este tipo de sistemas de recomendación, cuando utilizan el razonamiento basado en casos, para generar las recomendaciones. Los pasos que se siguen son los siguientes:

1. *Obtiene un ejemplo de las necesidades del usuario:* el usuario le proporciona al sistema un ejemplo que represente sus necesidades.

2. *Búsqueda de productos que cumplen las necesidades del usuario:* a partir de la descripción del ejemplo proporcionado por el usuario, el sistema de recomendación buscará otros productos similares a él y los devolverá como recomendaciones (parte izquierda de la figura 2.4).

3. *Fase de refinamiento:* en algunas circunstancias puede ocurrir que los productos recomendados, a partir del ejemplo no coincidan exactamente con las necesidades del usuario. En estos casos el usuario debe refinar las características del producto seleccionado como ejemplo, modificando y/o añadien-

do nuevas características (parte derecha de la figura 2.4).

Estos sistemas de recomendación utilizan el conocimiento que tienen sobre el usuario y los productos para inferir las recomendaciones. Este conocimiento puede clasificarse de la siguiente forma:

- *Conocimiento de catálogo:* conocimiento sobre los productos que pueden ser recomendados y sus características. Por ejemplo, un sistema de recomendación de restaurantes podría saber que la cocina Tailandesa es un tipo de cocina asiática.

- *Conocimiento funcional:* el sistema debe ser capaz de establecer relaciones entre las necesidades del usuario y los productos que podrían satisfacer estas necesidades. Por ejemplo, si el usuario necesita un sitio romántico, un restaurante que podría satisfacer esta necesidad sería uno que es silencioso y con una vista marítima.

- *Conocimiento de usuario:* para proporcionar buenas recomendaciones, el sistema debe tener algún conocimiento sobre los usuarios. Éste podría venir de información demográfica o de información específica sobre las necesidades del usuario en ese momento.

Uno de los principales inconvenientes de estos sistemas de recomendación es que la mayoría del conocimiento que utilizan debe ser proporcionado por un experto. La precisión de las recomendaciones dependerá en gran medida de la calidad y la cantidad del conocimiento proporcionado por dicho experto. Además, es muy importante el tipo de herramientas que se le faciliten al usuario para proporcionar sus necesidades. Por un lado, el sistema debe proporcionar al usuario una interfaz fácil y sencilla para que este pueda proporcionar sin ningún problema un ejemplo

cercano a sus necesidades. Debemos tener en cuenta que este conjunto de ejemplos potenciales no debe de ser muy numeroso y sus elementos ser conocidos por el usuario, bien porque los conozca por su propia experiencia, o bien porque sean bien conocidos por todo el mundo. Si el conjunto potencial de ejemplos ofrecidos por el sistema fuera muy numeroso, o bien contuviera muchos ejemplos no conocidos por el usuario, correríamos el riesgo de que éste desistiera de recibir recomendaciones, ya que, éste tendría que emplear mucho tiempo explorando todas las alternativas, hasta encontrar un ejemplo que represente adecuadamente sus necesidades.

Otro de los grandes inconvenientes de este tipo de sistemas de recomendación está relacionado con la fase de refinamiento. Es muy difícil que un usuario encuentre un ejemplo exacto de lo que necesita, por lo que es bastante probable que éste deba refinar su perfil del usuario modificando o añadiendo algunas de las características del ejemplo dado. Las herramientas que se ofrezcan al usuario para proporcionar esta nueva información juegan un papel fundamental en este tipo de sistemas de recomendación. Por un lado, si los productos están descritos con un número excesivo de características puede ocurrir que el usuario se sienta abrumado al tener que revisar dichas características. Si por el contrario sólo se presenta un número reducido de ellas, por ejemplo, las más relevantes, puede ocurrir que se limite en exceso la posibilidades de búsqueda del usuario. Por otro lado, estas características han sido valoradas normalmente por expertos en el tema, y por lo tanto, es de esperar, que estén expresados en dominios precisos y ricos en valoraciones, que un usuario normal puede encontrar difícil de entender y evaluar. Por lo tanto, también encontrará difícil su utilización para modificar o expresar sus necesidades.

A pesar de estas dificultades los sistemas de recomendación basados en conocimiento, han sido utilizados con éxito, por ejemplo en la recomendación de res-

taurantes [30], y presentan muchos beneficios cuando son hibridados con otro tipo de sistemas de recomendación [31], ya que, son capaces de generar recomendación sin necesidad de usar información histórica sobre los usuarios.

En esta memoria, en el capítulo 5, presentamos dos modelos de sistemas de recomendación basados en conocimiento que resuelven los problemas anteriormente mencionados. El primero es un modelo que ofrece un contexto lingüístico a los usuarios para expresar sus opiniones y de esta forma facilitar el proceso de refinamiento del perfil de usuario. El segundo modelo tiene como objetivo optimizar la recogida de información de las necesidades del usuario y la generación de perfiles de usuario.

2.1.4. Técnicas de recomendación: ventajas y desventajas

Todas las técnicas de recomendación tienen sus puntos fuertes y sus debilidades que discutiremos a continuación y que podemos ver de forma resumida en la tabla 2.6. Quizás el problema más conocido es el problema del "incremento" (ramp-up problem) [113]. Este término de hecho se refiere a dos problemas distintos pero que están relacionados:

- *Problema del nuevo usuario:* cuando las recomendaciones se basan solamente en comparaciones entre las valoraciones acumuladas de un usuario objetivo y otros usuarios, si este usuario objetivo tiene pocas valoraciones es bastante difícil clasificarlo de forma correcta.

- *Problema del nuevo producto:* de forma análoga, un nuevo producto que no tiene muchas valoraciones tampoco puede ser fácilmente recomendado. Este problema aparece en dominios de recomendación de nuevos artículos, donde existe un flujo continuo de nuevos productos y cada usuario sólo

Tabla 2.6: Ventajas y desventajas de las técnicas de recomendación

Técnica	Positivo	Negativo
Colaborativa (CF)	A. Puede identificar nichos entre géneros B. No necesita dominio de conocimiento C. Adaptativo: la calidad mejora a lo largo del tiempo D. Feedback implícitos son suficientes (haber comprado un producto,...)	I. Problema del nuevo usuario J. Problema del nuevo producto K. El problema de la oveja "gris" L. Calidad dependiente del tamaño del conjunto de datos históricos M. Problema de la estabilidad vs maleabilidad
Basados en contenido	B,C,D	I,L,M
Demográficos	A,B,C	I,K,L,M N. Debe usar información demográfica
Basados en utilidad	E. No sufre problema del incremento F. Sensible a los cambios de preferencias G. Puede incluir características no relacionadas con los productos	O. El usuario debe introducir una función de utilidad P. Habilidad estática de sugerencia (no aprende)
Basados en conocimiento	E,F,G H. Puede establecer relaciones de las necesidades de los usuarios con los productos	P Q. Requiere adquisición del conocimiento

valora unos pocos. Este problema también es conocido como el problema del "valorador prematuro", porque la primera persona que valora un producto apenas obtiene beneficio por hacerlo (esas valoraciones "prematuras" no mejoran la habilidad con que se compara un usuario con otro [9]). Una de las soluciones que se ha propuesto para resolver este problema es incentivar a los usuarios de alguna forma para que proporcionen más valoraciones.

Los sistemas de recomendación colaborativos dependen de las coincidencias entre las valoraciones de los distintos usuarios y tienen dificultades en encontrar estas coincidencias si el espacio de valoraciones es poco denso (sólo unos pocos usuarios han valorado los mismos productos). El problema de la "densidad" se reduce de alguna forma cuando empleamos enfoques basados en modelos. La densidad sigue siendo un problema importante en dominios tales como el filtrado de noticias, ya que existen muchos productos disponibles, y a no ser que, la base de datos sea muy grande, es bastante difícil obtener otro usuario que comparta un gran número de productos valorado.

Estos problemas sugieren que las técnicas puramente colaborativas, funcionan mejor en problemas donde la densidad de usuarios es relativamente alta en comparación con el universo de productos, el cual debería ser pequeño y estático. Si el conjunto de productos cambia rápidamente, las valoraciones antiguas apenas sirven para los nuevos usuarios, que no han sido capaces de tener valoraciones en comparación con aquellos usuarios existentes. Si el conjunto de productos es muy grande y los usuarios apenas valoran productos, entonces las probabilidades de tener valoraciones comunes con otros usuarios es muy pequeña.

Los sistemas de recomendación colaborativos trabajan mejor para un usuario que encaja en un grupo de usuarios numeroso de gustos similares. La técnica no

trabaja bien cuando existen usuarios "ovejas grises" [40], que están entre las fronteras de dos grupos de usuarios. Este problema también se presenta en los sistemas de recomendación demográficos que intentan clasificar a los usuarios basándose en sus características personales. Por otro lado, los sistemas de recomendación demográficos no sufren del problema del "nuevo usuario", porque no requieren una lista de valoraciones de un usuario. En vez de esto, tienen el problema de que deben recoger información demográfica. Con el incremento de la sensibilidad de los datos *on-line*, especialmente en los contextos de comercio electrónico [43], es bastante difícil que los sistemas de recomendación demográficos tengan éxito, ya que, la mayoría de la información necesaria para realizar las recomendaciones es información que el usuario no suele estar dispuesto a revelar.

Los sistemas de recomendación basados en contenido tienen el problema de que deben acumular suficiente información sobre los gustos y necesidades de los usuarios para realizar recomendaciones fiables. Además, están limitados a las características que explícitamente están asociadas con los productos que pueden recomendar. Por ejemplo, un sistema de recomendación basado en contenido de películas, sólo puede basar sus recomendaciones en la descripción escrita de las películas: nombres de los actores, argumento... Debido a que no puede obtener ninguna información de la película en si, esto hace que estás técnicas basen sus recomendaciones únicamente en los datos descriptivos disponibles. Los sistemas de recomendación colaborativos sólo confían en las valoraciones que pueden ser utilizadas para recomendar productos sin tener en cuenta los datos descriptivos. Incluso en la presencia de datos descriptivos, se ha demostrado que los sistemas de recomendación colaborativos pueden ser más precisos que los basados en contenido [5].

La gran ventaja de los sistemas de recomendación colaborativos sobre los basados en contenido está relacionado con la habilidad de "cruzar géneros" (cross-genre recommendations) o "salir de la caja" (outside the box). Por ejemplo, puede ocurrir que un aficionado a la música que le guste el Jazz, también le guste la música clásica, pero un sistema de recomendación basado en contenido, entrenado en su afición en jazz, sería incapaz de sugerir productos de música clásica si ninguna de las características (interpretes, instrumentos o repertorio) coincide con los productos de las otra categoría, cosa bastante habitual. Sólo si se mira fuera de las preferencias del individuo se podrían realizar tales recomendaciones.

Ambas técnicas de recomendación sufren del efecto "cartera" (portfolio). Un sistema de recomendación ideal no sugeriría algo que el usuario ya conoce o una película que ya ha visto. El problema se complica en dominios tales como el filtrado de noticias, donde es muy fácil encontrarse la misma noticia repetida varias veces, pero de fuentes distintas, pero donde también es muy fácil encontrarse la misma noticia presentando nuevos hechos o perspectivas que, podrían ser de interés para el usuario y éstas no serían recomendadas por el sistema. Por ejemplo, el sistema DailyLearner [19] utiliza un umbral superior de similitud en su recomendador basado en contenido para dejar fuera aquellas noticias demasiado similares a aquellas que el usuario ya ha leído.

Los sistemas de recomendación basados en conocimiento no tienen problemas de incremento o densidad, pues no basan sus recomendaciones en datos acumulados. Sin embargo, presenta un inconveniente como todos los sistemas basados en conocimiento: necesitan adquirir el conocimiento.

A pesar de este inconveniente, los sistemas de recomendación basados en conocimiento tienen algunas características beneficiosas. Son adecuados para demandas puntuales, pues requiere menos información del usuario que otros sistemas de

recomendación. No tienen ningún problema al comienzo del funcionamiento del sistema, en donde las sugerencias son de baja de calidad porque el sistema acaba de arrancar y no tienen ninguna información de preferencia de ningún usuario. Como inconveniente, podría citarse que no es capaz de descubrir grupos de usuarios tal y como hacen los sistemas de recomendación colaborativos. Por otro lado, puede hacer recomendaciones tan variadas como su base de conocimiento lo permita.

Los sistemas de recomendación colaborativos y demográficos tienen la capacidad única de poder establecer relaciones entre géneros y poder recomendar productos no cercanos a las preferencias habituales de los usuarios. Los sistemas de recomendación basados en conocimiento podrían hacer lo mismo, sólo si tales asociaciones han sido identificadas a priori por un experto.

Todas estas técnicas basadas en el aprendizaje (colaborativas, basadas en contenido y demográficas) sufren del problema del comienzo de una forma u otra. Una vez que el perfil de usuario se ha establecido en el sistema, es difícil que se cambien estas preferencias. Por ejemplo, si a alguien le gustaba comer carne y se hace vegetariano, continuará recibiendo recomendaciones de comida no vegetariana por algún tiempo, hasta que las nuevas valoraciones hayan cambiado las preferencias almacenadas. Muchos sistemas de recomendación solventan estos problemas incluyendo algún sistema de envejecimiento de valoraciones, para que las más antiguas tengan menos influencias, pero se corre el riesgo de perder información de preferencia antigua pero valiosa, que represente de forma precisa las preferencias del usuario.

2.1.5. Técnicas de hibridación

Dado que las técnicas de recomendación anteriormente presentadas tienen algunas limitaciones, se planteó como solución a dichos problemas: la hibridación de técnicas. En ella, combinamos dos o más técnicas de recomendación para obtener un mejor rendimiento que utilizando cada una de ellas de forma independiente. Lo más común, es combinar las técnicas de filtrado colaborativo con alguna otra técnica con el objetivo de evitar el problema del "incremento". En la tabla 2.7 podemos ver algunas de las combinaciones más frecuentemente utilizadas [31].

En esta sección estudiaremos estas técnicas y terminaremos señalando qué conclusiones se pueden obtener sobre la hibridación

Modelos de hibridación

A continuación explicaremos cada una de estas técnicas más detalladamente:

1. *Mediante pesos:* los sistemas de recomendación híbridos combinan las recomendaciones de cada sistema, dándole un peso a cada una de las recomendaciones dependiendo de que sistema las ha generado. La importancia o puntuación de cada producto se calcula a partir de los resultados obtenidos por todas las técnicas de recomendación presentes en el sistema.

   Por ejemplo, el sistema de recomendación híbrido mediante pesos más simple sería aquel que calcula la puntuación final mediante una combinación lineal. El sistema P-Tango [40] es un ejemplo de este tipo de hibridación. Inicialmente tanto la técnica colaborativa como la basada en contenido tienen el mismo peso (la misma importancia), pero gradualmente va ajustando los pesos de acuerdo con las opiniones de los usuarios sobre si están conformes o no con los resultados mostrados. Este modelo de combinación no

Tabla 2.7: Métodos de hibridación

Método de hibridación	Descripción
Mediante pesos	Los valoraciones (o votos) de varias técnicas de recomendación son combinadas para producir una única recomendación
Conmutación	El sistema utiliza una u otra técnica dependiendo de la situación actual
Mezcla	Las recomendaciones provenientes de varias técnicas de recomendación son mostradas al mismo tiempo
Combinación de características	Las características de las fuentes de datos de varias técnicas de recomendación son combinadas en un algoritmo de recomendación único
Cascada	Uno de los sistemas de recomendación refina los resultados dado por otro
Aumento de características	La salida de una técnica es utilizada como una entrada de característica de otro
Meta-nivel	El modelo aprendido por un sistema de recomendación es utilizado como entrada de otro

utiliza valoraciones numéricas, sino que trata las salidas de cada técnica de recomendación (colaborativa, basada en contenido o demográfica) como un conjunto de votos, los cuales son combinados mediante un esquema de consenso [155].

Las ventajas de este tipo de sistema de recomendación, es que todas las capacidades del sistema son utilizadas en el proceso de recomendación de una forma sencilla y directa, por lo que es muy fácil ajustar los pesos manualmente o mediante algoritmos simples. Sin embargo, esta técnica parte de la hipótesis de que ese valor de importancia relativa entre las distintas técnicas es más o menos uniforme en todo el espacio de productos. Como hemos visto anteriormente, ésto no siempre es verdad pues, por ejemplo, los sistemas de recomendación colaborativos tienen un peor comportamiento en aquellos productos que tienen pocas valoraciones.

2. *Conmutación:* estos sistemas híbridos utilizan distintos criterios para alternar la técnica de recomendación que deben emplear en cada momento.

   Por ejemplo, el Daily Leaner [19], es un sistema de recomendación híbrido basado en contenido y colaborativo. Este sistema emplea primero métodos de recomendación basados en contenido y si no puede obtener ninguna recomendación con suficiente confianza, hace uso del método colaborativo. Este modelo no evita el problema del incremento ya que ambos métodos, el colaborativo y el basado en contenido, presentan el problema del nuevo usuario, y son utilizados de forma independiente.

   En estos sistemas, la técnica colaborativa proporciona la posibilidad de realizar búsquedas a través de géneros, por lo que podemos obtener recomendaciones de productos, relevantes, pero que no son cercanos en semántica

con aquellos productos que han sido evaluados por el usuario.

Este método de hibridación conlleva una complejidad adicional en el proceso de recomendación pues tenemos que determinar el criterio por el cual cambiaremos de una a otra alternativa, lo cual introduce otro nivel de parametrización. Por otro lado, si este criterio es seleccionado de forma adecuada puede aprovechar de mejor forma las cualidades de los sistemas de recomendación que lo componen y evita las debilidades de otros sistemas en esas situaciones.

3. *Mezcla:* este tipo de hibridación mezcla los resultados provenientes de varias técnicas de recomendación mostrándolas al mismo tiempo. Puede ser utilizado en situaciones donde se requieran un gran número de recomendaciones y de esta forma, presentar al mismo tiempo las recomendaciones de todas las técnicas implicadas.

   El sistema PTV [186] usa este enfoque para recomendar un programa de televisión. Utiliza técnicas basadas en contenido para tratar la información textual de los programas de televisión e información colaborativa para aprovechar la información de preferencia de otros usuarios. Las recomendaciones de ambas técnicas se combinan en una sugerencia final.

   Estas hibridaciones evitan el problema del "nuevo producto" debido principalmente a que, el componente basado en contenido basa sus recomendaciones en nuevos programas utilizando la descripción de éstos, incluso si no han sido evaluados por nadie todavía. Sin embargo, si presenta el problema del "nuevo usuario", ya que, ambos sistemas de recomendación necesitan algunos datos sobre las preferencias del usuario.

4. *Combinación de características:* otra forma de hibridar los sistemas de reco-

mendación basado en contenido y colaborativos, es manejar la información colaborativa como una característica de datos asociada con cada ejemplo, y utilizar técnicas basadas en contenido sobre este conjunto de datos ampliado.

Por ejemplo, en [14] emplean un algoritmo de aprendizaje de reglas para recomendar películas, no sólo utilizando las características contextuales, sino también las valoraciones de los usuarios. Aunque este modelo presenta mejoras sustanciales con respecto al enfoque puramente colaborativo, esto sólo ocurre cuando se tratan manualmente las características contextuales. Los autores de este modelo se dieron cuenta que empleando todas las características contextuales aumentaban el número de productos que se recomendaban, pero no la precisión de las recomendaciones.

Este tipo de hibridación permite a los sistemas usar datos colaborativos sin depender exclusivamente de ellos, reduciendo de esta forma, la sensibilidad del sistema al número de usuarios que han valorado un producto. En cambio, permite al sistema obtener información sobre la similitud inherente que no es posible encontrar en sistemas de recomendación basados en contenido.

5. *Cascada:* este método combina las técnicas mediante procesos compuestos de varias fases, en donde, en la primera fase, hace una primera selección de productos que deben ser recomendados mediante una de las técnicas, y en las fases siguientes, se utilizan las otras técnicas para filtrar este conjunto de candidatos.

   En [31] se presenta un modelo de recomendación de restaurantes, EntreeC, que combina técnicas colaborativas y basadas en conocimiento. EntreeC usa el conocimiento que tiene sobre los restaurantes para hacer recomendaciones

según los intereses declarados por el usuario. Las recomendaciones son colocadas en conjuntos de igual preferencia y se utiliza la técnica colaborativa para seleccionar uno de los conjuntos, y adicionalmente, para ordenar las recomendaciones de dicho conjunto.

La técnica en cascada evita que la segunda técnica tenga que tratar con productos que ya han sido desechados por la primera o que tienen suficientes valoraciones negativas como para que nunca sean recomendados. Gracias a esto, las fases posteriores a la inicial son más eficientes, ya que, sólo tienen que tratar con un conjunto reducido de candidatos y no con todo el conjunto de productos. Además, está técnica es tolerante al ruido en las recomendaciones, en el sentido, de que los productos recomendados por las fases superiores sólo pueden ser refinados y no puede ocurrir, por ejemplo, que un producto no valorado positivamente en una fase anterior sea introducido en una fase posterior.

6. *Aumento de características:* esta técnica es empleada para producir una valoración o una clasificación de un producto, y esa información, es incorporada en el proceso utilizado en la siguiente técnica de recomendación.

   Por ejemplo, el sistema Libra [145] hace recomendaciones basadas en contenido de libros utilizando los datos encontrados en Amazon.com. Usa una red bayesiana para clasificar los distintos campos de texto presentes en la descripción. En los datos textuales se incluyen los "autores relacionados" y los "títulos relacionados" que Amazon genera usando su sistema de recomendación colaborativo. Estas características se han demostrado que mejoran los resultados de las recomendaciones.

   Esta técnica es interesante debido principalmente a que permite mejorar

el rendimiento de la técnica principal empleada en la recomendación, sin necesidad de que dicha técnica se vea alterada en su funcionamiento ni concepción. Hay que observar que esta técnica es diferente de la combinación de características donde los datos eran combinados de forma directa.

También esta técnica es distinta de la combinación en cascada. Aunque ambas técnicas usan los algoritmos de recomendación uno detrás de otro.

7. *Meta-nivel:* otra forma de combinar varias técnicas de recomendación es utilizando el modelo generado por uno como entrada para otro. Esto difiere del aumento de características en el sentido que, en el aumento de características, utiliza el modelo aprendido para generar características como entradas para el segundo algoritmo. Aquí, la entrada es el propio modelo.

   El primer híbrido meta-nivel fue el sistema de filtrado de páginas web Fab [11, 12]. En Fab, un agente de selección específico para el usuario realiza un filtrado basado en contenido utilizando el método de Rocchio [170] para mantener un modelo de vector de términos que describa las áreas de interés del usuario. Una colección de agentes, que recogen nuevas páginas de la web, usa los modelos de todos los usuarios en sus operaciones de recogida. De esta forma, los documentos que son primeramente recogidos son aquellos interesantes para la comunidad al completo y entonces es distribuido a los usuarios particulares. Además de la forma de compartir los modelos de usuario, Fab también realiza una colección de recomendación en cascada colaborativas y basadas en contenido, aunque el modelo colaborativo sólo genera un conjunto de elementos y la información sobre su ranking no es utilizada en la selección de componentes.

   El beneficio de este tipo de hibridación, especialmente cuando estamos hi-

bridando basados en contenido y colaborativos, es que el modelo aprendido es una representación comprimida de los intereses del usuario y el mecanismo colaborativo opera sobre esta representación condensada de información más fácilmente que sobre datos sin tratar.

En el capítulo 6 presentamos el software REJA, el cual tiene una hibridación de varios sistemas de recomendación. En este caso la hibridación implementada es la de conmutación, ya que el sistema alternará entre los distintos sistemas de recomendación dependiendo de que información se tenga sobre el usuario y sus necesidades.

Conclusiones sobre la hibridación

La hibridación puede aliviar algunos de los problemas asociados con el filtrado colaborativo y otras técnicas de recomendación.

Los sistemas de recomendación híbridos que utilizan algoritmos de recomendación basados en contenido y colaborativos, independientemente de la técnica de hibridación que empleen, siempre presentarán el problema del "incremento" pues ambas técnicas necesitan un base de datos de valoraciones.

De todas maneras, esta hibridación es bastante utilizada, porque en muchas situaciones tales valoraciones ya existen, o porque presentan más flexibilidad y mejores resultados que si emplearan estas técnicas de forma independiente.

Las meta técnicas evitan el problema de la densidad de datos comprimiendo las valoraciones sobre muchos ejemplos en un modelo, el cual, puede ser fácilmente utilizado para compararlo entre los usuarios.

Las técnicas basadas en conocimiento y utilidad parecen ser buenas candidatas para la hibridación, ya que no presentan ninguno de los problemas relacionados

Tabla 2.8: Sistemas de recomendación híbridos posibles y actuales (CF=Colaborativo, CN= Basado en contenido, DM= Demográfico, KB= Basado en conocimiento)

	Mediante pesos	Mezcla	Conmut.	Comb. de caracterísitcas	Cascada	Aumento de características	Meta-nivel
CF/CN	P-Tango	PTV, ProfBuilder	DailyLeaner	[14]	Fab	Libra	
CF/DM	[155]						
CF/KB	[198]		[199]				
CN/CF							Fab,[41], LaboUr
CN/DM	[155]			[41]			
CN/KB							
DM/CF							
DM/CN							
DM/KB							
KB/CF					EntreeC	GroupLens	
KB/CN							
KB/DM							

con el incremento de nuevos usuarios o de productos.

La tabla 2.8 resume algunas de las aportaciones más importantes en sistemas de recomendación híbridos. Para simplificar la tabla, se mostrarán juntos los basados en conocimiento y en utilidad ya que los basados en utilidad son un caso especial de los basados en conocimiento.

Debemos destacar que cuatro de las técnicas presentadas obtienen los mismos resultados independientemente del orden en el que se utilicen los algoritmos de recomendación: pesos, mezcla, conmutación y combinación de características. Sin embargo la hibridación mediante cascada, aumento y meta-nivel son dependientes del orden.

2.2. Sistemas de recomendación comerciales

Aquí vamos a revisar el funcionamiento de varios sistemas de recomendación que podemos encontrarnos en Internet. En primer lugar, revisaremos en profundidad Filmaffinity debido a que es uno de los sistemas de recomendación más conocidos y que más éxito ha tenido. Y posteriormente, veremos más brevemente otros sistemas de recomendación comerciales que podemos encontrarnos en Internet (Amazon.com, Drugstore.com,...).

El objetivo de esta revisión es aclarar o facilitar el entendimiento de las técnicas presentadas en este capítulo. Además, estos sistemas de recomendación han aportado algunas ideas para nuestras propuestas y la implementación que hemos realizado en el capítulo 6.

2.2.1. Filmaffinity

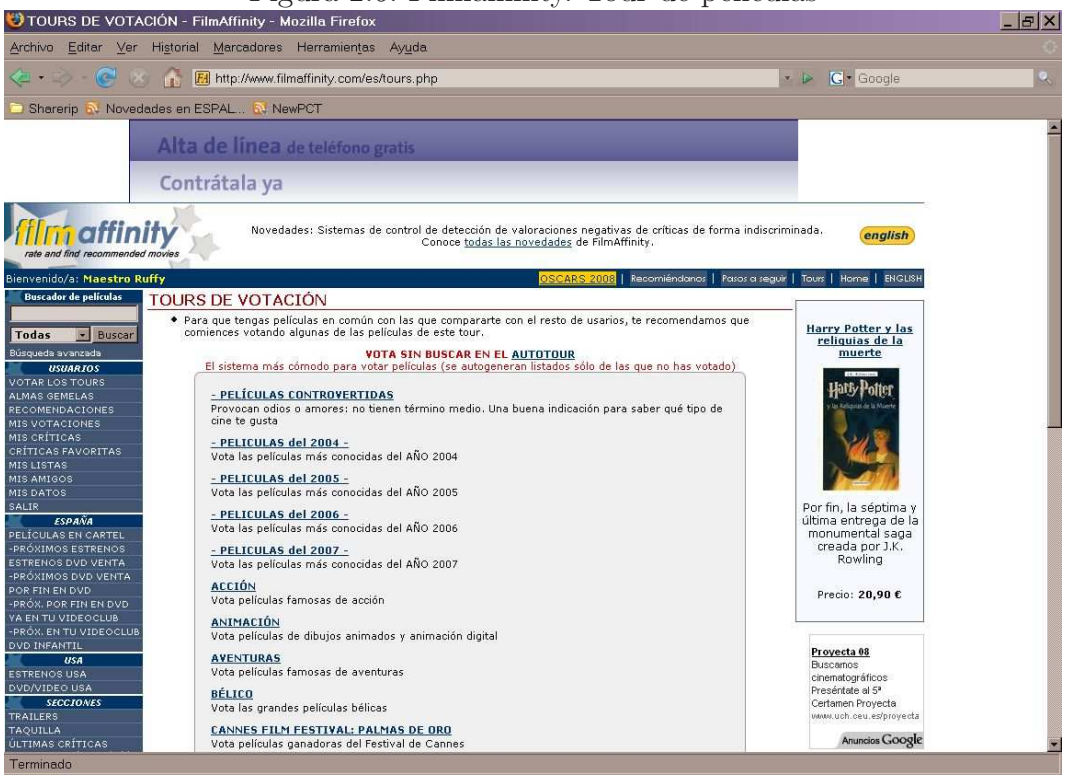
Filmaffinity (http://www.filmaffinity.com) es una página web dedicada al cine (ver figura 2.5) creada en 2002 por el crítico Pablo Kurt y el programador Daniel Nicolás. Actualmente es una de las bases de datos más importantes en español y es uno de los sistemas de recomendación colaborativos más importantes y utilizados en la actualidad. Permite obtener información sobre las películas como: su nombre, el director, actores, nacionalidad, argumento... Pero quizás la cualidad más importante, y que ha supuesto su éxito en Internet, es que proporciona información sobre que puntuaciones han obtenido de los usuarios, sus críticas, y sobre todo, que es capaz de generar recomendaciones de otras películas que le podrían gustar a un usuario en concreto.

Figura 2.5: Filmaffinity. Página principal

Filmaffinity implementa un sistema de recomendación colaborativo. Este sistema de recomendación buscará usuarios con gustos similares a un usuario dado, lo que Filmaffinity denomina almas gemelas, y agregará las puntuaciones de estas almas gemelas para generar las recomendaciones para dicho usuario.

Filmaffinity utiliza escalas numéricas para que los usuarios proporcionen una evaluación sobre las películas. Los usuarios evaluarán las películas, independientemente de su grado de conocimiento, de si son cinéfilos o usuarios casuales, mediante una escala numéricas del 1 al 10 donde 1 significa “Muy mala” y 10 “Excelente”.

Figura 2.6: Filmaffinity. Tour de películas



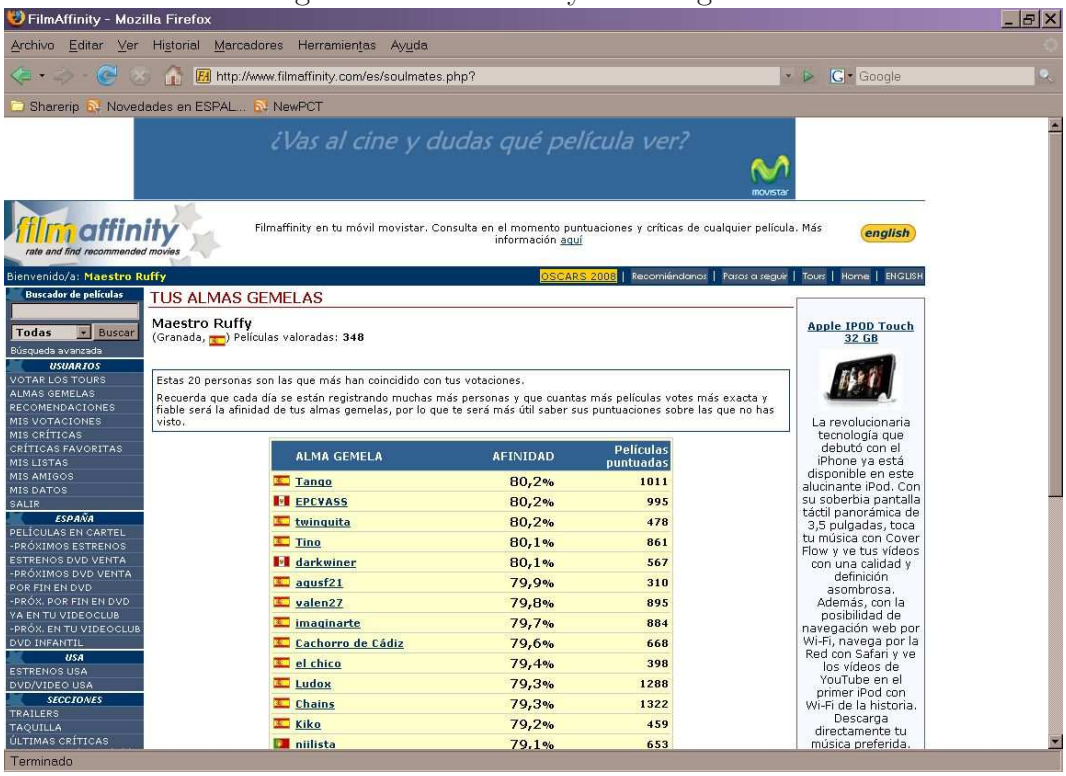
Este sistema de recomendación presenta los mismos inconvenientes de los sistemas de recomendación colaborativos y que vimos en la sección 2.1.4. Es muy difícil que recomiende un producto nuevo que tenga pocas valoraciones, y los nuevos usuarios no recibirán buenas recomendaciones hasta que no hayan valorado un conjunto suficiente de películas.

Para facilitar esto último, Filmaffinity ofrece a los usuarios Tours de votaciones de películas (ver figura 2.6) donde podrán valorar un conjunto de películas clasificadas por años, géneros... De esta forma los nuevos usuarios fácilmente pueden proporcionas las valoraciones necesarias para que Filmaffinity pueda empezar a generar recomendaciones.

Otra de las opciones que ofrece Filmaffinity es la búsqueda de "Almas gemelas" (ver figura 2.7), personas que coinciden en gran medida con nuestros gustos

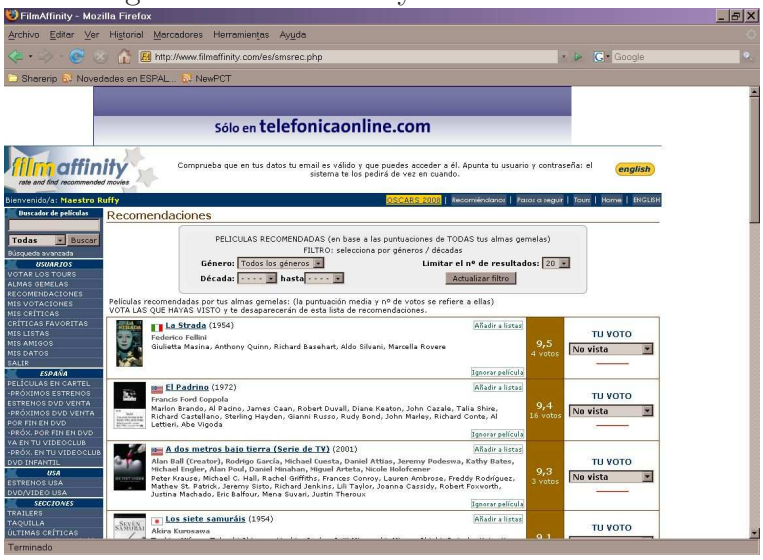
y opiniones en las películas comunes. Con esta opción el usuario puede buscar películas que él no ha visto, pero que si han visto otras personas que tienen gustos similares a él.

Figura 2.7: Filmaffinity. Almas gemelas



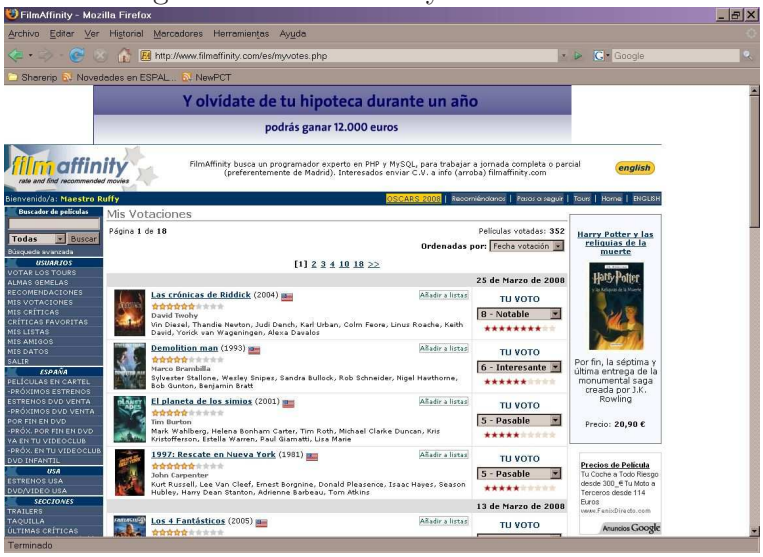
Quizás, la opción más interesante y potente de Filmaffinity sea la generación de recomendaciones para un usuario en particular. A partir de las puntuaciones dadas por los otros usuarios, teniendo en cuenta la afinidad de estos en las valoraciones de las películas con el usuario dado, el sistema es capaz de generar un conjunto de recomendaciones (ver figura 2.8). Esta opción permite seleccionar cuantas recomendaciones quiero recibir, por defecto 20, el rango de años de las películas recomendadas y el género de éstas.

Figura 2.8: Filmaffinity. Recomendaciones



La última opción que ofrece Filmaffinity relacionada con el sistema de recomendación es la posibilidad de alterar las puntuaciones proporcionadas por el usuario (ver figura 2.9), bien porque éste se equivocara al introducirla o porque haya cambiado de opinión. Esta opción, de hecho, lo que muestra es el perfil de usuario actual y que utiliza para encontrar otros usuarios similares a él.

Figura 2.9: Filmaffinity. Puntuaciones



2.2.2. Otros

A continuación veremos brevemente otras páginas web comerciales que emplean sistemas de recomendación para ofrecer servicios personalizados a sus clientes.

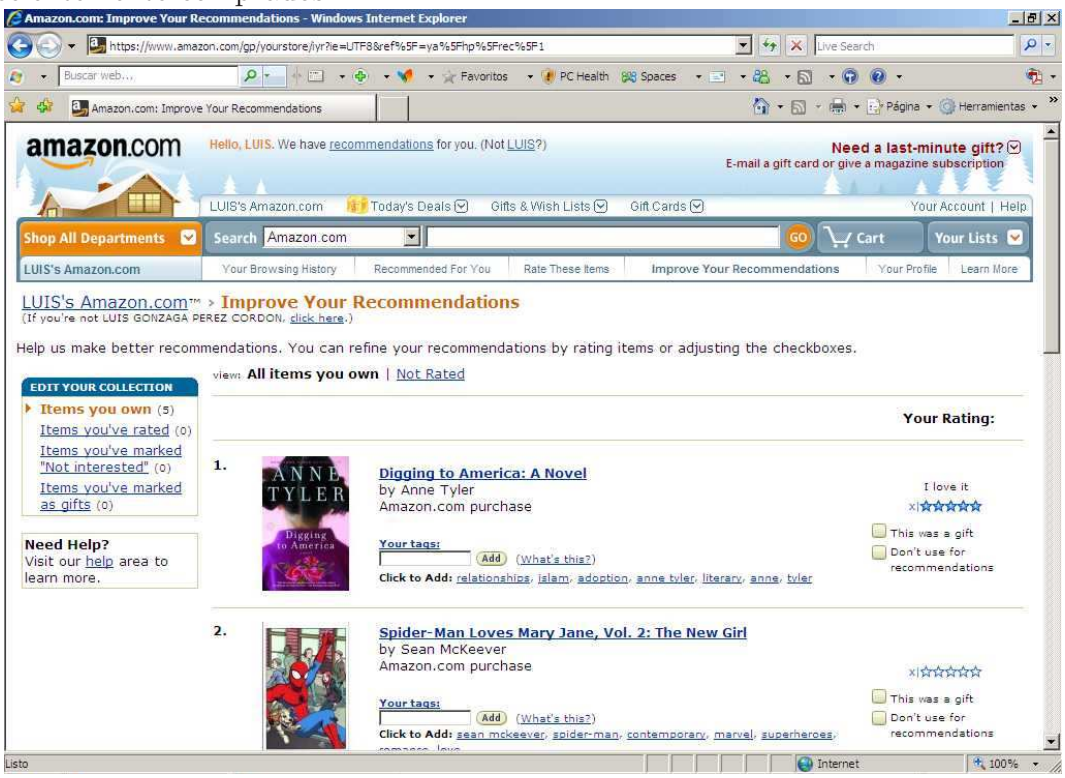
Amazon.com

Amazon.com Inc., es una compañia de comercio electrónico con sede en Seattle, estado de Washington, y fue una de las primeras grandes compañías en vender bienes a través de Internet. En la actualidad vende una gran variedad de productos, desde libros hasta ropa, pasando por artículos electrónicos, cds de música, software, muebles, comida, etc. Amazon se ha establecido en varios países como Canadá, Reino Unido, Alemania, Austria, Francia, China y Japón para poder ofrecer los productos en estos países.

Amazon fue fundada como Cadabra.com por Jeff Bezos en 1994 y lanzada en 1995. Cadabra.com comenzó como una librería online que tenía más de 200.000 títulos que se podían comprar por correo electrónico. Tiempo después, su nombre fue cambiado a Amazon, por el río sudamericano del mismo nombre.

Amazon implementa una gran variedad de sistemas de recomendación, tanto clásicos, como híbridos. Por ejemplo, parte de las recomendaciones que generan parecen estar hechas por un sistema de recomendación basado en contenido, que utiliza las descripciones de los productos valorados positivamente por el usuario para encontrar otros nuevos. Para valorar estos productos usa una escala de 1 al 5 de estrellas donde el uno es la valoración mas baja y significa que no le ha gustado y el 5 la más alta.

Figura 2.10: Página de valoraciones donde el cliente puntúa los productos más recientemente comprados.



En esta sección nos centraremos únicamente en los servicios personalizados que ofrece Amazon en la sección de Libros (ver figura 2.10), los cuales están basados en sistemas de recomendación:

Clientes que compraron (Customers Who Bought): como otros muchos sistemas de recomendación comerciales, Amazon.com (www.amazon.com) estructura la información en una página donde podemos encontrar información sobre un libro en concreto, dando detalles del texto o información de compra. La característica *Clientes que compraron* se encuentra en la página de información de cada libro del catálogo. De hecho, muestra dos listas de recomendación diferentes.

Tus recomendaciones: Amazon también anima a que los clientes aporten sus opiniones sobre los libros que han leído. Los clientes valoran dichos libros

usando una escala de 5 puntos que representa valoraciones desde "lo odio" hasta la valoración máxima "me encanta". Después de valorar una muestra de libros, los clientes pueden pedir recomendaciones de libros que le podrían gustar.

Observador (eyes): esta característica permite que los clientes reciban notificaciones por correo electrónico de nuevos productos que han aparecido en el catálogo de Amazon.

Entregas de Amazon.com: es una variación de la característica anterior. Los clientes seleccionan una serie de géneros o categorías de una lista de ellas (libros de cocina, biografías,...) y periódicamente los editores de Amazon.com les envían sus últimas recomendaciones vía e-mail para los subscriptores de dicha categoría.

Ideas para regalos de la librería: la característica Ideas para regalos permite a los clientes recibir recomendaciones de los editores. Los clientes seleccionan una categoría de libros de la que le gustaría recibir sugerencias. Mientras navega por esta sección de regalos, se puede ver una lista general de recomendaciones creada por los editores de Amazon.com. Esta característica es, en muchos aspectos, una versión *on-line* de la característica *Entregas* de Amazon.com explicada anteriormente. La principal diferencia es que aquí, los clientes reciben las recomendaciones anónimamente ya que, no tienen ninguna necesidad de registrase en Amazon como si ocurre en Entregas de Amazon.com.

Comentarios de los clientes: esta característica permite a los clientes recibir recomendaciones basadas en las opiniones de otros clientes. Estos comentarios aparecen en la página de información de los productos con una valoración de 1 a 5 estrellas y con comentarios escritos proporcionados por otros clientes que han leído el libro en cuestión y dado una crítica. Los clientes tienen la posibilidad de incorporar estas recomendaciones en sus decisiones de compra. Además, los clien-

tes pueden valorar estos comentarios. En cada comentario podemos encontrarnos la pregunta "*¿Te ha ayudado este comentario?*" en donde los clientes pueden decir si o no. Los resultados son tabulados y se informa de la utilidad de ese comentario diciendo "*5 de cada 7 personas encontraron esta crítica útil*".

Círculos de compras: el círculo de compras permite a los clientes ver la lista de los 10 más destacados dada una región geográfica, compañía, institución educativa, gobierno u otra organización. Por ejemplo, un cliente podría pedir ver que libros son los más vendidos para los clientes que trabajan en Oracle, MIT, o los residentes de Nueva York.

CDNOW

CDNOW es una de las primeras tiendas de comercio electrónico en tener éxito. Fue fundada en febrero de 1994 por los hermanos gemelos Jason y Matthew Olim. Inicialmente fue lanzado como un servicio Telnet en agosto de 1994, y en septiembre de 1994 se lanzó la página web de CDNOW.

CDNOW fue una de las compañías pioneras en el uso del marketing por Internet, ofreciendo a sus clientes entre otras cosas recomendaciones, video *on-line*, marketing por correo electrónico...

En la actualidad CDNOW es una tienda dedicada a la venta de CDs por Internet y utiliza los sistemas de recomendación para aumentar dichas ventas y lograr una mayor fidelidad de los clientes. Por ejemplo, al igual que ocurría con Amazon, parte de las recomendaciones que generan parecen estar hechas por un sistema de recomendación basado en contenido, que utiliza las descripciones de los productos valorados positivamente por el usuario para encontrar otros nuevos. Para valorar estos productos usa una escala de 1 al 5 de estrellas donde el uno es la valoración mas baja y significa que no le ha gustado y el 5 la más alta.

Figura 2.11: Página de CDNOW

CDNOW ofrece los siguientes servicios para proporcionar recomendaciones a sus clientes:

Consejero de álbumes: este consejero de CDNOW trabaja de tres modos distintos. Los dos primeros son similares al *Clientes que compraron* que vimos en Amazon.com. Los clientes van explorando y examinando distintos álbumes o artistas. El sistema al mismo tiempo, va recomendando diez álbumes distintos relacionados con el álbum o artista en cuestión. Las recomendaciones se muestran con frases como "*Los clientes que compraron X también compraron el conjunto S*" o "*Los clientes que compran el producto creado por Y también compraron el conjunto T*". El tercer modo de trabajo es un consejero de regalos. Los clientes escriben los nombres de hasta tres artistas y el sistema devuelve una lista de 10 álbumes que CDNOW considera similares a los artistas en cuestión.

Artistas relacionados: esta característica de CDNOW basa su funciona-

miento en la idea de que si a un cliente le gusta un intérprete en concreto, existe un grupo de artistas con estilo similar que también le puede gustar. Los clientes localizan a un artista y selecciona los enlaces de artistas relacionados. Al hacerlo, se proporciona una lista de estos artistas que son considerados artistas similares y una lista de artistas que pertenecen a sus orígenes o influyen en el artista seleccionado.

Guías del comprador: la característica guía del comprador permite a los clientes recibir recomendaciones relacionadas con un tipo específico de música. Los clientes navegan en una lista de géneros y seleccionan uno de estos enlaces de esta lista llevando a los clientes a una nueva lista de álbumes que los editores consideran una parte fundamental de este género.

El artista escoge: en esta opción se presentan recomendaciones proporcionados por los artistas. Cada semana un artista diferente proporciona una lista de álbumes que representen sus gustos o que artistas está escuchando en la actualidad.

Los 100 más importantes: tradicionalmente los productos más vendidos o los más promocionados son los más utilizados para hacer recomendaciones a sus clientes. Después de todo, si un álbum esta en los 10 más vendidos, entonces debe de ser un álbum bueno. Esta característica permite a los clientes de CDNOW recibir este tipo de recomendaciones, estos 100 se obtienen de las ventas realizadas en el sitio web y teóricamente son actualizadas para reflejar las ventas actuales.

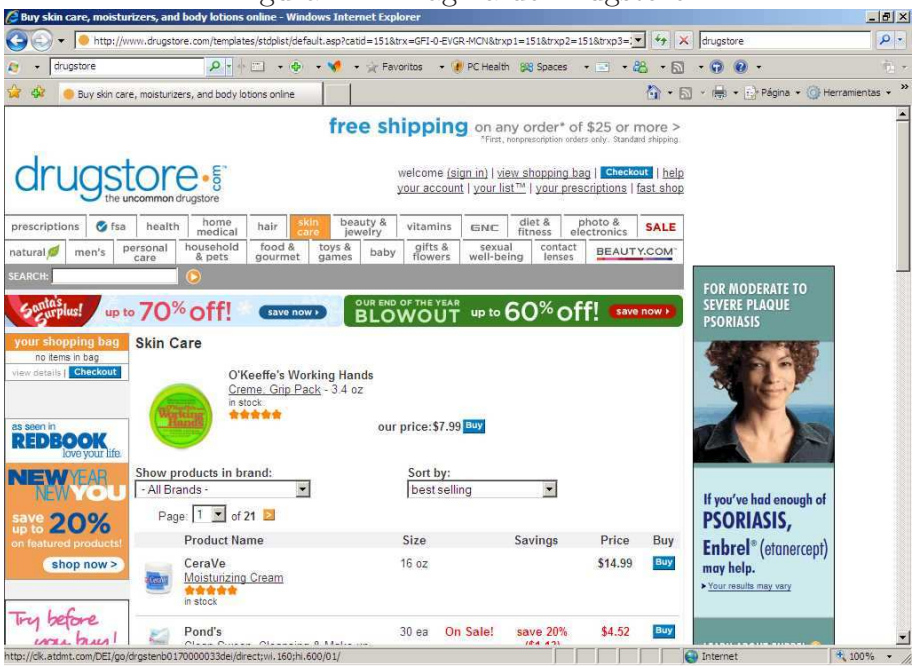
Mi CDNOW: mi CDNOW permite a los clientes crear su propia tienda de música, basándose en los álbumes y los artistas que le gustan. Los clientes indican qué álbumes poseen y qué artistas son sus favoritos. Las compras desde CDNOW se añaden automáticamente en la lista álbumes que poseo y estas compras, por defecto, se supone que le han gustado a los clientes. Posteriormente estos clientes

pueden modificar esa suposición e indicar que productos que posee le gustan y cuales no. Cuando los clientes solicitan las recomendaciones, el sistema predice seis álbumes que al cliente le podría gustar basándose en aquellos que ya posee. Se produce una retroalimentación cuando el cliente proporciona información sobre si ya posee este producto, si quiere añadirlo a los productos que desearía comprar o si por el contrario ese producto no es de su agrado. En la figura 2.11 podemos ver un ejemplo de funcionamiento de CDNOW.

Drugstore.com

Drugstore.com es una farmacia *on-line* cuya sede central está en Bellevue, Washington. Empezó a funcionar el 24 de febrero de 1999. Está asociada con otras compañías como la cadena de tiendas Raid Store, y la General Nutrition Center (GNC). Para vender productos de dichas cadenas.

Figura 2.12: Página de Drugstore



Entre los productos que podemos adquirir podemos encontrarnos desde cremas de protección solar, hasta remedios para la gripe o catarro pasando por productos del cuidado del cabello, etc.

Dado el funcionamiento del servicio *Consejero* que explicaremos a continuación se intuye que trabaja con un sistema de recomendación basado en utilidad. A continuación presentaremos detalladamente este servicio y otros que ofrece drugstore.com para ayudar a sus clientes mediante recomendaciones:

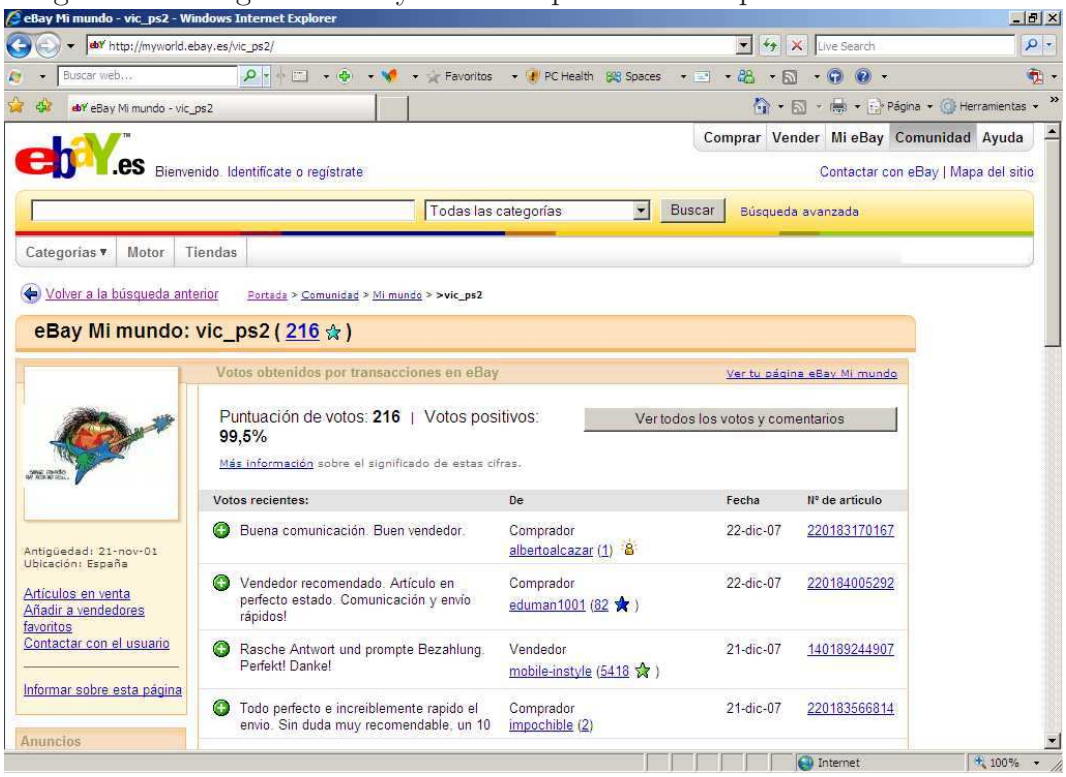
Consejero: el consejero en Drugstore.com permite a los clientes declarar sus referencias cuando desean comprar un producto en algunas de las características tales como "protección solar" o "remedios contra la gripe y resfriados". Por ejemplo, en este último caso, los clientes indican los síntomas que desean aliviar (nariz que moquea o estornudos), la forma en que quieren que le proporcionen dichos productos y la edad del paciente al cual se le va a administrar el producto. Una vez se haya proporcionado todos estos datos el consejero devuelve una lista de productos que cumplen estas condiciones.

Test de pruebas: se le envía un producto a un grupo de voluntarios, obtenido de los clientes del sitio web, para que proporcionen críticas del producto incluida una puntuación basada en estrellas y comentarios (ver figura 2.12).

eBay

eBay Inc., es una compañía de comercio electrónico americana que gestiona el sitio web eBay.com, dedicado a las subastas online y comercio, donde la gente y las compañías compran y venden bienes y servicios por todo el mundo mediante subastas o compra directa. A parte del sitio web situado en Estados Unidos, eBay ha establecido otros sitios webs en otros treinta países.

Figura 2.13: Página de eBay donde se puede ver un perfil de observaciones



El sitio web fue fundado en San José, California el 3 de septiembre de 1995 por Pierre Omidyar con el nombre de AuctionWeb. La compañía cambio oficialmente de nombre en septiembre de 1997 pasando a conocerse como eBay.

En este caso, uno de los sistemas de recomendación implementados en ebay, es uno basado en utilidad, como puede comprobarse si se emplea la opción de *Comprador personal* que presentaremos posteriormente.

A continuación explicaremos, no sólo esta herramienta, sino otras que ofrece esta empresa para ayudar a sus clientes:

Perfil de observaciones: el perfil de observaciones es una de las características de eBay.com que permite tanto a compradores y a vendedores contribuir en el perfil de observaciones de otros clientes con quien ha hecho alguna transacción económica. Estas observaciones consisten en una valoración de satisfacción (sa-

tisfecho/neutral/descontento) y un comentario específico sobre este cliente. Estas observaciones son utilizadas para proporcionar un sistema de recomendación a los compradores, que serán capaces de ver los perfiles de los vendedores. Este perfil consiste en una tabla con el número de valoraciones en los últimos 7 días, mes pasado, seis meses además de un resumen global (ejemplo, 867 valoraciones positivas de 776 clientes distintos). Si el cliente lo desea, puede navegar por las distintas valoraciones individualmente y consultar los comentarios que se han puesto sobre el vendedor (ver figura 2.13).

Comprador personal: el comprador personal permite a los clientes indicar en que productos está interesado en su compra. El cliente introduce un plazo en el que se realizar dicha búsqueda y se busca en base a un conjunto de términos, incluyendo información sobre los límites de precio que desea pagar. Cada cierto tiempo, con uno, dos o tres días de intervalo, el sitio web realiza las búsqueda del cliente en todas las subastas y devuelve un correo con los resultados de esta búsqueda.

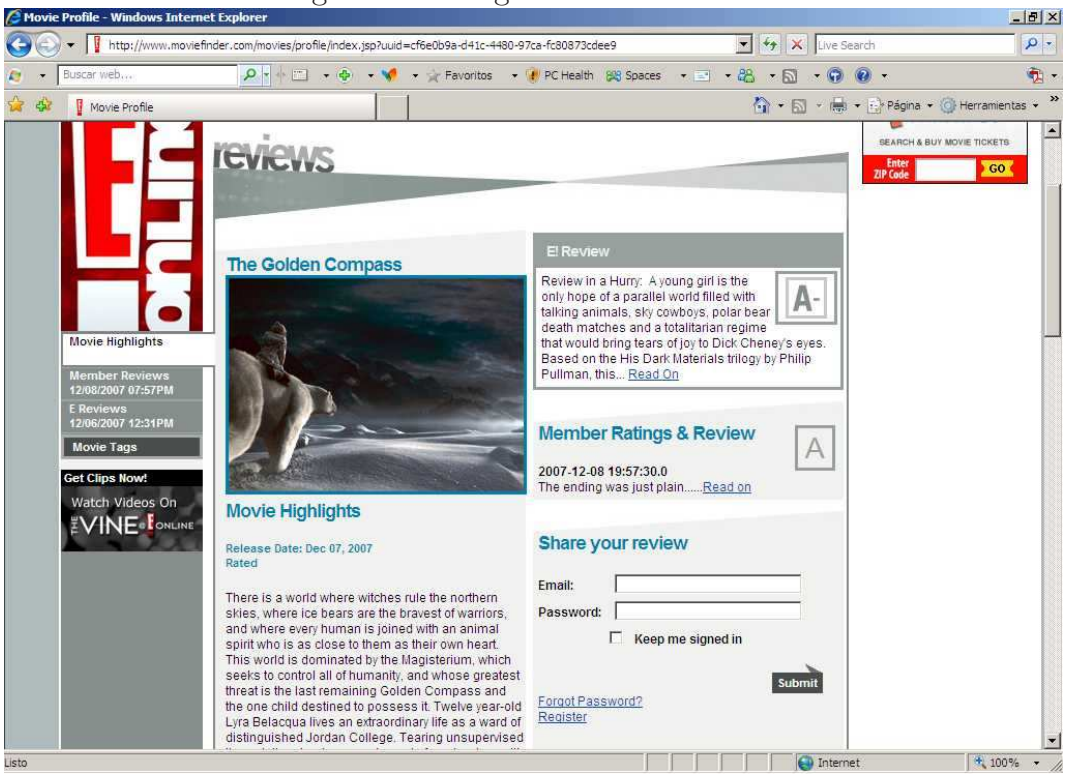
MovieFinder.com

MovieFinder.com es un sitio web de películas mantenido por E! Entertainment Television (ver figura 2.14) que permite obtener información sobre una película, actores, actrices, o por ejemplo, ver las últimas noticias relacionadas con el mundo del cine.

E! es una compañía de televisión americana que opera tanto por cable como vía satélite. Esta compañía fue creada por, entre otros, Larry Namer, Alan Mruvka, Brian Owens y Rick Portin el 31 de julio de 1987 y se denominó Movietime. En principio solo ofrecía servicios de bajo coste como la emisión de tráiler de películas, noticias de actualidad, eventos y entrevistas. Tres años después, en junio de 1990,

cambio su nombre a E! Entertainment Television.

Figura 2.14: Página de MovieFinder



Aunque propiamente dicho MovieFinder.com no tiene implementado ningún sistema de recomendación, si tiene herramientas básicas para producir recomendaciones o servicios personalizados a sus clientes. En este caso, la escala que utilizan los usuarios, los clientes y los editores, para valorar las películas es una escala con letras (A-F). A partir de esta información MovieFinder.com ofrece los siguientes servicios personalizados a sus clientes:

Calificación usuarios/nuestra calificación: tanto la calificación de los usuarios como nuestra calificación se realizan utilizando una letra (A-F) que representa esta clasificación. Estas evaluaciones son agregadas y este valor agregado se muestra como la calificación de los usuarios. "Nuestra calificación" es una calificación proporcionada por los editores de E! Online. De esta forma, los clientes

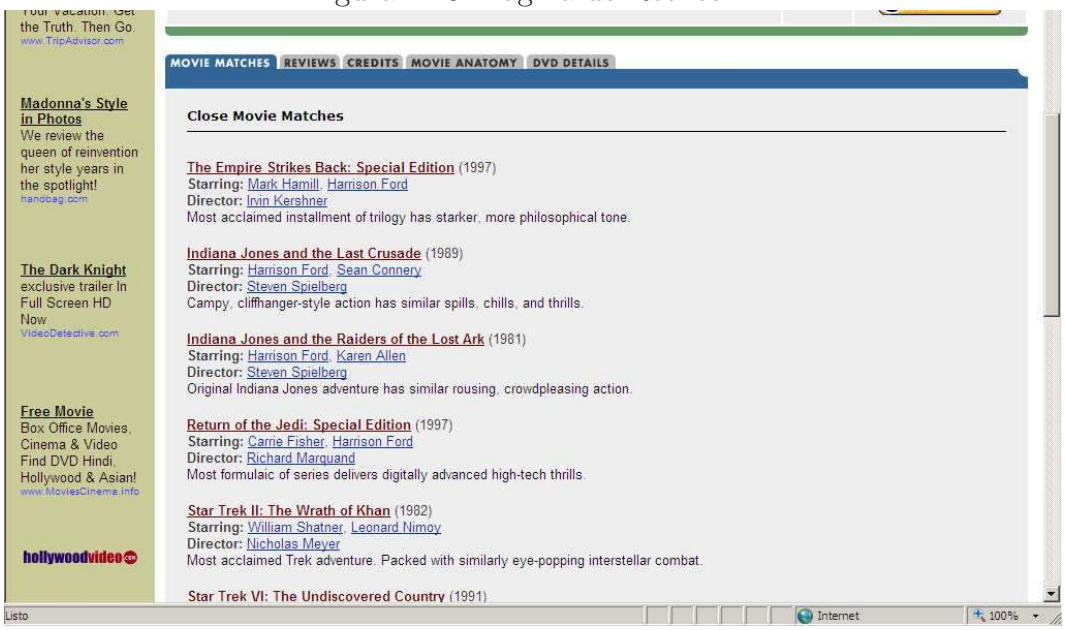
cuando ven las características de una película, ven también la calificación que le han dado tanto los editores como otros usuarios de ese sitio web.

Los 10 mejor valorados: los 10 más importantes permiten a los clientes obtener recomendaciones de los editores en una categoría seleccionada por el usuario. Los clientes seleccionan dicha categoría de una lista de categorías previamente definidas y le devuelve las 10 películas mejor valoradas por los editores de E! Online.

Reel.com

Reel.com (ver figura 2.15) es el sitio web de Reel Video, también conocido como Reel the Store. Reel Video es una tienda de alquiler de películas situado en Berkeley, California, desde 1997. Era la única tienda física real que estaba detrás del sitio web Reel.com, que fue fundada con la intención de convertir la tienda en una cadena de tiendas nacional.

Figura 2.15: Página de Reel.com



Reel.com fue adquirida por Hollywood Video en 1998 y la parte comercial de este sitio web fue clausurada en el 2000. Aunque la tienda real sigue alquilando películas, actualmente Reel.com es otro sistema de recomendación de películas que me permite consultar información relativa a películas actuales, que están en cartelera, clásicas...

Reel.com también permite que sus usuarios reciban recomendaciones pero no ofrece ninguna información sobre qué técnica de recomendación está empleando. En principio puede ser una basada en conocimiento o una variante de uno basado en contenido. Gracias a este sistema de recomendación Reel.com ofrece el siguiente servicio a sus clientes:

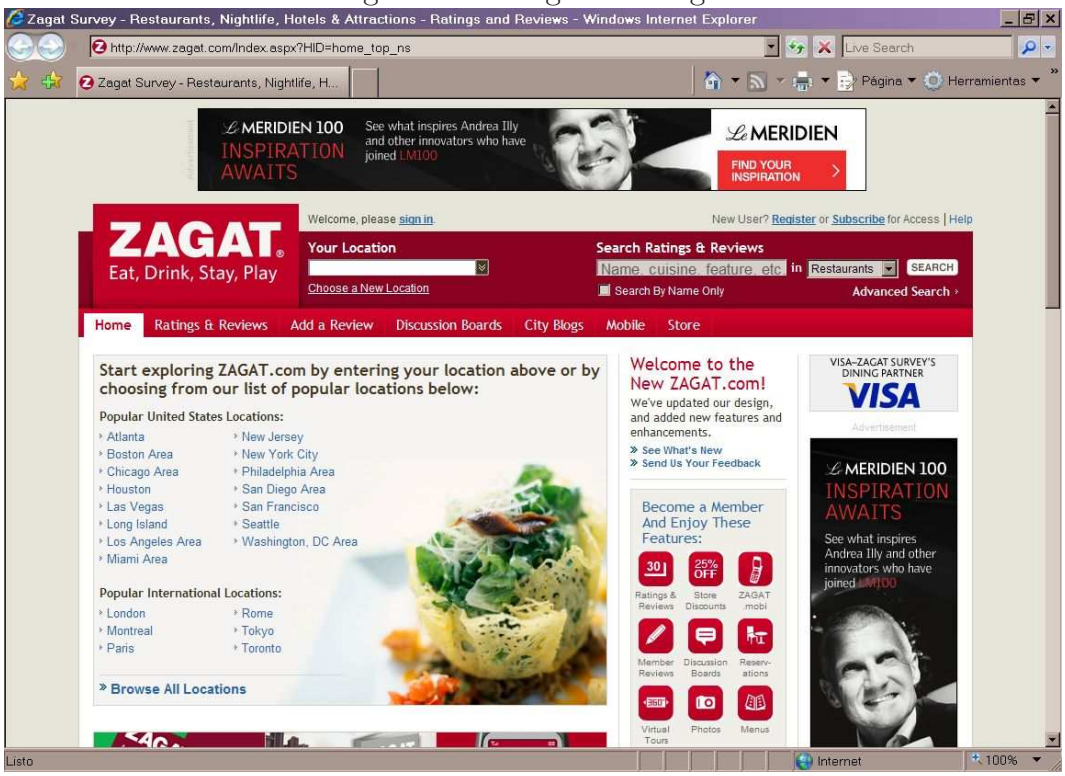
Combinación de películas: tiene un funcionamiento muy similar al que vimos en Amazon.com en *Clientes que compraron* y proporciona las recomendaciones en la misma página donde se muestra la información sobre cada película. Estas recomendaciones consisten en combinaciones cercanas o combinaciones creativas. Cada conjunto contiene una docena de enlaces a las páginas de información de cada una de estas películas. Estos enlaces están descritos con una frase que indica lo similar que esta película a la película original en cuestión.

Zagat.com

Zagat Survey es una empresa americana fundada en 1979 por Tim y Nina Zagat que se dedica a la edición de todo tipo de guías de restaurantes, hoteles, clubes o tiendas de distintas ciudades de los Estados Unidos y Canadá. En zagat.com los usuarios registrados pueden votar distintos aspectos (hasta 30) del local preferido y, además, introducir pequeños comentarios con su experiencia. En base a estas votaciones los responsables de la empresa asignan su puntuación en sus guías anuales y hacen recomendaciones individuales a sus usuarios a través de su web.

Estas recomendaciones son hechas por un sistema de recomendación colaborativo.

Figura 2.16: Página de Zagat



Zagat (ver figura 2.16) es la página web comercial que utiliza un sistema de recomendación colaborativo, donde las valoraciones de los restaurantes son actualizadas constantemente por los usuarios. Incluye muchas opciones de búsquedas avanzadas donde el usuario puede especificar sus criterios de búsqueda en categorías tales como comida, decoración, coste, etc., o restringir la búsqueda a cierta área o un tipo de cocina concreta. Este sistema de recomendación tiene el mismo objetivo que el desarrollado en esta memoria en la capítulo 6. Entre otras opciones incluye:

Valoraciones y críticas de restaurantes: permite explorar los restaurantes de una ciudad en concreto, de Estados Unidos o del mundo. Una vez seleccionado la zona y el restaurante, podemos leer las críticas hechas por otros usuarios o ver

qué valoraciones tienen en comida, decoración, servicio y coste.

Valoraciones y críticas de ocio nocturno: similar al anterior pero especializado en sitios que se pueden visitar para el ocio nocturno.

Valoraciones y críticas de hoteles: nos proporciona información detallada sobre un hotel determinado, junto con la puntuación que ha recibido y críticas de otros usuarios.

Valoraciones y críticas de atracciones: similar al anterior pero centrado en atracciones turísticas.

Capítulo 3

Modelado de la información en procesos de recomendación

Como hemos visto en el capítulo anterior, los sistemas de recomendación, utilizan información acerca de sus usuarios para generar recomendaciones personalizadas. Dicha información está relacionada con los gustos, necesidades y preferencias de los usuario. Esta información suele ser vaga o imprecisa, ya que, la mayoría de las veces se están expresando, o bien aspectos cualitativos de los objetos que han comprado, utilizado o visto, o bien cualidades de los objetos que desean adquirir, ver o utilizar. Sin embargo, tal y como hemos visto en el capítulo anterior, los sistemas de recomendación obligan a los usuarios a expresar esta información mediante valoraciones numéricas precisas. En la literatura se han afrontado situaciones similares a ésta y se han desarrollado técnicas y modelos que permiten representar, o modelar, dicha información desde otras perspectivas más adecuadas tales como, el enfoque lingüístico difuso [218].

Este enfoque se ha utilizado con éxito sobre un gran número de problemas tales como la toma de decisiones [23, 46, 60, 78, 125, 129, 158, 172, 193, 200, 212], selección de materiales [38], administración de personal [80], recuperación de información [24], economía [8], diagnosis clínica [44], gestión comercial [214],

psicología [35, 42, 105], evaluación [130, 131, 132, 138, 139], o planificación y secuenciación [1, 107].

A la hora de modelar las preferencias de los usuarios, además de elegir el dominio de la información, también hay que decidir la estructura con la que se representará esa información.

En el caso de los sistemas de recomendación es habitual el uso de vectores de utilidad y de valoraciones numéricas. El dominio numérico fue el primer dominio utilizado por los sistemas de recomendación, ya que es el dominio utilizado internamente por los sistemas de recomendación clásicos para generar las recomendaciones. Como hemos visto, este dominio no siempre es el más adecuado si tenemos en cuenta que el usuario está expresando sus preferencias, valoraciones o percepciones subjetivas sobre productos. Un usuario difícilmente puede expresar información precisa, pues lo habitual es que un usuario sólo tenga una idea más o menos clara sobre lo que quiere comprar.

Por lo tanto, para estas situaciones, parece más acertado el uso de valoraciones lingüísticas más cercanas al modo en el que se expresan los usuarios y así modelar de forma más adecuada la incertidumbre y vaguedad de dicha información.

Dado que uno de nuestros objetivos es aumentar la calidad del modelado de la información recogida para aumentar la calidad de las recomendaciones, proponemos en esta memoria el uso de información lingüística.

En nuestras distintas propuestas propondremos el uso de distintas estructuras de representación de información así como, el uso de múltiples escalas lingüísticas para ofrecer una mayor flexibilidad de expresión a los usuarios y expertos involucrados en el sistema de recomendación.

A continuación, vamos a hacer una revisión de los conceptos relacionados con el modelado de preferencias que vamos a usar en esta memoria. Primero veremos

distintas estructuras de representación de información que podemos utilizar en el sistema de recomendación, para después revisar el concepto de dominios de expresión, centrándonos en el modelado lingüístico.

En la literatura existen distintas propuestas para modelar la información lingüística [118, 218]. En esta memoria, nosotros propondremos el uso del enfoque lingüístico difuso del que revisaremos los conceptos necesarios para entender nuestras propuestas, y terminaremos revisando un marco de trabajo para manejar múltiples escalas lingüísticas que utilizaremos en algunas de nuestras propuestas.

3.1. Modelado de Preferencias

El modelado de preferencias es una área de investigación de gran importancia en muchos campos de aplicación y actividades tales como, la toma de decisión [50, 172], consenso [87] o los sistemas de recomendación [134, 135]. El modelado de preferencias es el encargado de estudiar la representación más adecuada para representar las preferencias de los expertos en un problema dado. A la hora de decidir como modelar la información debemos estudiar por un lado, la estructura para representar la información , y por otro, el dominio en el que se expresarán las preferencias sobre los elementos del problema. En las siguientes secciones estudiaremos las alternativas más comúnmente utilizadas a la hora de modelar esta información centrando nuestro interés en las estructuras y dominios que utilizaremos en el resto de la memoria.

3.1.1. Estructuras para la representación de preferencias

A la hora de modelar las preferencias, no sólo tenemos que tener en cuenta qué dominio se va a utilizar para representar las valoraciones, sino también

que estructura escogeremos para representar esta información. La elección de la estructura juega un papel fundamental en los procesos de recomendación. Hay estructuras de representación de preferencias más fáciles o más completas de utilizar dependiendo del usuario, del problema a resolver y del tipo de información que deseamos representar [202].

En esta sección hacemos un repaso de las estructuras de información más utilizadas en la literatura para representar las preferencias por parte de los expertos y/o usuarios involucrados en un problema como son [87, 94, 189]:

- Vectores de Utilidad
- Órdenes de Preferencia
- Relaciones de Preferencia

Vectores de Utilidad

En esta estructura almacenamos la utilidad, o la preferencia, de cada una de las alternativas del problema. Esta estructura ha sido una de las más utilizadas en muchos problemas [49, 127, 189] al ser muy fácil de usar, entender y explotar, siendo la más común en sistemas de recomendación. Cada usuario tendrá asociado un vector de utilidad donde almacenará la valoración de cada producto que conozca o haya comprado. En esta estructura, aquellos elementos que tengan una valoración mayor, serán los preferidos por el usuario.

Ejemplo

Sea $E = \{e_1, ..., e_m\}$ $(m \geq 2)$ un conjunto de usuarios y $X = \{x_1, x_2, ..., x_n\}$ $(n \geq 2)$ conjunto de productos que se van a valorar. La preferencia dada por el usuario, e_j, sobre el conjunto de elementos X se expresará de la siguiente forma utilizando vectores de utilidades:

$$U^j = \left\{u_1^j, ..., u_n^j\right\}$$

donde u_i^j representa la utilidad o valoración dada por el usuario j al producto i.

Órdenes de preferencia

En esta estructura lo que representamos es el orden de preferencia de los elementos para un determinado problema [94, 184, 187]. Esta estructura, aunque es más fácil de utilizar que la anterior presenta el inconveniente que, no hay forma de conocer el grado con que un elemento es preferido sobre otro, sólo conocemos el orden de preferencia. Para un individuo, e_j, un orden de preferencia se representará mediante un vector ordenado decreciente del conjunto de elementos $O^j = \{o^j(1), ..., o^j(n)\}$ de forma que cuanto menor sea la posición de una alternativa, más preferida es para dicho individuo. En sistemas de recomendación esta estructura no es muy utilizada ya que, es bastante difícil de utilizar si el número de elementos es grande, lo que es muy habitual en sistemas de recomendación. Además aporta menos información que los vectores de utilidad y que las relaciones de preferencia que veremos a continuación.

Ejemplo

Sea $E = \{e_1, ..., e_m\}$ $(m \geq 2)$ un conjunto de usuarios y $X = \{x_1, x_2, ..., x_n\}$ $(n = 4)$ conjunto de elementos o productos que se van a valorar. Las preferencias dadas por los individuos 1 y 2 sobre el conjunto de elementos X utilizando órdenes de preferencia podrían ser las siguientes:

$$O^1 = \{x_3, x_2, x_1, x_4\}$$
$$O^2 = \{x_2, x_3, x_1, x_4\}$$

En este ejemplo, el individuo 1 considera que el elemento que más prefiere es el elemento es x_3 y el menos preferido el x_4. Sin embargo, para el individuo 2 el más preferido es x_2 y el menos también es el x_4.

Relaciones de Preferencia

El tercer tipo de estructura que revisamos son las relaciones de preferencia. Este tipo de estructuras han sido muy utilizadas en problemas de toma de decisión [34, 56, 65, 79, 86, 103, 104, 144, 206] y se utilizan para representar relaciones binarias entre pares de elementos $x_l R x_k$ x_l con $x_k \in X$. Una relación binaria mide la intensidad o el grado con que un elemento x_l es preferido sobre otro elemento x_k.

La principal ventaja de este tipo de estructura frente a las anteriores es que para las personas es más fácil expresar su preferencia sobre un par de elementos, que dar un vector utilidad o un orden sobre todos los elementos en cuestión. Además, la información aportada por una relación de preferencia suele ser mayor y proporciona una mayor flexibilidad a la hora de expresar la información de preferencia [202], sin embargo, también pueden inducir la aparición de inconsistencias.

En condiciones ideales, una persona no debería tener ningún problema en expresar sus preferencias sobre todos los pares de elementos de un conjunto de elementos. Además, estas preferencias deberían ser consistentes para que las soluciones obtenidas a partir de esta información fueran siempre satisfactorias. Bajo estas suposiciones se han hechos estudios para determinar que propiedades deben de cumplir las relaciones de preferencias, de forma que se pueda garantizar que las soluciones obtenidas son satisfactorias.

Independientemente del dominio en el que estén expresadas estas preferencias, se define la consistencia en términos de transitividad. A continuación vamos a

establecer algunas propiedades de las relaciones de preferencia que nos ayudarán a estudiar su consistencia. Enumeramos estas propiedades, tal y como fueron definidas inicialmente, en dominios numéricos definidos en el intervalo $[0, 1]$ donde el valor 0.5 representa la indiferencia:

1. *Condición triangular* [127]*:* $p_{ij} + p_{jk} \geqslant p_{ik}$

   Si interpretamos esta condición geométricamente debemos considerar las tres alternativas x_i, x_j, x_k, como los vértices de un triángulo con el tamaño de los lados p_{ij}, p_{jk} y p_{ik}, y por lo tanto, la longitud correspondiente a los vértices x_i, x_k no debería exceder la suma de las longitudes correspondientes a los vértices x_i, x_j y x_j, x_k.

2. *Transitividad débil* [188]*:* $p_{ij} \geqslant 0.5, p_{jk} \geqslant 0.5 \Rightarrow p_{ik} \geqslant 0.5$

   La interpretación de esta condición es la siguiente: si x_i es preferido a x_j y x_j es preferido a x_k, entonces x_i es preferida a x_k. Este tipo de transitividad es la condición usual de transitividad que una persona lógica y consistente debería utilizar si no quiere expresar opiniones inconsistentes, y por lo tanto es la condición requerida mínima que una relación de preferencia debería cumplir.

3. *Transitividad max-min* [51, 224]*:* $p_{ik} \geqslant \min(p_{ij}, p_{jk}) \, \forall i, j, k$

   En este caso, el valor de preferencia obtenido por la comparación directa de dos alternativas debería ser igual o mayor que el mínimo de los valores obtenidos por la comparación de esas dos alternativas con una intermedia. Este tipo de transtividad ha sido el requisito tradicional para caracterizar la consistencia en el caso de relaciones de preferencia numéricas $[0, 1]$ [224]. Sin embargo, a la hora de verdad, en situaciones prácticas, nos podemos encontrar con relaciones de preferencia que podrían ser consideradas como

perfectamente consistentes y que no cumplen esta transitividad.

4. *Transitividad max-max* [51, 224]*:* $p_{ij} \geqslant \text{máx}\,(p_{ij}, p_{jk})\,\forall i, j, k$

   Este concepto representa la idea de que el valor de preferencia obtenido por una comparación directa de dos alternativas debería ser igual o mayor que el máximo valor obtenido de comparar esas dos alternativas con una intermedia. Esta transitividad es más fuerte que la anterior, y por tanto las relaciones que no verificaban la anterior tampoco verifican ésta.

5. *Transitividad max-min restrictiva* [188]*:* $p_{ij} \geqslant 0.5, p_{jk} \geqslant 0.5 \Rightarrow p_{ik} \geqslant \text{mín}\,(p_{ij}, p_{jk})\,\forall i, j, k$

   Cuando una relación de preferencia verifica esta condición entonces una alternativa cualquiera x_i es preferida a x_j con un valor p_{ij} y x_j es preferido a x_k con un valor p_{jk}, entonces x_i debería ser preferido a x_k con, al menos, una intensidad de preferencia p_{ik} igual al mínimo de los valores anteriores. Esta transitividad es más fuerte que la transitividad débil pero es menos restrictiva que la transtividad max-min.

6. *Transitividad max-max restrictiva* [188]*:* $p_{ij} \geqslant 0.5, p_{jk} \geqslant 0.5 \Rightarrow p_{ik} \geqslant \text{máx}\,(p_{ij}, p_{jk})\,\forall i, j, k$

   Cuando una alternativa x_i es preferida a x_j con un valor p_{ij} y x_j es preferida a x_k con un valor p_{jk}, entonces x_i debería ser preferido a x_k con al menos una intensidad de preferencia p_{ik} igual al máximo de los valores anteriores. Este concepto es más restrictivo que la transitividad max-min restrictiva y más suave que la transitividad max-max.

7. *Transitividad multiplicativa* [188]*:* $(p_{ji}/p_{ij}) \cdot (p_{kj}/p_{jk}) = p_{ki}/p_{ik} \forall i, j, k$

   Tanino [188] introdujo este concepto de transitividad en el caso de que $p_{ij} >$

$0\ \forall i, j$ e interpretando p_{ij}/p_{ji} como un ratio de la intensidad de preferencia de x_i a x_j (x_i es p_{ij}/p_{ji} veces tan bueno como x_j). La transitividad multiplicativa incluye la transitividad max-max restrictiva [187, 188] y puede ser reescrita como $p_{ij} \cdot p_{jk} \cdot p_{ki} = p_{ik} \cdot p_{kj} \cdot p_{ji} \forall i, j, k$ para ser extendida a todo el conjunto de relaciones de preferencia numéricas $[0, 1]$.

8. *Transitividad aditiva* [187, 188]*:* $(p_{ij} - 0.5) + (p_{jk} - 0.5) = (p_{ik} - 0.5)\ \forall i, j, k$ o equivalentemente $p_{ij} + p_{jk} + p_{ki} = \frac{3}{2} \forall i, j, k$

   Este tipo de transitividad ha sido muy utilizada en los proceso de rellenado de relaciones de preferencia ya que proporciona un valor único dada dos preferencias, y por lo tanto, su cálculo y su utilización es sencillo. Este tipo de transitividad es más fuerte que la definida con la transitividad max-max restrictiva.

Sin embargo, en muchas situaciones reales no podemos garantizar ni suponer que un usuario, e_j, pueda o quiera proporcionar todas las preferencias sobre los pares de elementos, ni que no existan inconsistencias entre dichas preferencias.

Además, esta representación tiene otra desventaja debido a que la información que debe aportar el usuario, es mayor que con los vectores de utilidad o con los órdenes de preferencia. Con el primero, sólo tienen que aportar n valoraciones, en el segundo, ordenar n elementos, mientras que con esta representación tiene que aportar $n \times n$ valoraciones.

Este tipo de relaciones se representa mediante una matriz $P_{e_j} \subset X \times X$, donde el valor $\mu_{P_{e_j}}(x_l, x_k) = p_j^{lk}$ representa el grado de preferencia de la alternativa x_l sobre la alternativa x_k dada por e_j [121, 189, 217],

$$P_{e_j} = \begin{pmatrix} p_j^{11} & \cdots & p_j^{1n} \\ \vdots & \ddots & \vdots \\ p_j^{n1} & \cdots & p_j^{nn} \end{pmatrix}$$

Hay situaciones en los que no todos los valores de la matriz son conocidos. Puede ocurrir que haya valores desconocidos por las siguientes razones[202]:

- *Desconocimiento:* el usuario puede no conocer las preferencias entre alguno de los elementos.

- *Falta de tiempo:* puede ser que al usuario no se le exija o no tenga el tiempo necesario para proporcionar toda la información.

- *Incomparabilidad:* esto ocurre cuando estamos comparando elementos totalmente distintos que no se pueden comparar (la preferencia de un coche sobre un ratón, por ejemplo).

En el apéndice C revisamos varios algoritmos que pueden reconstruir, en algunos casos, relaciones de preferencia incompletas y que serán útiles en el segundo modelo de recomendación basado en conocimiento que presentamos en el capítulo 5.

En sistemas de recomendación esta representación no ha sido utilizada debido a la dificultad de que un usuario proporcione una relación de preferencia completa sobre todos los productos del sistema de recomendación. Sin embargo, en el capítulo 5, proponemos la utilización de esta representación aplicada sobre un subconjunto reducido de elementos cercanos a las necesidades del usuario para la generación de un perfil de usuario que represente las necesidades de éste.

Ejemplo

Sea $E = \{e_1, ..., e_m\}$ $(m \geq 2)$ un conjunto de usuarios y $X = \{x_1, x_2, ..., x_n\}$ $(n = 4)$ conjunto de elementos que se van a valorar. Si el dominio en el que trabajamos es un dominio numérico $[0, 1]$ tal que:

- $p_i^{lk} = \frac{1}{2}$, significa que hay indiferencia sobre la preferencia entre ambos elementos.
- $p_i^{lk} \geq \frac{1}{2}$, significa que el elemento x_l es preferida sobre el x_k.
- $p_i^{lk} = 1$, significa que el elemento x_l es totalmente preferido sobre la x_k.

Las preferencias dadas por el individuo, e_1, sobre el conjunto de elementos X tendría el siguiente aspecto:

$$P_{e_1} = \begin{pmatrix} - & 0.3 & 0.7 & 0 \\ 0.7 & - & 0.6 & 0.6 \\ 0.3 & 0.4 & - & 0.2 \\ 1 & 0.4 & 0.8 & - \end{pmatrix}$$

3.1.2. Dominios de expresión de preferencias

Una vez se haya establecido cuál es la estructura de representación de preferencias más adecuada para el tipo de problema a resolver, se debe determinar qué dominio utilizarán los individuos para valorar los elementos. Si la elección del dominio ha sido adecuada, los individuos se sentirán más seguros a la hora de expresar sus valoraciones y tendremos una mayor garantía de éxito [50]. Lo habitual es que todos los individuos expresen sus preferencias en un mismo domino de información, es decir en un contexto homogéneo [3, 23, 36, 46, 55, 115, 123, 129, 161, 207]. Sin embargo, existen situaciones y problemas donde es conveniente que los individuos expresen sus preferencias utilizando distintos dominios de información o

dominios con distintas escalas o granularidad, es decir, en un contexto heterogéneo [45, 57, 83, 86, 196, 223].

En la literatura [45, 86, 223] encontramos que la información puede ser expresada en distintos dominios dependiendo de distintos factores, siendo los siguientes dominios los más utilizados por los expertos y/o usuarios:

- *Dominio Numérico:* es el dominio más utilizado en el modelado de preferencias debido principalmente a la facilidad con la que se puede manejar matemáticamente. En la literatura podemos encontrarnos con tres variantes a la hora de valorar información numéricamente:

  - *Numérico Binario:* en el cual los individuos sólo pueden utilizar dos valores para expresar sus alternativas, el 0 para representar una valoración negativa de las alternativas y 1 para representar una valoración positiva.

  - *Numérico normalizado en el intervalo* $[0, 1]$. En este dominio se puede expresar una intensidad o grado de preferencia utilizando un valor numérico perteneciente al intervalo $[0, 1]$ [58, 86, 121]. La principal ventaja de esta representación sobre la anterior es que ahora, al existir una mayor flexibilidad y una mayor expresividad, podemos obtener resultados más precisos.

  - *Numérico en un escala* $(a, \cdots, b)$*:* en sistemas de recomendación, lo habitual es utilizar escalas numéricas con un limite inferior y otro superior, y en donde solo se permiten utilizar valores enteros para expresar la preferencia. Por ejemplo, Filmaffinity almacena las preferencias de sus usuarios mediante un vector de utilidad numérico, donde cada

elemento recibirá una valoración entre entre 1 y 10; Amazon usa la misma representación y dominio pero con una escala entre 1 y 5. En estas escalas el 1 representa la peor valoración y el 5 o el 10 la mejor dependiendo de la escala.

- Dominio Intervalar: este dominio se utiliza en situaciones donde no se puede dar un valor preciso, y se quiere de alguna forma, recoger la imprecisión o vaguedad con la que una persona expresa sus valoraciones o preferencias [115, 191]. Este problema ha hecho que se necesiten definir nuevos modelados de preferencia más flexibles y capaces de recoger esta incertidumbre, siendo el modelado intervalar uno de ellos. En este dominio, las preferencias se expresan por medio de intervalos $[\underline{a}, \overline{a}]$ $(\underline{a} \leq \overline{a})$. De esta forma las personas no se ven forzadas a dar una valoración exacta, sino el intervalo en donde se encuentra dicha preferencia.

  En los sistemas de recomendación este dominio no ha sido utilizado y además presenta un inconveniente adicional. Los usuarios deben aportar dos valores, el límite superior y el límite inferior de la relación. Por lo tanto, deben esforzarse más y dedicar más tiempo por cada valoración. Esto no suele ser aconsejable en este tipo de problemas, ya que, puede hacer que los usuarios desistan de sus búsquedas.

- Dominio Lingüístico: aunque, el enfoque lingüístico es menos preciso que el numérico, proporciona algunas ventajas tales como que las valoraciones lingüísticas, cuando estamos valorando aspectos cualitativos, se entienden mejor que las numéricas además de disminuir los efectos de ruido, debido a que, cuanto más refinada es una valoración, más sensible es al ruido y por lo tanto mayores errores conllevan. Además de ser el dominio utilizado

habitualmente en el razonamiento humano.

En esta memoria, prestaremos una mayor atención al dominio lingüístico, ya que, es en el que se han centrado la mayor parte de nuestras propuestas de modelos de recomendación a la hora de modelar la información proporcionada por los usuarios de un sistema de recomendación. Esta decisión se debe a que consideramos que en este dominio, los usuarios (expertos o clientes), se sienten más cómodos utilizando términos lingüísticos para valorar aspectos relacionados con las percepciones subjetivas de naturaleza cualitativa. Las cuales, en el mundo real, se expresan de forma habitual utilizando palabras del lenguaje natural en lugar de números.

En este capítulo revisaremos la aproximación que utilizaremos en nuestra memoria para modelar este tipo de situaciones, el enfoque lingüístico difuso [218]. Este enfoque ha sido utilizado con éxito, por ejemplo, en problemas de secuenciación [1], toma de decisión en política [6], teoría de la decisión [21, 46, 61, 133, 206], consenso [90, 91, 140], recuperación de información [24, 89], diagnosis clínica [44] o en la evaluación de productos o servicios [62, 77, 122, 132, 136, 138, 137].

El uso de información lingüística implica procesos de computación con palabras, por lo que en primer lugar revisaremos los conceptos básicos del enfoque lingüístico difuso y sus modelos clásicos de computación con palabras. Posteriormente, revisaremos una extensión del enfoque lingüístico difuso basado en la 2-tupla lingüística y su modelo computacional, debido a las ventajas que presentan frente a modelos computacionales clásicos [82]. Por lo que será el modelo computacional que utilizaremos en nuestras propuestas de esta memoria. Finalmente, también revisaremos un marco de representación y un modelo computacional para tratar con múltiples escalas lingüísticas, ya que, algunas de nuestras propuestas están definidas en este tipo de contextos.

3.2. Enfoque lingüístico difuso

Los problemas que nos encontramos en el mundo real presentan aspectos que pueden ser de distinta naturaleza. Cuando dichos aspectos o fenómenos son de naturaleza cuantitativa, éstos se valoran fácilmente utilizando valores numéricos. Sin embargo, cuando se trabaja con información vaga e imprecisa o cuando la naturaleza de tales aspectos no es cuantitativa sino cualitativa, no es sencillo ni adecuado utilizar un modelado numérico, aconsejándose el uso de un modelado lingüístico. Los aspectos cualitativos aparecen frecuentemente en problemas en los que se pretende evaluar fenómenos relacionados con percepciones y relaciones de los seres humanos (diseño, gustos,...). En estos casos se suelen utilizar palabras del lenguaje natural (bonito, feo, dulce, salda, simpático,...) en lugar de valores numéricos para emitir tales valoraciones. Tal y como se indica en [221], el uso de un modelado lingüístico de preferencias puede deberse a varias razones:

1. La información disponible con la que trabajan los expertos es demasiado vaga o imprecisa para ser valorada utilizando valores numéricos precisos.

2. La información puede no ser cuantificable debido a su naturaleza y sólo puede ser descrita mediante términos lingüísticos (ej., al evaluar el "confort" o el "diseño" de un coche [122]).

3. Puede ser que la naturaleza de lo que estemos midiendo permita la utilización de valores numéricos, pero en la práctica esto no sea posible, bien porque no existan los instrumentos necesarios para llevar a cabo estas mediciones, o bien porque el coste de estas medición sea demasiado elevado. En este caso el uso de un "valor aproximado", es decir, una etiqueta lingüística, puede ser adecuado.

El Enfoque Lingüístico Difuso ha demostrado ser una técnica adecuada para modelar este tipo de información [1, 6, 21, 24, 44, 46, 76, 197, 206, 212] debido a que proporciona un método directo para representar la información lingüística mediante variables lingüísticas, que se diferencian de las numéricas en que sus valores son palabras o frases en un lenguaje natural en lugar de números [218].

Una variable lingüística, en el enfoque lingüístico difuso, se caracteriza por un valor sintáctico o etiqueta y por un valor semántico o significado. La etiqueta es una palabra o frase perteneciente a un conjunto de términos lingüísticos y el significado de dicha etiqueta viene dado por un conjunto difuso en un universo del discurso. Esta forma de representar la información permite modelar la incertidumbre de la información. Formalmente una variable lingüística se define como [218]:

Definición 3.1. *Una variable lingüística está caracterizada por una quíntupla* $(H, T(H), \mathbb{U}, G, M)$*, en la que:*

- H *es el nombre de la variable.*
- $T(H)$ *es el conjunto de valores lingüísticos o etiquetas lingüísticas.*
- $\mathbb{U}$ *es el universo de discurso de la variable.*
- G *es una regla sintáctica (que normalmente toma forma de gramática) para generar los valores de* $T(H)$*.*
- M *es una regla semántica que asocia a cada elemento de* $T(H)$ *su significado. Para cada valor* $L \in T(H)$*,* $M(L)$ *será un subconjunto difuso de* $\mathbb{U}$*.*

Para definir el conjunto de valores lingüísticos que tomará una variable lingüística desde punto del vista del Enfoque Lingüístico Difuso, es necesario llevar a cabo

dos operaciones fundamentales:

1. Elección de un conjunto de términos lingüísticos adecuado, $T(H)$.

2. Definición de la semántica asociada a cada término lingüístico

3.2.1. Elección del conjunto de términos lingüísticos

Para que una fuente de información (un experto, un usuario, un consultor, ...) pueda expresar con facilidad sus opiniones y/o sus conocimientos es necesario que disponga de un conjunto adecuado de descriptores lingüísticos. Uno de los aspectos más relevantes a la hora de definir o seleccionar un conjunto de descriptores lingüísticos, es el número de etiquetas lingüísticas disponible para expresar esta información, o lo que lo mismo, la granularidad de la incertidumbre de dicho conjunto de etiquetas [22]. Se dice que un conjunto de términos lingüísticos tiene:

- Una granularidad baja, o un tamaño de grano grueso, cuando la cardinalidad del conjunto de etiquetas lingüísticas es pequeña. Esto significa que el dominio está poco particionado y que existen pocos niveles de distinción de la incertidumbre, pudiéndose producir una pérdida de expresividad, y por lo tanto de información, si no es posible encontrar ninguna etiqueta que represente con precisión la opinión o la valoración de la fuente de información.

- Una granularidad alta, o un tamaño de grano fino, se produce cuando la cardinalidad del conjunto de etiquetas lingüísticas es alta. Esta situación puede provocar un exceso de complejidad en la descripción del dominio, de forma, que las fuentes de información no sean capaces de discernir cuál, de un conjunto posible de etiquetas, representa mejor su opinión o valoración.

Por lo tanto, podemos afirmar que la cardinalidad de un conjunto de términos lingüísticos no debe de ser demasiado pequeña como para imponer una restricción en la precisión de la información que se quiere expresar, pero debe ser lo suficientemente grande, como para permitir hacer una discriminación adecuada de las valoraciones en un número limitado de grados. Normalmente, se prefiere utilizar cardinalidad impar, 7 ó 9, donde el término medio representa una valoración "aproximada de 0.5" y el resto de términos se distribuyen alrededor de él [22]. Estos valores clásicos de cardinalidad están directamente relacionados con el estudio realizado por Miller [142] sobre la capacidad humana, en el que se indica que se pueden manejar razonablemente y recordar alrededor de siete o nueve términos diferentes.

Una vez establecida la cardinalidad, se necesita un mecanismo para generar los términos lingüísticos. Existen dos enfoques diferenciados para esta tarea, uno los define a partir de una gramática libre de contexto, y el otro mediante un orden total definido sobre el conjunto de términos. A continuación revisaremos brevemente ambos mecanismos:

1. *El enfoque basado en una gramática libre de contexto:* una posibilidad para generar el conjunto de términos lingüísticos consiste en utilizar una gramática libre de contexto G, donde el conjunto de términos pertenece al lenguaje generado por G [21, 24, 220]. Una gramática generadora, G, es una 4-tupla (V_N, V_T, I, P), donde V_N es el conjunto de símbolos no terminales, V_T el conjunto de símbolos terminales, I el símbolo inicial y P el conjunto de reglas de producción. La elección de estos cuatro elementos determinará la cardinalidad y forma del conjunto de términos lingüísticos. El lenguaje generado por esta gramática deberá ser lo suficientemente grande como para que pueda

describir cualquier posible situación del problema, pero no infinito o muy grande pues dificultaría su comprensión y utilización.

2. *El enfoque basado en términos primarios con una estructura ordenada.* Una alternativa para reducir la complejidad de definir una gramática consiste en dar directamente un conjunto de términos distribuidos sobre una escala con un orden total definido [23, 78, 211, 212]. Por ejemplo, consideremos el siguiente conjunto de siete etiquetas:

$$T(H) = \{N, MB, B, M, A, MA, P\}$$

$s_0 = N = Nada$ $s_1 = MB = Muy\,bajo$ $s_2 = B = Bajo$

$s_3 = M = Medio$ $s_4 = A = Alto$ $s_5 = MA = Muy\,alto$

$s_6 = P = Perfecto$

donde $s_i < s_j$ si y sólo si $i < j$.

Normalmente en estos casos es necesario que los términos lingüísticos satisfagan las siguientes condiciones adicionales:

a) Que exista un operador de negación. Por ejemplo, $Neg(s_i) = s_j$, $j = g - i$ ($g + 1$ es la cardinalidad de $T(H)$).

b) Tiene que haber un operador de maximización: máx$(s_i, s_j) = s_i$ si $s_i \geq s_j$.

c) Y un operador de minimización: mín$(s_i, s_j) = s_i$ si $s_i \leq s_j$.

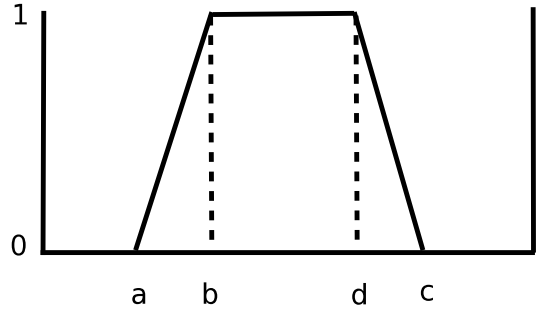
3.2.2. Semántica del conjunto de términos lingüísticos

En la literatura existen varias alternativas para definir la semántica del conjunto de etiquetas lingüísticas [23, 195, 196, 218], siendo la alternativa más común

el uso de funciones de pertenencia [22, 24, 46, 120, 193]. De esta forma, para definir la semántica del conjunto de términos lingüísticos, se utilizan números difusos en el intervalo $[0, 1]$, donde cada número difuso está definido por una función de pertenencia. En el apéndice A, hacemos una pequeña revisión de los conceptos básicos de la Teoría de Conjuntos Difusos que utilizaremos en esta sección. Para mayor detalle, véase [108].

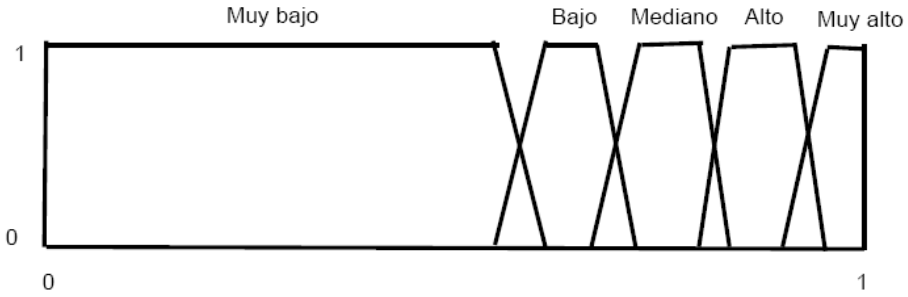
Un método eficiente, desde un punto de vista computacional, para caracterizar un número difuso es el uso de una representación basada en parámetros de la función de pertenencia [21]. Algunos autores consideran que las funciones de pertenencia trapezoidales son lo suficientemente buenas como para modelar la imprecisión de la información que se desea representar [21, 22, 46, 193, 194]. Esta representación paramétrica se expresa usando una 4-tupla (a, b, d, c) que se muestra en la figura 3.1. Los parámetros "b" y "d" indican el intervalo en el que la función de pertenencia vale 1; mientras que "a" y "c" indican los extremos izquierdo y derecho de la función de pertenencia [21].

Figura 3.1: Función trapezoidal



En la figura 3.2 se muestra la semántica de una variable lingüística que evalúa la altura de una persona utilizando números difusos definidos por funciones de pertenencia trapezoidales:

Figura 3.2: Función altura



$$T\,(Altura) = \{Muy\,bajo, Bajo, Mediano, Alto, Muy\,alto\}$$

$$Muy\,bajo = (0, 0, 0.6, 0.62)$$

$$Bajo = (0.6, 0.62, 0.67, 0.7)$$

$$Mediano = (0.67, 0.7, 0.8, 0.82)$$

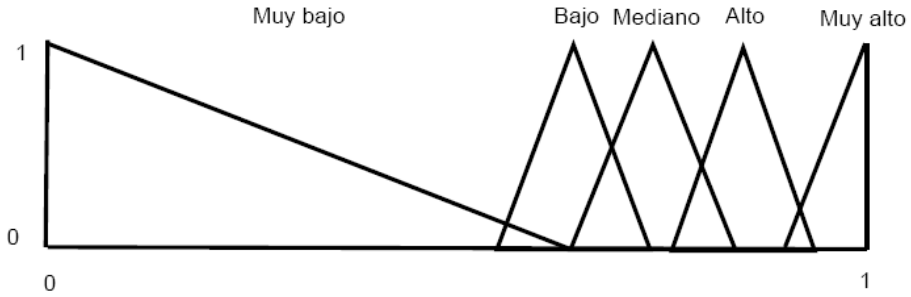
$$Alto = (0.8, 0.82, 0.92, 0.95)$$

$$Muy\,Alto = (0.92, 0.95, 1, 1)$$

Un caso particular de este tipo de representación son las funciones de pertenencia triangulares, en las que $b = d$. Se representan mediante una 3-tupla (a, b, c), donde "b" es el valor donde la función de pertenencia vale 1, mientras que "a" y "c" indican los extremos izquierdo y derecho de la función.

La figura 3.3 muestra el mismo conjunto visto anteriormente, pero representado ahora con funciones de pertenencia triangulares.

Figura 3.3: Función altura con función de pertenencia triangular



Otros autores prefieren utilizar otro tipo de funciones como por ejemplo las Gaussianas [24].

3.3. Modelos de computación con palabras

El uso de información lingüística implica procesos de computación con palabras tales como la fusión, agregación y comparación de valoraciones lingüísticas. Para realizar estos cálculos el enfoque lingüístico difuso ha utilizado difentes modelos de computación:

- *El modelo computacional lingüístico basado en el principio de extensión*: cuando la semántica de las etiquetas está definida mediante conjuntos difusos se puede utilizar el principio de extensión [218] para operar sobre dichos conjuntos. Este principio nos permite operar con términos lingüísticos a través de cálculos sobre sus funciones de pertenencia asociadas. El principal inconveniente de este modelo de computación es, que el uso de la aritmética difusa desarrollada a partir del principio de extensión, puede incrementar la imprecisión de los resultados ya que estos resultados vienen dados por números difusos que no tienen porque coincidir con ningún número difuso de los utilizados para definir la semántica de las etiquetas lingüísticas iniciales. En estos casos, se lleva a cabo un proceso de aproximación lingüística [22, 44] que identifica el número difuso obtenido en las operaciones con el número difuso más cercando asociado a una etiqueta inicial.

- *El modelo simbólico [46, 212]:* este modelo realiza cálculos directamente en las etiquetas usando el orden de dichas etiquetas en el conjunto de términos lingüísticos. El resultado de dichas operaciones estará en un dominio con-

tinuo (intervalo de valores de $\mathbb{R}$), por lo que estos puede que no coincidan con ningún valor asociado al orden que ocupa una etiqueta. En tal caso, hay que hacer una aproximación del resultado al valor discreto más cercano que represente una etiqueta.

Los modelos computacionales anteriores se explican con mayor detalle a continuación y se verá como presentan una serie de problemas a la hora de la precisión y la interpretabilidad. Para mejorar estos problemas se presentó un nuevo modelo de representación de información basado en la 2-tupla lingüística en [81] y posteriormente en [203]. Este modelo de reprentación define un modelo computacional que permite operar con estos valores obteniendo a su vez valores pertenecientes al mismo dominio. Este modelo será el utilizado a lo largo de la memoria para operar con la información lingüística manejada por los modelos de recomendación aquí presentados.

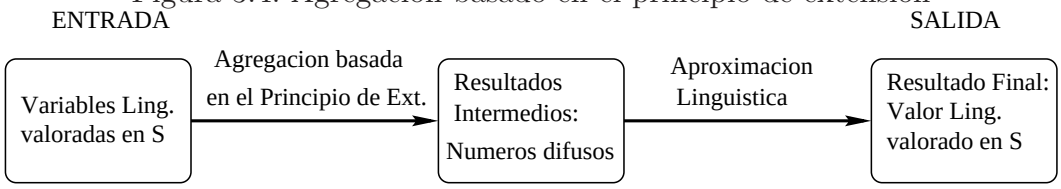
A continuación revisaremos los modelos computacionales que acabamos de nombrar.

3.3.1. Modelo computacional lingüístico basado en el principio de extensión

El principio de extensión fue introducido para generalizar las operaciones matemáticas definidas sobre datos precisos (crisp) a los conjuntos difusos. El uso de la aritmética basada en el principio de extensión [51] incrementa la incertidumbre o vaguedad de los resultados. Los resultados obtenidos mediante la aritmética difusa son números difusos, que habitualmente no coinciden con ningún término lingüístico del conjunto de términos inicial. Esto hace que sea necesario un proceso de aproximación para expresar los resultados en el dominio de expresión ori-

ginal. En la literatura podemos encontrar diferentes operadores de aproximación lingüística [22, 44].

Figura 3.4: Agregación basado en el principio de extensión

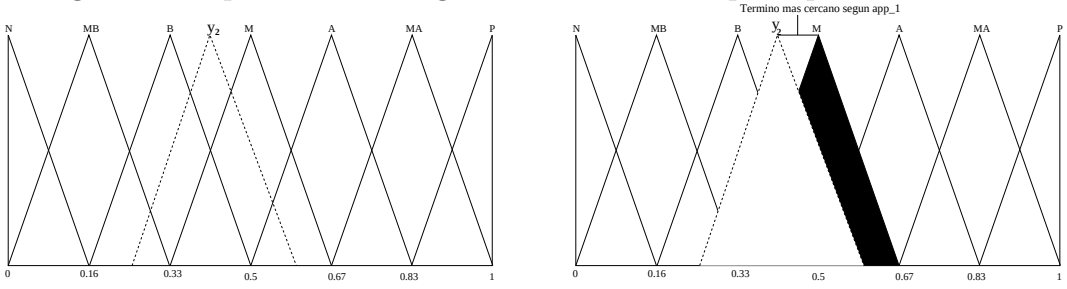


Un operador de agregación lingüístico basado en el principio de Extensión actúa de acuerdo a la siguiente expresión (ver figura 3.4):

$$S^n \overset{\tilde{F}}{\rightarrow} F(R) \overset{app_1(\cdot)}{\rightarrow} S$$

donde S^n simboliza el producto cartesiano n de S, F es un operador de agregación basado en el Principio de Extensión, $F(R)$ el conjunto de conjuntos difusos sobre el conjunto de números reales R, $app_1 : F(R) \rightarrow S$ es una función de aproximación lingüística que devuelve una etiqueta del conjunto lingüístico S cuyo semántica sea la más cercana al número difuso obtenido de la agregación, y S es el conjunto de términos inicial.

Figura 3.5: Aproximación lingüística basada en el principio de extensión



Por ejemplo, en la figura 3.5 a la izquierda, podemos ver que el resultado de una agregación, y_2, usando el principio de extensión. Este resultado no coincide

con ninguna de las etiquetas lingüísticas del conjunto de etiquetas inicial. Por lo tanto, si queremos que el resultado pertenezca a este conjunto de etiquetas, deberemos aproximarlo a la más cercana. En este caso, si nos fijamos en la figura de la derecha veremos que la más cercana es la etiqueta M. Como podemos ver e intuir, al hacer esta aproximación se pierde información.

3.3.2. Modelo simbólico

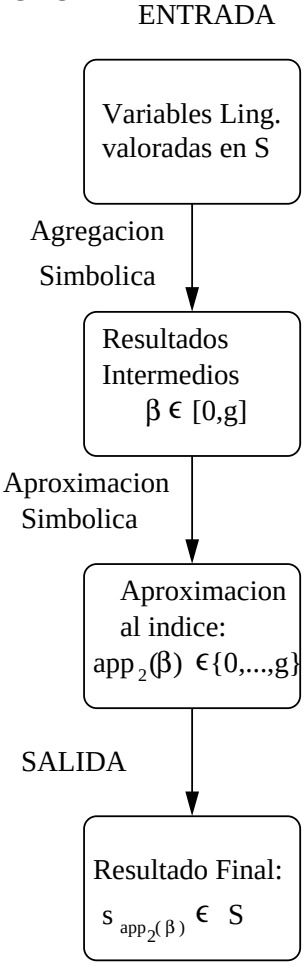
Un segundo enfoque para operar con información lingüística es el modelo computacional simbólico [47] que realiza los cálculos utilizando los índices de las etiquetas lingüísticas. Para realizar estos cálculos se asume que las etiquetas que forman el conjunto de términos lingüísticos, $S = \{s_0, \ldots, s_g\}$, tienen una estructura ordenada, $s_i < s_j$ si y sólo si $i < j$. Los resultados intermedios son también numéricos, $\alpha \in [0, g]$, el cual debe ser aproximado a una etiqueta lingüística por medio de una función de aproximación $app_2 : [0, g] \rightarrow \{0, \ldots, g\}$.

Esta función obtiene un valor numérico, de tal forma que represente el índice asociado con el termino lingüístico más cercano, $s_{app_2(\alpha)} \in S$. Formalmente, se puede expresar de la siguiente forma (ver figura 3.6):

$$S^n \overset{C}{\rightarrow} [0, g] \overset{app_2(\cdot)}{\rightarrow} \{0, \ldots, g\} \rightarrow S$$

donde C es un operador de agregación lingüístico simbólico y $app_2(\cdot)$ es una función de aproximación utilizada para obtener un índice $\{0, \ldots, g\}$ asociado a un termino de $S = \{s_0, \ldots, s_g\}$ a partir de un valor en $[0, g]$. El uso de esta función de aproximación conlleva pérdida de información en los resultados ya que, el valor obtenido es aproximado al índice más cercano.

Figura 3.6: Agregación con el modelo simbólico



3.3.3. Modelo computacional de las 2-tuplas

El modelo de representación lingüístico basado en 2-tuplas fue presentado en [81] para mejorar los problemas de pérdida de información en los procesos de computación con palabras que se presentaban en los modelos computacionales basados en el principio de extensión [44] y el modelo simbólico [46] tal y como hemos visto en las secciones anteriores.

Además este modelo se ha demostrado útil cuando trabaja en contextos de información no homogéneos [83, 85, 86] como algunos de los que abordaremos en

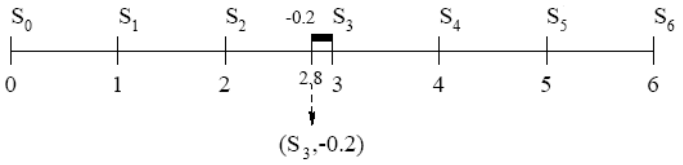
esta memoria.

Una Representación Lingüística con 2-tuplas

Para entender el modelo computacional basado en 2-tuplas primero revisaremos el modelo de representación de información lingüística basado en 2-tupla. Este modelo se basa en el concepto de translación simbólica.

Definición 3.2. *Sea* $S = \{s_0, ..., s_g\}$ *un conjunto de términos lingüísticos, y* $\beta \in [0, g]$ *un valor en el intervalo de granularidad de* S*. La Translación Simbólica (ver figura 3.7) de un término lingüístico* s_i *es un número valorado en el intervalo* $[-0.5, 0.5)$ *que expresa la "diferencia de información" entre una cantidad de información expresada por el valor* $\beta \in [0, g]$ *obtenido en una operación simbólica y el valor entero más próximo,* $i \in \{0, ..., g\}$*, que indica el índice de la etiqueta lingüística (*s_i*) más cercana en* S*.*

Figura 3.7: Etiqueta lingüística 2-tupla



A partir de este concepto el modelo de representación de la información lingüística modela dicha información mediante un par de valores o 2-tupla, (s_i, α_i) donde:

1. $s_i \in S$ representa una etiqueta lingüística

2. $\alpha_i \in [-0.5, 0.5)$ es un número que expresa el valor de distancia desde el resultado original β al índice de etiqueta lingüística más cercana (s_i) en el conjunto de términos lingüísticos S, es decir, su translación simbólica.

A continuación se define un conjunto de funciones básicas que permitirán convertir un número definido en el intervalo $[0, g]$ en su equivalente en 2-tuplas y viceversa.

Definición 3.3. *Sea* $S = \{s_0, ..., s_g\}$ *un conjunto de términos lingüísticos y* $\beta \in [0, g]$ *un valor que representa el resultado de una operación simbólica, entonces la 2-tupla lingüística que expresa la información equivalente a* β *se obtiene usando la siguiente función:*

$$\Delta : [0, g] \rightarrow S \times [-0.5, 0.5)$$

$$\Delta(\beta) = (s_i, \alpha), con \begin{cases} s_i, & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-0.5, 0.5) \end{cases}$$

donde round es el operador usual de redondeo, s_i *es la etiqueta con índice más cercano a* β *y* α *es el valor de la translación simbólica.*

Proposición 1. *Sea* $S = \{s_0, ..., s_g\}$ *un conjunto de términos lingüísticos y* (s_i, α) *una 2-tupla lingüística. Existe la función* Δ^{-1}*, tal que, dada una 2-tupla* (s_i, α) *esta función devuelve su valor numérico equivalente* $\beta \in [0, g]$*.*

Demostración.

Es trivial si consideramos la siguiente función.

$$\Delta^{-1} : S \times [-0.5, 0.5) \rightarrow [0, g]$$
$$\Delta^{-1}((s_i, \alpha)) = i + \alpha = \beta$$

Comentario 1: *A partir de las definiciones 1 y 2 y de la proposición 1, la conversión de un término lingüístico en una 2-tupla consiste en añadir el valor*

cero como translación simbólica:

$$s_i \in S \rightarrow (s_i, 0)$$

Modelo computacional lingüística para la representación con 2-tuplas

Este modelo de representación tiene un modelo computacional asociado presentado en [81] que permite:

1. Comparación de 2-tuplas
2. Agregación de 2-tuplas
3. Operador de Negación de una 2-tupla

A continuación estudiaremos cada una de estas operaciones

Comparación de 2-tuplas

La comparación de información representada con 2-tuplas se realiza de acuerdo a un orden lexicográfico clásico.

Sea (s_k, α_1) y (s_l, α_2) dos 2-tuplas que representan dos valoraciones:

- *Si* $k < l$ *entonces* (s_k, α_1) *es más pequeño que* (s_l, α_2)
- *Si* $k = l$ *entonces:*
    1. *Si* $\alpha_1 = \alpha_2$ *entonces* (s_k, α_1) *y* (s_l, α_2) *representan el mismo valor*
    2. *Si* $\alpha_1 < \alpha_2$ *entonces* (s_k, α_1) *es más pequeño que* (s_l, α_2)
    3. *Si* $\alpha_1 > \alpha_2$ *entonces* (s_k, α_1) *es mayor que* (s_l, α_2)

Ejemplos:

$(s_4, 0.3) < (s_5, -0.3)$

$(s_4, 0.3) > (s_4, -0.2)$

$(s_1, 0.2) = (s_1, 0.2)$

Agregación de 2-tuplas

La agregación de 2-tuplas lingüísticas permite obtener un valor "resumen" de un conjunto de valores. El resultado de esta operación será una 2-tupla lingüística. En [81] podemos encontrar varios operadores de agregación basados en los operadores de agregación clásicos, entre los cuales podemos destacar:

1. *Media aritmética:* este operador simboliza el concepto intuitivo de punto de equilibrio o centro del conjunto de valores.

   Definición 3.4. [81] *Sea* $x = \{(s_1, \alpha_1), ..., (s_m, \alpha_m)\}$ *un conjunto de 2-tuplas, su media aritmética se calcularía con el operador media aritmética extendida,* $\overline{x}^e$*, que es definido como,*

$$\overline{x}^e ((s_1, \alpha_1), ..., (s_m, \alpha_m)) = \Delta\left(\frac{1}{m}\sum_{i=1}^{m} \Delta^{-1}(s_i, \alpha_i)\right) = \Delta\left(\frac{1}{m}\sum_{i=1}^{m} \beta_i\right)$$

2. *Media ponderada:* la media ponderada permite que diferentes valores x_i tenga diferente importancia en la agregación. Esto se realiza asignando a cada valor x_i un peso asociado w_i, que indica cuál es la importancia de ese valor.

   Definición 3.5. [81] *Sea* $x = \{(s_1, \alpha_1), ..., (s_m, \alpha_m)\}$ *un conjunto de 2-tuplas y* $W = (w_!, ..., w_m)$ *un vector numérico con los pesos asociados a cada 2-tupla. La media ponderada extendida* $\overline{x}^e$ *se define como:*

$$\overline{x}^e = \Delta\left(\frac{\sum_{i=1}^{m} \Delta^{-1}(s_i, \alpha_i) \cdot w_i}{\sum_{i=1}^{m} w_i}\right) = \Delta\left(\frac{\sum_{i=1}^{m} \beta_i w_i}{\sum_{i=1}^{m} w_i}\right)$$

3. *Operador OWA (Ordered Weighted Aggregation):* este operador introducido por Yager en [210] es un operador de agregación ponderado, en el cuál, los pesos no están asociados a un valor predeterminado sino que están asociados a una posición determinada.

   Definición 3.6. *[81] Sea un* $x = \{(s_1, \alpha_1), ..., (s_m, \alpha_m)\}$ *un conjunto de 2-tuplas y* $W = (w_!, ..., w_m)$ *un vector de pesos asociado que satisface que (i)* $w_i \in [0, 1]$ *y (ii)* $\sum w_i = 1$. *El operador OWA extendido* F^e *para combinar 2-tuplas actúa como:*

$$F^e((s_1, \alpha_1), ..., (s_m, \alpha_m)) = \Delta\left(\sum_{j=1}^{m} w_j \cdot \beta_j^*\right)$$

   siendo β_j^* *el j-ésimo mayor valor de los* $\Delta^{-1}((s_i, \alpha_i))$

4. *Operador IOWA (Induced OWA operator):* este operador fue propuesto por Yager en [213] y es utilizado para agregar tuplas de la forma (v_i, a_i). El valor v_i es conocido como valor de inducción y a_i es el valor pasado como argumento. El objetivo del operador IOWA es realizar una agregación similar al que se realiza mediante el operador OWA, pero ordenando los valores, no por su propio valor, sino por el valor de inducción que tiene asociado cada valor. Este operador extendido a las 2-tuplas se define de la siguiente forma:

   Definición 3.7. *Sea un* $x = \{((s_1, \alpha_1), a_1), ..., ((s_m, \alpha_m), a_m)\}$ *un conjunto de 2-tuplas con sus valores de inducción, los cuales pueden estar expresados en cualquier dominio en donde podamos establecer un orden, y* $W = (w_!, ..., w_m)$ *un vector de pesos asociado que satisface que (i)* $w_i \in [0, 1]$ *y (ii)* $\sum w_i = 1$.

   El operador IOWA extendido F_w^e *para combinar 2-tuplas actúa como:*

$$F_w^e\left(\left(\left(s_1,\alpha_1\right),a_1\right),...,\left(\left(s_m,\alpha_m\right),a_m\right)\right)=\Delta\left(\sum_{j=1}^{m}w_j\cdot\beta_j^*\right)$$

siendo β_j^* la j-ésima 2-tupla obtenida de ordenar el vector x en función de las variables de inducción.

Operador de negación de una 2-tupla:

El operador de negación de una 2-tupla se define como:

$$Neg\left(s_i,\alpha\right)=\Delta\left(g-\Delta^{-1}\left(s_i,\alpha\right)\right)$$

donde $g+1$ es la cardinalidad de S, $s_i\in S=\{s_0,...,s_g\}$

3.4. Información lingüística multigranular

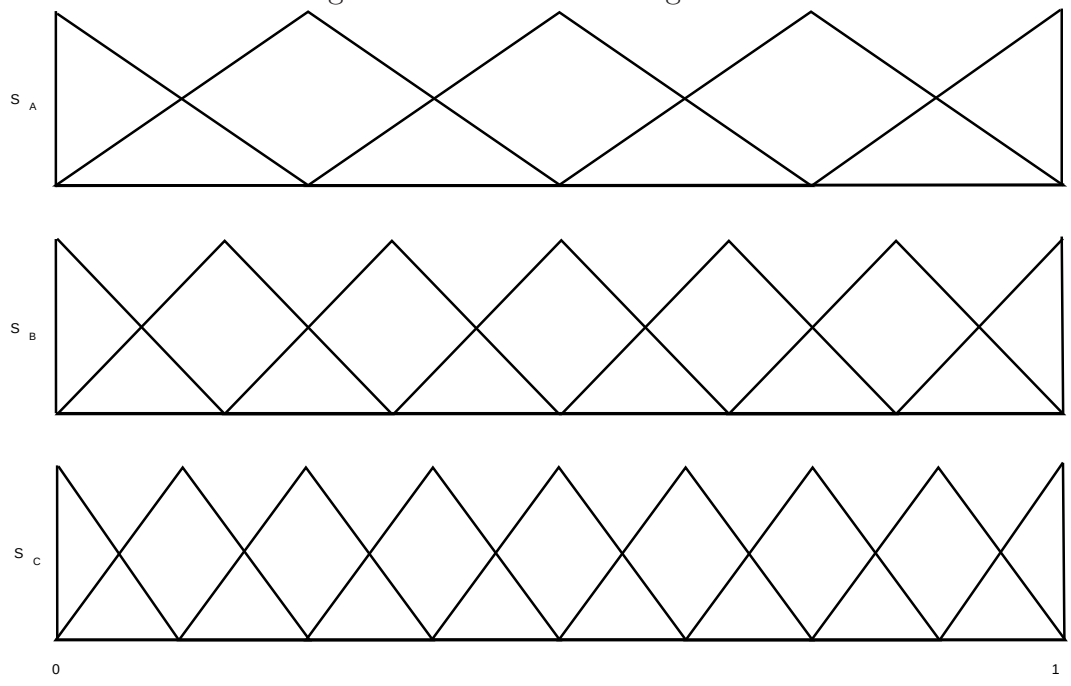
Como vimos anteriormente, un aspecto fundamental del modelado lingüístico de la información es la granularidad de la incertidumbre, o lo que es lo mismo, el número de etiquetas que utilizamos en la definición del conjunto de términos lingüísticos. Cuando nos encontramos en un problema en donde una o varias variables lingüísticas puede tomar valores de distintos conjuntos de etiquetas con distinta granularidad, decimos que estamos trabajando en un contexto con información lingüística multigranular [76, 97].

En esta memoria presentaremos dos modelos de sistemas de recomendación, uno en el capítulo 4 y en el capítulo 5 que trabajan en contextos multigranulares lingüísticos. Por lo que en esta sección vamos a revisar los conceptos, operadores y herramientas necesarios para manejar este tipo de información en nuestras propuestas.

En estos modelos, la multigranularidad puede deberse a dos causas. Por un lado, al grado distinto de conocimiento que tienen los usuarios del sistema a la hora de evaluar los aspectos de los productos: (i) tenemos los expertos, que describirán los productos con unas etiquetas con mayor granularidad ya que su grado de conocimiento es mayor, y (ii) tenemos usuarios, que se presupone que tendrán un grado de conocimiento menor y por lo tanto deberán utilizar conjuntos de etiquetas con menor granularidad. Por otro lado, también hay que tener en cuenta que los aspectos evaluados pueden tener una naturaleza distinta o ser adquiridos de forma distinta (mediante la vista, el tacto,...) y por lo tanto la granularidad entre distintos aspectos de un elemento también debería de ser la adecuada para cada situación.

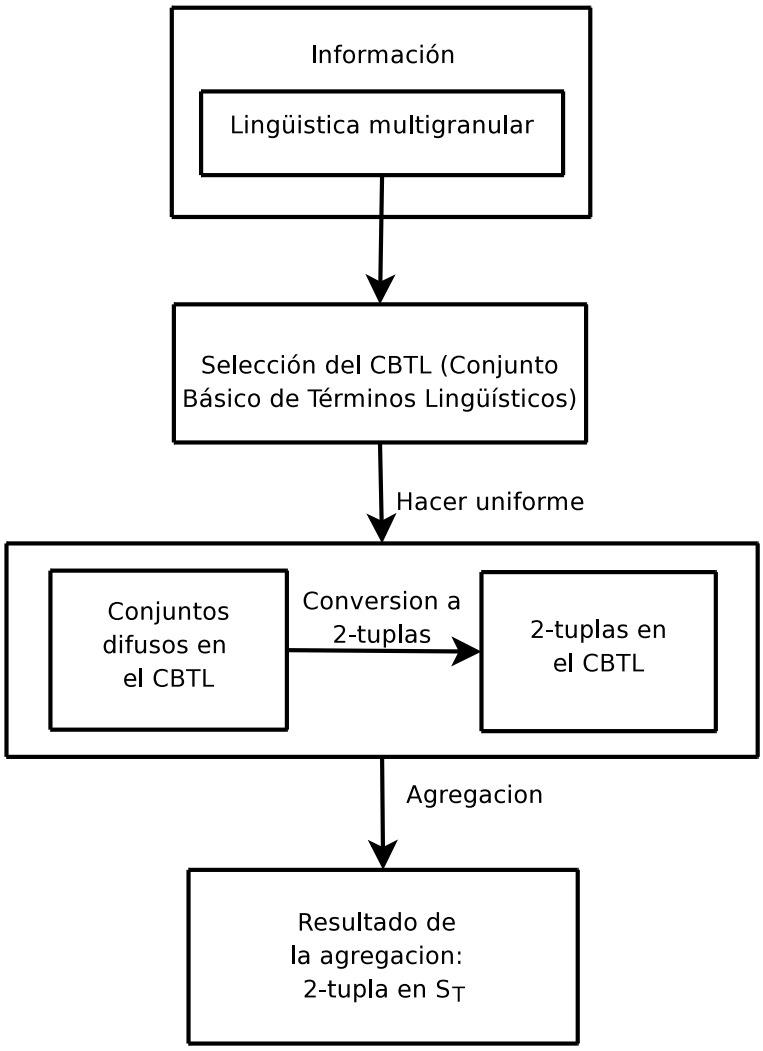
En la figura 3.8 podemos ver varios ejemplos de un conjunto de etiquetas S_A, S_B y S_C que definen un contexto multigranular para evaluar un elemento dado.

Figura 3.8: Contexto multigranular



No es posible operar directamente sobre variables lingüísticas valoradas en contextos multigranulares. Además hay que añadir que en estas situaciones no existen definidos procesos de normalización estándar para operar con este tipo de información ni operadores de agregación para la misma. Para solucionar estos problemas utilizaremos un proceso de unificación propuesto en [86], basado en el modelo de representación lingüístico con 2-tuplas, que se desarrolla de acuerdo al esquema que podemos ver en la figura 3.9.

Figura 3.9: Esquema de proceso de agregación de información lingüística multigranular



Para llevar a cabo el proceso de unificación de la información debemos seguir los siguientes pasos:

1. *Selección del CBTL.* Para convertir la información multigranular primero debemos seleccionar el dominio en donde unificaremos dicha información, A este dominio lo denominaremos CBTL (Conjunto Básico de Términos Lingüísticos).

2. *Transformación de la información lingüística multigranular en conjuntos difusos.* Consiste en transformar cada etiqueta lingüística de entrada a un "conjunto difuso" definido sobre un CBTL, que notaremos como S_T.

3. *Conversión de los conjuntos difusos a 2-tuplas lingüísticas.* Cada conjunto difuso sobre el CBTL obtenido en la fase anterior es transformado en una 2-tupla lingüística basada en la translación simbólica y valorada sobre el CBTL.

A continuación veremos cada uno de estos pasos en detalle.

3.4.1. Selección del CBTL

Antes de poder realizar cálculos sobre las variables lingüísticas valoradas en un contexto multigranular, tenemos que expresarlas en un único dominio, al que denominamos como CBTL [86] (Conjunto Básico de Términos Lingüísticos) y se denota como S_T.

El primer paso de este proceso es seleccionar dicho conjunto, S_T. Este conjunto debe cumplir que:

1. Permita mantener el grado de incertidumbre asociado a cada experto.

2. Conserve la capacidad de discriminación que expresan los valores de preferencia.

Partiendo de estas premisas, buscamos un CBTL con la máxima granularidad de los S_i que participan en el problema. Nos podemos encontrar con las siguiente situaciones:

- Que exista un único conjunto de términos lingüísticos con máxima granularidad, en este caso, lo seleccionaremos como S_T.

- Que existan dos o más conjuntos de etiquetas con máxima granularidad, entonces, S_T, será seleccionado dependiendo de la semántica de estos conjuntos de etiquetas, pudiéndose dar los dos siguientes casos a la hora de establecer S_T:

    1. Todos los conjuntos de etiquetas con máxima granularidad tienen idéntica semántica, entonces, S_T puede ser cualquiera de ellos.

    2. Existen varios conjuntos de etiquetas con diferente semántica. Entonces, S_T será un conjunto básico de términos lingüísticos con una cardinalidad mayor a la que una persona es capaz de discriminar (normalmente 11 ó 13, ver [142]).

Una vez seleccionado el CBTL que vamos a utilizar para unificar la información de entrada (expresar en un único dominio de expresión), ya podemos realizar las distintas fases del proceso de normalización.

3.4.2. Transformación de la información lingüística multigranular en conjuntos difusos

El primer paso para unificar la información lingüística multigranular sobre el dominio de expresión, S_T, es convertir cada etiqueta de entrada valorada en un conjunto de etiquetas, S_i, en un conjunto difuso sobre S_T. Para realizar esta conversión se definió la siguiente función de transformación [86]:

Definición 3.8. *Sea* $A = \{l_0, ..., l_p\}$ *y* $S_T = \{c_0, ..., c_g\}$ *dos conjunto de términos lingüísticos* $g \geq p$. *Definimos una función de transformación multigranular,* τ_{AS_T}, *como:*

$$\tau_{AS_T} : A \rightarrow F(S_T)$$

$$\tau_{AS_T}(l_i) = \left\{ \left(c_k, \alpha_k^i \right) / k \in \{0, ..., g\} \right\}, \forall l_i \in A$$

$$\alpha_k^i = \max_y \min \left\{ \mu_{l_i}(y), \mu_{c_k}(y) \right\},$$

donde $F(S_T)$ *es el conjunto de conjuntos difusos definidos sobre* S_T, *siendo* $\mu_{l_i}(y)$ *y* $\mu_{c_k}(y)$ *las funciones de pertenencia de los conjuntos difusos asociados a los términos* l_i *y* c_k, *respectivamente.*

El resultado de τ_{AS_T} para cualquier etiqueta lingüística de A es un conjunto difuso definido sobre el CBTL, S_T.

Para simplificar la notación, notaremos $\tau_{S_i S_T}(y_{ij})$ como r^{ij}, que representa cada conjunto difuso de preferencia mediante sus grados de pertenencia.

$$r^{ij} = \left(\alpha_o^{ij}, ..., \alpha_g^{ij} \right)$$

3.4.3. Conversión de Conjuntos Difusos en 2-tuplas Lingüísticas

Hasta este momento, lo que hemos hecho es unificar la información lingüística multigranular de entrada transformando cada valor lingüístico "y_{ij}" en un conjunto difuso sobre S_T utilizando $\tau_{S_i S_T}(y_{ij})$, tal que, $\tau_{S_i S_T}(y_{ij}) = \left\{(c_0, \alpha_o^{ij}), ..., \left(c_g, \alpha_g^{ij}\right)\right\}$. Ahora vamos a convertir estos conjuntos difusos a 2-tuplas lingüísticas valoradas sobre S_T. Para ello, definimos una función χ que calcula una 2-tupla $(s_k, \alpha)^{ij}$ que soporta la información del conjunto difuso $\tau_{S_i S_T}(y_{ij})$.

Definición 3.9. *Sea* $\tau_{S_i S_T}(y_{ij}) = \left\{(c_0, \alpha_o^{ij}), ..., \left(c_g, \alpha_g^{ij}\right)\right\}$ *un conjunto difuso que representa un término lingüístico* $l_i \in S_i$ *sobre el conjunto básico de términos lingüísticos* S_T*. Vamos a obtener una 2-tupla,* $(s_k, \alpha)^{ij}$*, que soporta la información del conjunto difuso mediante la siguiente función:*

$$\chi : F(S_T) \rightarrow S_T \times [-0.5, 0.5)$$

$$\chi\left(\{(c_k, \alpha_k), k = 0, ..., g, c_k \in S_T, \alpha_k \in [-0.5, 0.5]\}\right) = \Delta\left(\frac{\sum_{j=0}^{g} j\alpha_j^i}{\sum_{j=0}^{g} \alpha_j^i}\right) = (s_k, \alpha)^{ij}$$

donde $s_k \in S_T$ y $\alpha \in [-0.5, 0.5)$ es el valor de la traslación simbólica.

En este momento toda la información de entrada está expresada de forma uniforme sobre un único conjunto de términos lingüísticos, S_T, utilizando 2-tuplas. Sobre las que podremos operar sin limitaciones de granularidad

Capítulo 4

Modelo de recomendación basado en contenido con múltiples escalas lingüísticas sin información histórica

En el capítulo 2 vimos que uno de los principales inconvenientes que presentan los sistemas de recomendación es que muchas veces fuerzan a los usuarios a expresarse usando un único dominio similar al utilizado internamente por el sistema para operar con los datos, normalmente numérico. Aunque, en un principio puede parecer que esto no supone un gran inconveniente, existen situaciones donde puede ser inadecuado. Por ejemplo, en los sistemas de recomendación en los que los usuarios expresan percepciones de carácter cualitativo cuyo modelado se adecúa mejor al lingüístico. Además, normalmente, la descripción de los productos de la base de datos se realiza por expertos con un grado de conocimiento mayor que el de los usuarios. En el capítulo 3, vimos que en estas situaciones puede ser más adecuado el uso de múltiples escalas lingüísticas para reflejar esa diferencia de conocimiento.

En este capítulo, presentamos un modelo de recomendación basado en conte-

nido, que trabaja sobre el conjunto de características que describen los productos y no sobre la información contextual que describe a los mismos.

Este modelo ofrecerá un marco de trabajo lingüístico multrigranular que proporciona a los usuario una mayor flexibilidad a la hora de expresar sus preferencias y/o necesidades. Cada característica utilizada para describir los productos está evaluada mediante una escala lingüística seleccionada, según el grado conocimiento del usuario, y según la naturaleza de la misma. Si el usuario es un experto, entonces se utilizarán escalas lingüísticas con una granularidad mayor, es decir, la precisión es superior que el utilizado por los clientes finales para declarar sus preferencias.

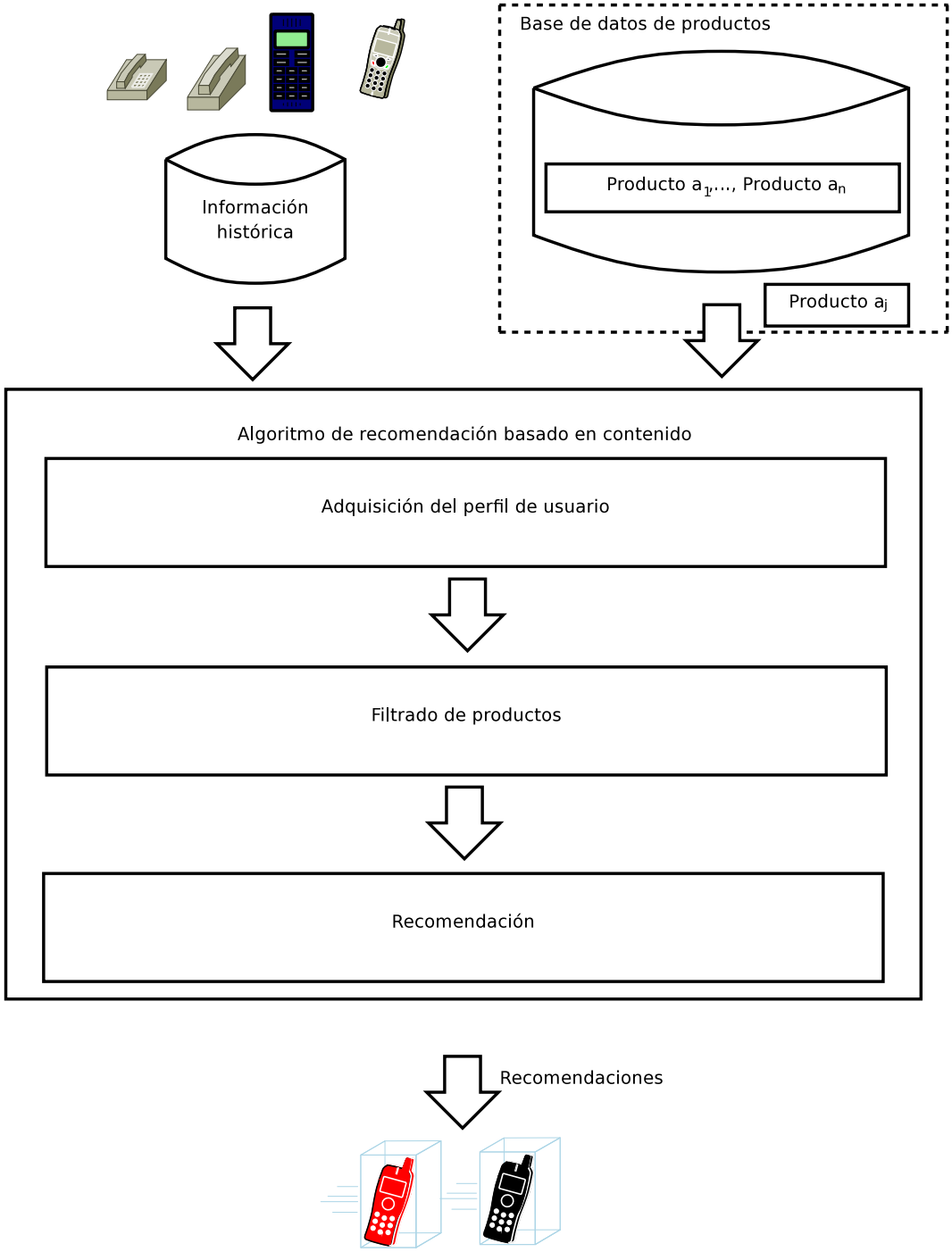
Al final de este capítulo, expondremos las conclusiones que se desprenden del funcionamiento de este modelo.

4.1. Modelo de recomendación basado en contenido con múltiples escalas lingüísticas

En esta sección presentamos un modelo de sistema de recomendación basado en contenido cuyas características principales vimos en el capítulo 2. Si recordamos brevemente su funcionamiento (ver figura 4.1), vemos que el perfil del usuario en estos sistemas recoge las características más importantes de todos los productos valorados positivamente por el usuario en el pasado.

Sin embargo, nuestra propuesta de modelo está orientada a procesos de recomendación en los que, por diversas razones, no existe información histórica. A pesar de esto, para generar las recomendaciones, utilizaremos los procesos y mecanismos propios de un modelo de recomendación basado en contenido. El modelo

Figura 4.1: Funcionamiento sistema de recomendación basado en contenido



propuesto será útil en situaciones donde:

- Lo realizado por el usuario en el pasado no está relacionado con lo que hace en el presente. Por ejemplo, en la venta de juguetes para niños, las necesidades de un niño a una determinada edad, no tiene porque tener relación con las necesidades de ese niño dos años después, o por ejemplo, un usuario necesita una recomendación sobre un restaurante porque quiere celebrar un hecho puntual (una boda, jubilación), pero que no tiene nada que ver con las veces que ha estado anteriormente en otros restaurantes.

- No existe información histórica sobre el usuario. Por ejemplo, supongamos un sistema de recomendación de restaurantes. Un número importante de sus usuarios pueden no haber interaccionado nunca con el sistema. Este problema se denomina el problema del nuevo usuario y ha sido estudiado con detenimiento en el capítulo 2.

- La información histórica no es suficiente como para generar las recomendaciones. Las técnicas técnicas clásicas basadas en contenido o colaborativas, necesitan que los usuarios hayan interaccionado un número de veces suficiente como para generar recomendaciones con garantías de éxito. Este problema también fue comentado en el capítulo 2 cuando se habló de la densidad de las bases de datos.

Además de generar las recomendaciones sin información histórica, este modelo está definido en un marco de trabajo lingüístico flexible. Para modelar adecuadamente las percepciones subjetivas de los usuarios y ofrecer una mayor flexibilidad a los usuarios, tanto clientes como expertos, a la hora de expresar sus necesidades o preferencias o describir los productos.

El esquema de funcionamiento de este modelo se muestra en la figura 4.3. Como se puede observar, la diferencia entre los modelos clásicos (figura 4.2) y nuestra propuesta, viene definida por la sustitución de la información histórica del usuario, por información explícita expresada por el usuario en un contexto multigranular. El resto de las fases serán iguales, aunque los procesos computacionales tendrán que ser redefinidos y adaptados para manejar información lingüística multigranular. Las fases de este modelo son las siguientes:

1. *Creación de la base de datos de productos:* el modelo de recomendación necesita una base de datos de productos en donde están almacenadas sus descripciones. Esta base de datos puede ser creada mediante:

   - Procesos automáticos de recuperación de información con o sin la supervisión de expertos.
   - Manualmente mediante un grupo de expertos.

2. *Adquisición del perfil de usuario:* en está fase el usuario expresará sus necesidades y éstas se almacenarán en el perfil de usuario, P_u, correspondiente. Esta fase es diferente de los modelos clásicos basados en contenido, ya que, anteriormente este perfil se generaba de forma automática analizando la información y características de los productos adquiridos por el usuario en el pasado. Debido a que esta información no existe, en situaciones cómo las anteriormente mencionadas, proponemos esta fase de adquisición explícita de la información.

3. *Filtrado de productos:* el sistema calculará la similitud entre el perfil de usuario y las descripciones de los productos almacenados en la base de datos de productos. Es importante remarcar que, el uso de múltiples escalas

Figura 4.2: Funcionamiento sistema de recomendación basado en contenido

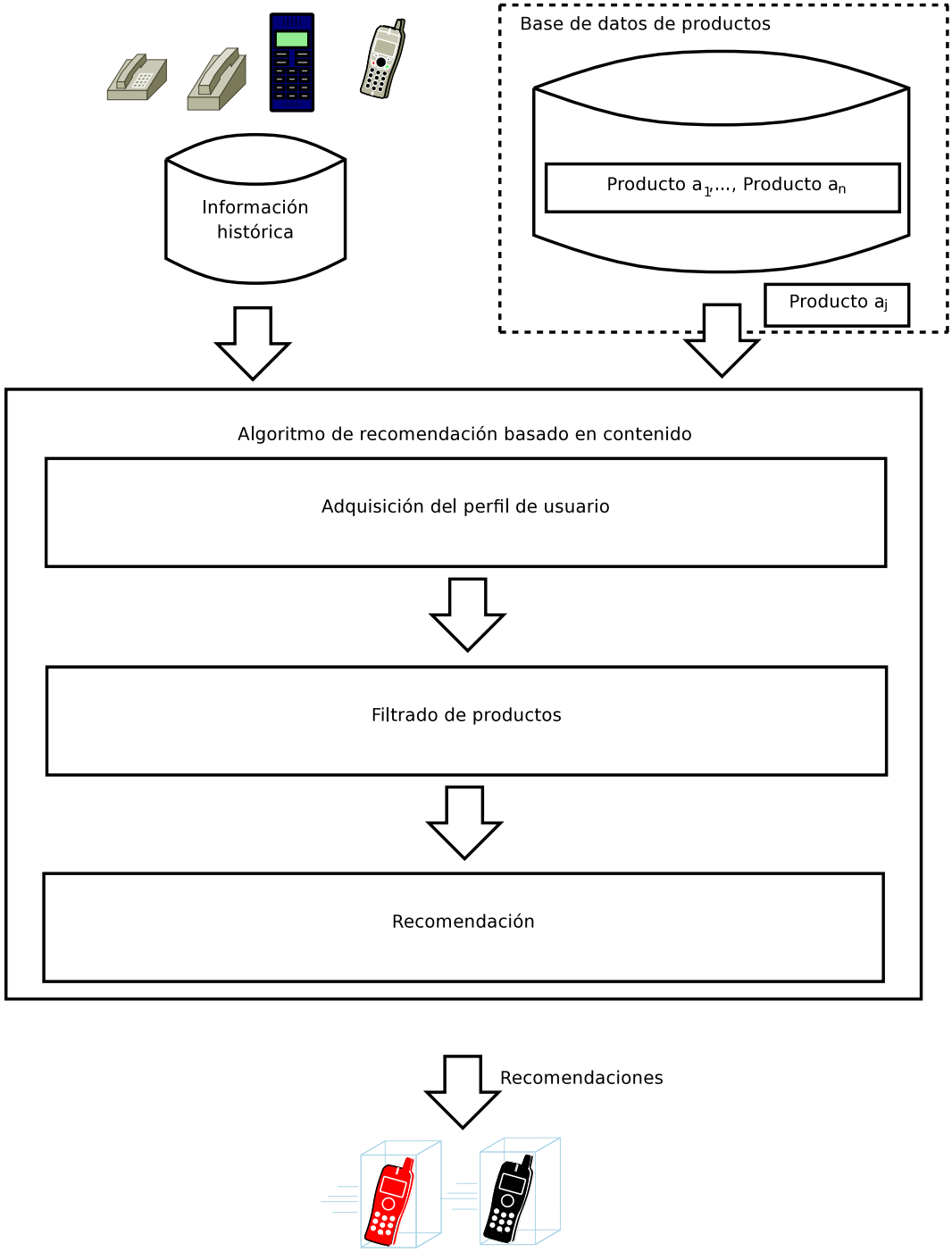
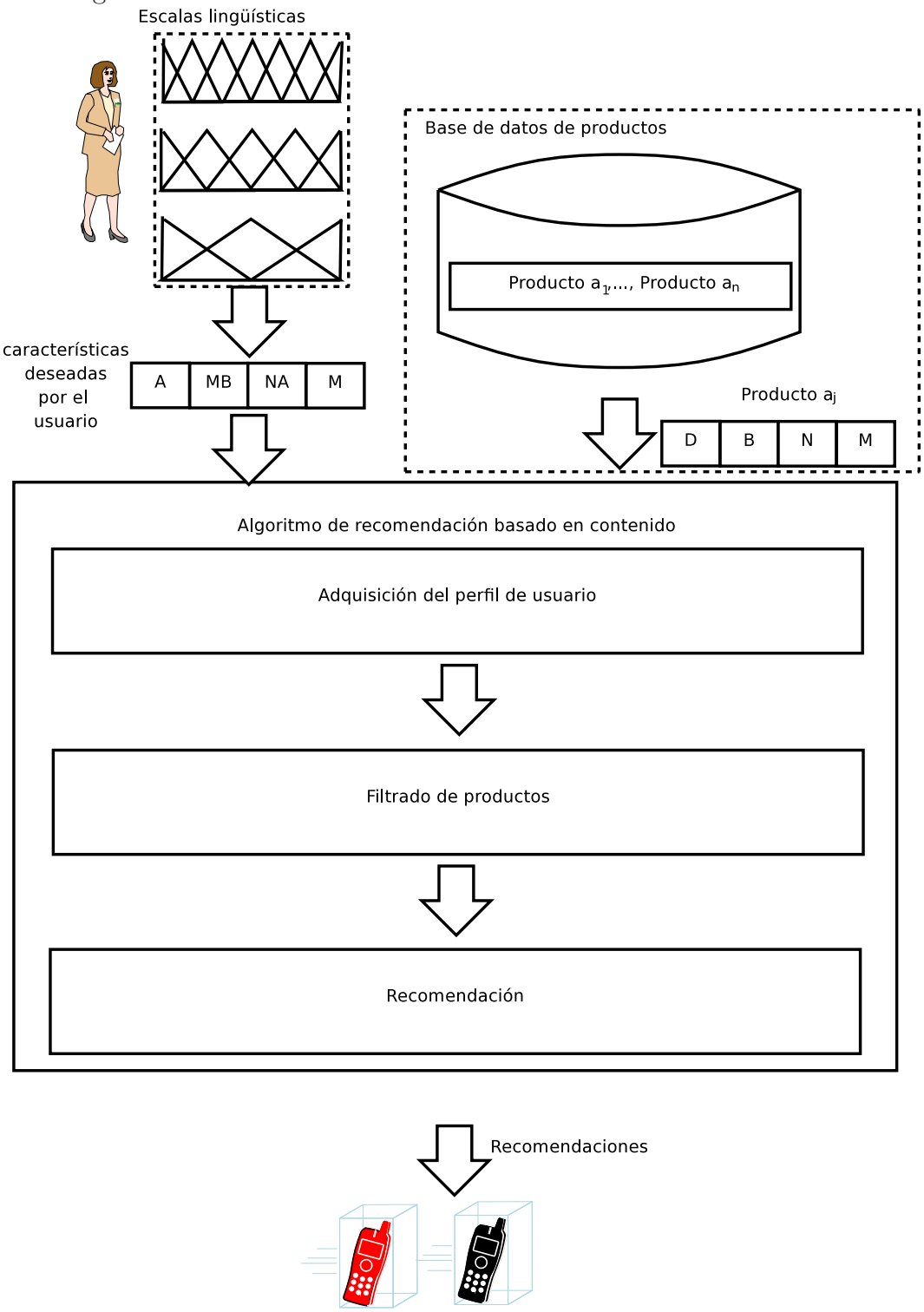


Figura 4.3: Sistema basado en contenido sin información histórica



lingüísticas supondrá la definición de operaciones y modelos que sean capaces de manejar este tipo de información. Así por ejemplo, necesitaremos definir operaciones que permitan calcular la similitud entre la semántica de dos etiquetas lingüísticas.

4. *Recomendación:* en esta última fase, el sistema deberá escoger los productos más adecuados para las necesidades del usuario. Para ello, ordenará los productos de acuerdo a aquellos productos que satisfagan mejor las necesidades del usuario.

A continuación, veremos el marco de recomendación que se propone en este modelo y, las herramientas necesarias para que el modelo genere las recomendaciones. En las secciones posteriores se explican las fases del modelo detalladamente.

4.1.1. Marco de recomendación

En esta sección, mostraremos el contexto en el que se define el proceso de recomendación del modelo que vamos a presentar. En él se fijará el modelado de preferencias que manejará el modelo y las herramientas necesarias para manejar dichas preferencias.

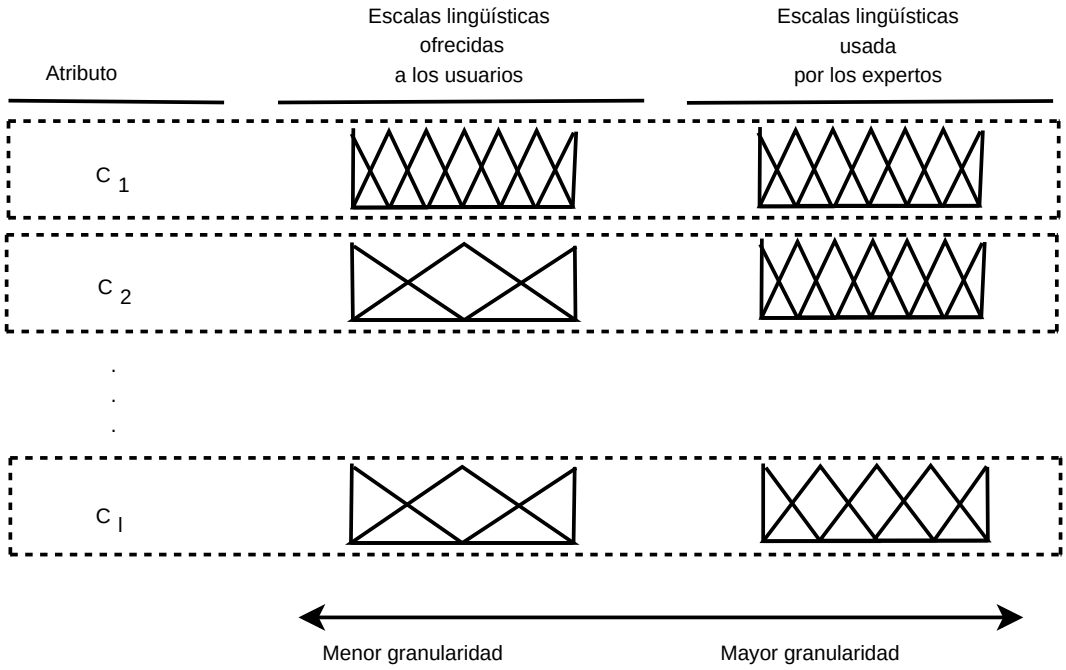
Nuestra propuesta consiste en un modelo que ofrecerá un contexto multigranular lingüístico (ver figura 4.4), debido fundamentalmente a las siguientes razones:

- La incertidumbre intrínseca a las percepciones hace que el grado de conocimiento sobre distintos atributos pueda ser distinto por lo que, el uso de distintas escalas de evaluación, puede ser adecuado para obtener mejores resultados.

- El grado de conocimiento que se tiene sobre estos atributos es distinto si, el

que los está evaluando es un experto o si es un usuario casual del sistema. Por lo que, sería conveniente ofrecerles a ambos escalas lingüísticas distintas que tengan en cuenta su grado de conocimiento.

Figura 4.4: Marco de trabajo del sistema basado en contenido sin información histórica



A continuación, veremos el contexto de definición del sistema de recomendación. El objetivo de este modelo es recomendar uno o varios productos de la base de productos, A:

$$A = \{a_1, ..., a_j, ..., a_n\}$$

donde cada uno de estos productos, a_j, está descrito por un conjunto de características

$$C = \{c_1, ..., c_k, ..., c_l\}.$$

Las necesidades o preferencias del usuario estarán almacenadas en el perfil

$$P_u = \{p_1^u, ..., p_k^u, ..., p_l^u\},$$

donde, p_k^u, representa la valoración, en la característica, c_k, del producto que deseada encontrar el usuario.

Como podemos ver en la figura 4.4, y debido al uso de un contexto lingüístico multigranular, las escalas lingüísticas utilizadas para valorar la característica, c_k, en el perfil de usuario, P_u, y en un producto, a_j, no tienen porqué coincidir. Sin embargo, de alguna forma, tenemos que llevar a cabo una comparación entre etiquetas pertenecientes a escalas lingüísticas distintas.

Para realizar los procesos computacionales necesarios en este modelo, operaremos sobre la semántica de las etiquetas lingüísticas representadas mediante números difusos, tal y como, describimos en el capítulo 3. Podemos llevar a cabo dicha comparación de etiquetas, empleando alguna medida de comparación que me permita conocer cuánto se parecen dos números difusos. En el apéndice B hemos realizado una breve revisión sobre las distintas medidas que podríamos utilizar para realizar dicha comparación.

4.1.2. Creación de la base de datos de productos

Un elemento fundamental, para un modelo de sistema de recomendación, es la generación de la base de datos de productos que tendrá una estructura similar a la mostrada en la tabla 4.1. Cada uno de los productos, a_j, de la base de datos estará descritos por un conjunto de características

$$C = \{c_1, ..., c_k, ..., c_l\}.$$

La base de datos puede generarse con distintas metodologías como por ejemplo:

Tabla 4.1: Base de datos

	c_1	...	c_l
a_1	v_1^1	...	v_l^1
...	...	...	...
a_n	v_1^n	...	v_l^n

- Mediante técnicas de recuperación de información de forma automática.
- De forma semiautomática o manual supervisada por un experto o un conjunto de expertos

Dada nuestra propuesta, cada uno de los objetos, a_j, (ver tabla 4.1) es descrito por medio de un vector de características,

$$F_{a_j} = \left\{v_1^j, ..., v_k^j, ..., v_l^j\right\}, j = 1, ..., n$$

Las valoraciones, v_k^j, de las características del producto, a_j, estarán expresadas usando la escala lingüística S_k, $v_k^j \in S_k$, donde $S_k = \left\{s_1^k, \cdots, s_2^k\right\}$ es el conjunto de términos lingüísticos definido para evaluar la característica, c_k, de acuerdo con el grado de conocimiento del experto o del grupo de expertos que está realizando dicha evaluación.

Una vez descritos todos estos productos, el sistema almacenará este conjunto de productos:

$$A = \{a_1, ..., a_j, ..., a_n\},$$

en una base de datos.

4.1.3. Adquisición del perfil del usuario

Dado que el modelo propuesto no utilizará información histórica, inicialmente, es necesario recoger las necesidades y preferencias del usuario y almacenarlas en

un perfil de usuario:

$$P_u = \{p_1^u, ..., p_k^u, ..., p_l^u\}.$$

Este perfil estará compuesto por un conjunto de atributos,

$$C = \{c_1, ..., c_k, ..., c_l\},$$

que el usuario utilizará para describir sus necesidades, preferencias o gustos del producto que desea encontrar. Este conjunto de atributos coincide con el utilizado en las descripciones de los productos de la base de datos.

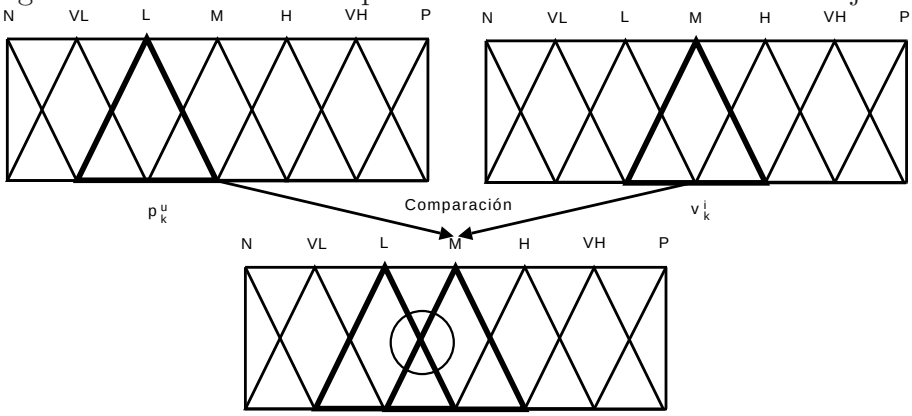
Diferentes usuarios pueden tener diferentes percepciones sobre sus preferencias o gustos, y de la misma forma, un mismo usuario puede tener un grado de conocimiento más o menos detallado dependiendo de la característica que esté valorando. Por esta razón, este modelo ofrece la posibilidad de que el usuario pueda seleccionar para expresar sus preferencias diferentes escalas lingüísticas de acuerdo a su grado de conocimiento y a la naturaleza del atributo que este valorando. De esta forma, el usuario trabajará en un contexto multigranular flexible (ver figura 4.4), en vez de forzarlos a utilizar una única escala, que podría reducir la calidad del conocimiento que tenemos sobre las necesidades del usuario.

Para ello, el sistema ofrecerá a sus usuarios una escala lingüística para cada atributo. Asi, la escala lingüística, S_{ku}, ha sido expresamente definida para valorar el atributo c_k del perfil de usuario P_u. Vemos cómo este proceso se adapta al marco de recomendación definido anteriormente

4.1.4. Filtrado de productos

En este momento, nuestro modelo cuenta con un perfil de usuario, $P_u = \{p_1^u, ..., p_l^u\}$, y las descripciones de los objetos, $F_{a_j} = \left\{v_1^j, ..., v_l^j\right\}$, de cada uno

Figura 4.5: Proceso de comparación mediante medidas de semejanza



de los productos, $\{a_j, j = 1, ..., n\}$ de la base de datos. Para averiguar cuáles son los productos más adecuados de la base de datos, para el perfil de usuario, P_u, el modelo llevará a cabo un proceso de filtrado entre, P_u, y cada objeto de la base de datos a_j. Para realizar estos cálculos, definiremos una medida de similitud (ver apéndice B), R_j^u, entre el perfil de usuario P_u y las características del producto $F_{a_j} = \left\{v_1^j, ..., v_l^j\right\}$ que pertenece a la base de datos de productos:

$$R_j^u = Similarity\left(P_u, F_{a_j}\right), j = 1, ..., n = \left(r_1^j, \ldots, r_l^j\right).$$

Debido a que, tanto el perfil del usuario, como las descripciones de los productos están valorados con etiquetas lingüísticas y su semántica está representada mediante números difusos, la similitud podrá ser calculada utilizando una medida de semejanza simple de usar y que fue propuesta en [52] (ver figura 4.5):

Definición 4.1. *Sea p_k^u el valor correspondiente al criterio, c_k, del perfil de usuario, P_u, y sea v_k^j el correspondiente valor lingüístico del criterio c_k del producto a_j (ver figura 4.5), la función que mide la similitud entre ambos valores lingüísticos se define de la siguiente forma:*

$$r_k^j = D\left(p_k^u, v_k^j\right) = \sup_x \min\left(f_{p_k^u}(x), f_{v_k^j}(x)\right), \tag{4.1}$$

donde $f_{p_k^u}$ es la función de pertenencia de los valores lingüísticos correspondiente al criterio, c_k en P_u, y $f_{v_k^j}$ es la función de pertenencia del valor lingüístico correspondiente al criterio, c_k, en F_{a_j}.

Para obtener la medida de similitud entre el perfil de usuario y el cada uno de los productos usaremos la siguiente función.

Definición 4.2. *Sea P_u el perfil de usuario y F_{a_j} la descripción del producto a_j, la similitud entre ambos se calcula mediante la siguiente función:*

$$Similarity\,(P_u, F_j) = R_j^u = \left(r_1^j, ..., r_l^j\right)$$

donde cada componente ,r_k^j, es calculado por la ecuación 4.1.

4.1.5. Recomendación

El objetivo del modelo recomendación es recomendar los productos más adecuados para los clientes. Hasta ahora, el modelo ha calculado la similitud, R_j^u, entre todos los productos, a_j, y el perfil de usuario, P_u. En esta fase del proceso de recomendación, la similitud puede ser interpretada como una preferencia, ya que, cuanto mayor sea el valor de esta preferencia, mejor satisface el producto las necesidades del usuario.

Por lo tanto, para poder llevar a cabo este objetivo, es necesario ordenar los productos de acuerdo a su similitud con los perfiles de usuario. Tenemos que tener en cuenta que dicha similitud está expresada por medio de conjuntos numéricos. Para ordenarlos y recomendar los más adecuados, nuestro modelo de recomendación usará un proceso de ordenación con tres fases que fue presentado en [76]:

1. Construir una relación de preferencia a partir de las medidas de similitud.

2. Calcular el grado de no dominancia (NDD) para cada producto.

3. Recomendar los mejores productos. Estos productos serán los primeros n con mayor grado de no dominancia.

A continuación mostraremos en detalle estas fases.

Construir una relación de preferencia a partir de las medidas de similitud

En este momento el sistema tiene una medida de similaridad, R_j^u, del perfil de usuario, P_u, con todos los productos, a_j. Nosotros proponemos la construcción de una relación de preferencia $Q_u = [q_{ij}]$, a partir de los valores de similitud, R_j^u. De esta forma, a partir de esta relación de preferencia, se podrá calcular cual es el producto, a_j, preferido por el usuario, o lo que es lo mismo, cual es el producto que mejor satisface sus necesidades.

Para ello, el modelo de recomendación utilizará una medida de inclusión. Sea A y B dos conjuntos de números, una medida de inclusión, $S(A, B)$, calculará el grado en el que A está incluido en B. En la literatura podemos encontrar diferentes tipos de medidas de inclusión [51, 175]; después de haber estudiado algunas de ellas, decidimos optar por la siguiente medida que es fácil de calcular y que presenta buenos resultados:

$$S(A, B) = \inf_x \min(1 - f_A(x) + f_B(x), 1), \tag{4.2}$$

donde $f_A(x)$ y $f_B(x)$ son las funciones de pertenencia de A y B respectivamente.

La función anterior calcula el grado en el que A está incluido en B, pero el modelo de recomendación necesita conocer cuanto de A cubre B para interpretar este valor como una preferencia. Para obtener el grado de preferencia de A sobre B, q_{AB}, necesitamos calcular su grado de inclusión de la siguiente forma:

$$q_{AB} = S\left(B, A\right)$$

Para construir la relación de preferencia, Q_u, proponemos utilizar la medida de inclusión 4.2 que mide cuanto cubre R_i^u a R_j^u, $\forall i, j \in \{1, ..., n\}$. Por lo tanto, para calcular el grado de preferencia utilizaremos la siguiente función:

Definición 4.3. *Sea R_i^u y R_j^u la similitud del producto a_i y a_j con respecto al perfil de usuario P_u. El valor, q_{ij}, que mide el grado de preferencia entre ambos productos se calcula de la siguiente forma:*

$$q_{ij} = S\left(R_j^u, R_i^u\right) = \inf_x \min\left(1 - f_{R_j^u}\left(x\right) + f_{R_i^u}\left(x\right), 1\right)$$

Realizando todos los cálculos para todas las alternativas obtendremos la relación de preferencia Q_u:

$$Q_u = \begin{pmatrix} q_{11} & \cdots & q_{1j} & \cdots & q_{1n} \\ \cdots & & \cdots & & \cdots \\ q_{j1} & \cdots & q_{jj} & \cdots & q_{jn} \\ \cdots & & \cdots & & \cdots \\ q_{n1} & \cdots & q_{nj} & \cdots & q_{nn} \end{pmatrix}$$

Calcular del grado de no dominancia

El objetivo de esta fase es calcular un valor que ayude a elegir y ordenar todos los productos de la base de datos, de forma que, fácilmente, pueda ordenar los productos según el grado en el que satisfagan las necesidades del usuario. Se pueden emplear distintas medidas para realizar esta tarea [153]. Este modelo

de recomendación utilizará el grado de no dominancia (NDD) que indica que alternativas no están dominadas por las otras.

Definición 4.4. [153]: *Sea $Q = [q_{ij}]$ un relación de preferencia difusa definida sobre un conjunto de alternativas X. Para la alternativa x_i su grado de no dominancia, NDD_i, se obtiene de la siguiente forma:*

$$NDD_i = \min_{X_j} \left\{ 1 - q^s_{ji}, j \neq i \right\}, \tag{4.3}$$

donde $q^s_{ji} = \max(q_{ji} - q_{ij}, 0)$ representa el grado en el cual x_i es estrictamente dominado por x_j. El grado de no dominancia (NDD) de todos los productos se obtiene de acuerdo a la ecuación 4.3.

Por lo tanto, para calcular el grado NDD de los distintos productos se realizarán los pasos que se describen a continuación:

1. Calcular el grado de preferencia estricto de la relación, Q^s_u, a partir de Q_u:

$$Q^s_u = \left[q^s_{ij} \right], \text{ donde } q^s_{ij} = \max(q_{ij} - q_{ji}, 0)$$

2. Calcular NDD de cada producto a_j como:

$$NDD_j = \min_{j \neq i} \left\{ 1 - q^s_{ij} \right\}$$

Recomendar los mejores productos

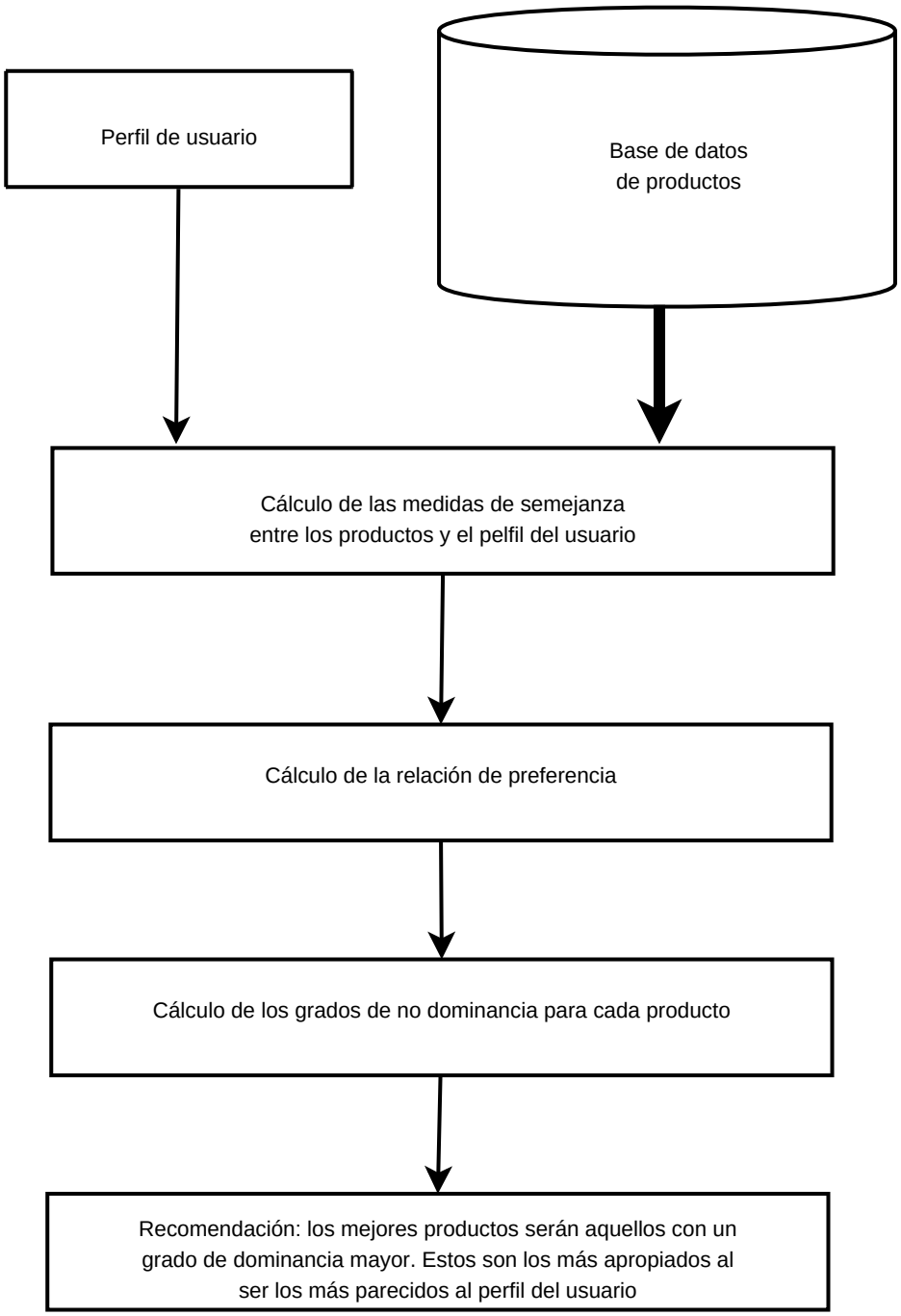
En este paso, el sistema recomendará los n mejores productos, es decir, aquellos que mejor satisfacen las necesidades del usuario. De acuerdo con nuestra interpretación, estos productos son aquellos que tiene un grado de NDD más alto, ya que, son los productos menos dominados. Los productos con el grado de NDD más

alto coincidirán con aquellos productos que han obtenido una similitud mayor con el perfil de usuario y por lo tanto, son los que mejor satisfacen sus necesidades.

Debemos tener en cuenta que puede haber varios productos con el mismo NDD. Éstos ocuparán la misma posición en la ordenación.

Un esquema general del proceso de recomendación que hemos seguido es el siguiente es el que se muestra en la figura 4.6.

Figura 4.6: Esquema sistema de recomendación



4.2. Ejemplo de aplicación del modelo propuesto

Supongamos un sistema de recomendación para recomendar juguetes para niños. En estas situaciones puede ser difícil, e incluso desaconsejable, utilizar la información histórica que tengamos del usuario. La principal razón es, que las necesidades de los niños o las intenciones del regalo suelen variar de una forma apreciable a lo largo del tiempo, muchas veces no teniendo nada que ver las necesidades de un niño tiempo atrás con las necesidades actuales. Por tanto, vamos a ver como funcionaría el modelo propuesto en este capítulo para este tipo de problemas, viendo cada uno de los elementos y fases del modelo propuesto para este ejemplo.

1. *Creación de la base de datos de productos*

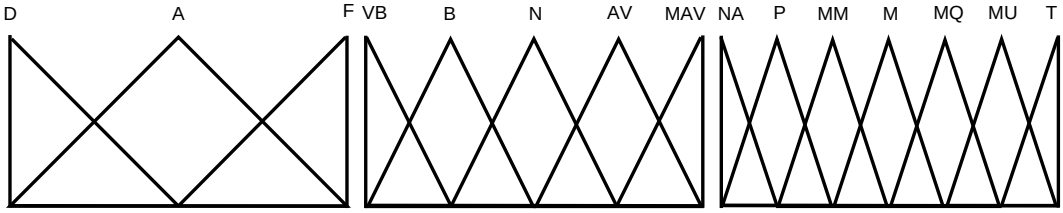
   En este ejemplo cada juguete vendrá descrito por un conjunto de características $C = \{JI, HM, HMU, HL, HMT, HC, HV, FA\}$ cuyo significado es:

   - *Juego independiente (JI):* este parámetro de aprendizaje promueve la autoestima y la confianza en los niños.
   - *Habilidades matemáticas (HM):* mide si los niños están involucrados en actividades de solución de problemas, razonamiento...
   - *Habilidades musicales (HMU):* si el juguete atrae al niño en la realización de actividades musicales.
   - *Habilidades lingüísticas (HL):* fomenta las habilidades verbales del niño.
   - *Habilidades motoras (HMT):* promueve y desarrolla las habilidades atléticas de los niños y/o la coordinación ojo/mano.

Tabla 4.2: Escalas lingüísticas A,B y C

Escala lingüística A		Escala lingüística B		Escala lingüística C	
Difícil (D)	$(0, 0, 0.5)$	**Muy básico (MB)**	$(0, 0, 0.25)$	**Nada (NA)**	$(0, 0, 0.16)$
Adecuado (A)	$(0, 0.5, 1)$	**Básico (B)**	$(0, 0.25, 0.5)$	**Un poco (P)**	$(0, 0.16, 0.33)$
Fácil (F)	$(0.5, 1, 1)$	**Normal (N)**	$(0.25, 0.5, 0.75)$	**Menos que la media (MM)**	$(0.16, 0.33, 0.5)$
		Avanzado (AV)	$(0.5, 0.75, 1)$	**Mediano (M)**	$(0.33, 0.5, 0.66)$
		Muy Avanzado (MAV)	$(0.75, 1, 1)$	**Mas que la media (MQ)**	$(0.5, 0.66, 0.83)$
				Mucho (MU)	$(0.66, 0.83, 1)$
				Todo (T)	$(0.83, 1, 1)$

Figura 4.7: Términos lingüísticos A,B y C



- *Habilidades de cooperación (HC):* mejora la cooperación y la interacción con el objetivo con conseguir logros comunes.
- *Habilidades visuales(HV):* estimula al niño en la evaluación visual y en actividades que aumentas la creatividad.
- *Facilidad de aprendizaje de como se juega (FA):* algunos juguetes requieren menos tiempo que otros a la hora de aprender como jugar con ellos.

Cada una de estas características estará valorada mediante una escala lingüística que no tiene porque coincidir con las utilizadas en las otras características. Utilizaremos las escalas lingüísticas definidas en la tabla 4.2 y cuya semántica se puede ver en la figura 4.7.

Las características que describen los productos han sido valorados usando las siguientes escalas lingüísticas:

Tabla 4.3: Descripciones de los productos de la base de datos

Juguete	JI	HM	HMU	HL	HMT	HC	HV	FA
T_1	NA	MV	MB	N	M	MB	MQ	D
T_2	P	B	MAV	MB	M	B	MU	D
T_3	M	B	B	AV	MM	N	T	A
T_4	MM	N	N	N	T	AV	M	A
T_5	M	AV	MAV	AV	MU	MAV	MM	D
T_6	M	MB	N	N	MQ	N	NA	A
T_7	MQ	N	N	MAV	M	AV	M	A
T_8	MU	MAV	N	N	NA	N	M	A
T_9	NA	N	B	MAV	NA	MAV	NA	D
T_{10}	P	AV	N	N	MQ	N	MU	D

- *Juego independiente (JI):* escala lingüística C.
- *Habilidades matemáticas (HM):* escala lingüística B.
- *Habilidades musicales (HMU):* escala lingüística B.
- *Habilidades lingüísticas (HL):* escala lingüística B.
- *Habilidades motoras (HMT):* escala lingüística C.
- *Habilidades de cooperación (HC):* escala lingüística B.
- *Habilidades visuales(HV):* escala lingüística C.
- *Facilidad de aprendizaje de como se juega (FA):* escala lingüística A.

Para seguir este ejemplo, mostraremos una vista parcial de la base datos (ver tabla 4.3) con 10 productos. En problemas reales, se emplean más características y escalas lingüísticas para describirlos. Cómo indicamos en la sección 4.1.2, esta base datos ha podido ser creada de forma automática mediante técnicas de recuperación de información, de forma semiautomática o de forma manual con la supervisión y ayuda de un experto o grupo de expertos.

2. *Adquisición del perfil del usuario*

   El sistema requerirá al usuario o cliente que exprese sus preferencias y/o necesidades con respecto al producto que desea. De esta forma, se obtendrá el perfil de usuario. Supongamos que el perfil proporcionando es el que se muestra en la tabla 4.4 y que describe las necesidades del usuario.

Tabla 4.4: Perfil del usuario

JI	HM	HMU	HL	HMT	HC	HV	FA
M	**B**	**B**	**AV**	**M**	**NA**	**T**	**A**

3. *Filtrado de productos*

   El modelo propuesto calcula la similitud del perfil de usuario, P_u, con los juguetes, T_j, de la base de datos. Supongamos la descripción del juguete T_1 tal y como se ve en la tabla 4.5. Se efectuarán los siguiente cálculos:

Tabla 4.5: Descripción del juguete T_1

JI	HM	HMU	HL	HMT	HC	HV	FA
N	MB	MB	N	M	MB	MQ	D

Similitud entre el perfil de usuario P_u y el juguete T_1:

$$\begin{aligned} R_1^u &= Similarity\,(P_u, T_1) = \left(r_1^1, r_2^1, r_3^1, r_4^1, r_5^1, r_6^1, r_7^1, r_8^1\right) \\ &= (0, 0.5, 0.5, 0.5, 1, 0, 0, 0.5) \end{aligned}$$

Los valores de similitud obtenidos se han calculado como sigue:

$$r_1^1 = \sup_x \text{mín}\,(M, N) = 0 \qquad r_2^1 = \sup_x \text{mín}\,(B, MB) = 0.5$$
$$r_3^1 = \sup_x \text{mín}\,(B, MB) = 0.5 \quad r_4^1 = \sup_x \text{mín}\,(AV, N) = 0.5$$
$$r_5^1 = \sup_x \text{mín}\,(M, M) = 1 \qquad r_6^1 = \sup_x \text{mín}\,(NA, MB) = 0$$
$$r_7^1 = \sup_x \text{mín}\,(T, MQ) = 0 \qquad r_8^1 = \sup_x \text{mín}\,(A, D) = 0.5$$

Los grados de similitud obtenidos en el paso anterior para todos los productos T_j con $j \in \{1, \cdots, 10\}$ son los que se muestran en la tabla 4.6.

Tabla 4.6: Grados de similitud entre el perfil del usuario y cada juguete

T_1	T_2	T_3
$R^u_{T_1} = (0; 0.5; 0.5; 0.5; 1; 0; 0; 0.5)$	$R^u_{T_2} = (0; 1; 0; 0; 1; 0.5; 0.5; 0.5)$	$R^u_{T_3} = (1; 1; 1; 1; 0.5; 1; 1; 1)$
T_4	T_5	T_6
$R^u_{T_4} = (0.5; 0.5; 0.5; 0.5; 0; 0.5; 0; 1)$	$R^u_{T_5} = (1; 0; 0; 1; 0; 0; 0; 0.5)$	$R^u_{T_6} = (1; 0.5; 0.5; 0.5; 0.5; 1; 0; 1)$
T_7	T_8	T_9
$R^u_{T_7} = (0.5; 0.5; 0.5; 0.5; 1; 0.5; 0; 1)$	$R^u_{T_8} = (0; 0; 0.5; 0.5; 0; 1; 0; 1)$	$R^u_{T_9} = (0; 0.5; 1; 0.5; 0; 0; 0; 0.5)$
T_{10}		
$R^u_{T_{10}} = (0; 0; 0.5; 0.5; 0.5; 1; 0.5; 0.5)$		

4. *Recomendación*

   Para recomendar los productos más adecuados, el sistema deberá realizar las siguientes fases:

   a) Construir una relación de preferencia a partir de las medidas de similitud

   A partir de la tabla 4.6 se obtiene la siguiente relación de preferencia,

Q_u:

$$Q_u = \begin{pmatrix} - & 0.5 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0.5 & - & 0 & 0.5 & 0 & 0 & 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & - & 1 & 1 & 1 & 0.5 & 1 & 1 & 1 \\ 0 & 0 & 0 & - & 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & - & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 1 & 0.5 & - & 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 1 & 0.5 & 0.5 & - & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & - & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & - & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0.5 & 0.5 & - \end{pmatrix}$$

Algunos ejemplos de como se obtienen los valores de esta matriz son:

$$\begin{aligned} q_{12} = \; & \text{ínf}_x \,\text{mín}\left(1 - f_{T_2}(x) + f_{T_1}(x), 1\right) = (((1-0+0) \wedge 1) \wedge \\ & (((1-1+0.5) \wedge 1) \wedge (((1-0+0.5) \wedge 1) \wedge (((1-1+0.5) \wedge 1) \wedge \\ & (((1-0.5+1) \wedge 1) \wedge (((1-0.5+0) \wedge 1) \wedge (((1-0.5+0) \wedge 1) \wedge \\ & (((1-0.5+0.5) \wedge 1)) = 0.5 \end{aligned}$$

$$\begin{aligned} q_{13} = \; & \text{ínf}_x \,\text{mín}\left(1 - f_{T_3}(x) + f_{T_1}(x), 1\right) = (((1-1+0) \wedge 1) \wedge \\ & (((1-1+0.5) \wedge 1) \wedge (((1-1+0.5) \wedge 1) \wedge (((1-1+0.5) \wedge 1) \wedge \\ & (((1-0.5+1) \wedge 1) \wedge (((1-1+0) \wedge 1) \wedge (((1-1+0) \wedge 1) \wedge \\ & (((1-1+0.5) \wedge 1)) = 0 \end{aligned}$$

$$\begin{aligned} q_{14} = \; & \text{ínf}_x \,\text{mín}\left(1 - f_{T_4}(x) + f_{T_1}(x), 1\right) = (((1-0.5+0) \wedge 1) \wedge \\ & (((1-0.5+0.5) \wedge 1) \wedge (((1-0.5+0.5) \wedge 1) \wedge (((1-0.5+0.5) \wedge 1) \wedge \\ & (((1-0+1) \wedge 1) \wedge (((1-0.5+0) \wedge 1) \wedge (((1-0+0) \wedge 1) \wedge \\ & (((1-1+0.5) \wedge 1)) = 0.5 \end{aligned}$$

$$\dots$$

donde el símbolo $\wedge$ representa el operador mínimo.

b) *Calcular el grado de no dominancia*

Para ello, en primer lugar, hay que obtener la relación de preferencia estricta, Q_u^s, a partir de Q_u:

$$Q_u^s = \begin{pmatrix} - & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & - & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & - & 1 & 1 & 1 & 0.5 & 1 & 1 & 1 \\ 0 & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0.5 & 0.5 & - & 0 & 1 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 & 0.5 & 0 & - & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & - \end{pmatrix}$$

Donde los valores de la matriz se obtienen como:

$$q_{12}^s = \text{máx}\,(q_{12} - q_{21}, 0) = \text{máx}\,(0.5 - 0.5, 0) = 0$$

$$q_{13}^s = \text{máx}\,(q_{13} - q_{31}, 0) = \text{máx}\,(0 - 0.5, 0) = 0$$

$$q_{14}^s = \text{máx}\,(q_{14} - q_{41}, 0) = \text{máx}\,(0.5 - 0, 0) = 0.5$$

$$\dots,$$

Finalmente, el grado de no dominancia calculado para todos los productos, T_j, tal y como se muestra en la tabla 4.7. Dichos valores se obtienen como:

Tabla 4.7: Grado de no dominancia (NDD) de los juguetes

T_1	T_2	T_3	T_4	T_5
0.5	0.5	1	0	0
T_6	T_7	T_8	T_9	T_{10}
0	0.5	0	0	0

$$NDD_1 = \text{mín}\{ \quad (1-0), (1-0.5), (1-0), (1-0), (1-0.5), \\ (1-0), (1-0), (1-0), (1-0.5)\} = 0.5$$

$$NDD_2 = \text{mín}\{ \quad (1-0), (1-0.5), (1-0), (1-0), (1-0.5), \\ (1-0), (1-0), (1-0), (1-0)\} = 0.5$$

$$NDD_3 = \text{mín}\{ \quad (1-0), (1-0), (1-0), (1-0), (1-0), \\ (1-0), (1-0), (1-0), (1-0)\} = 1$$

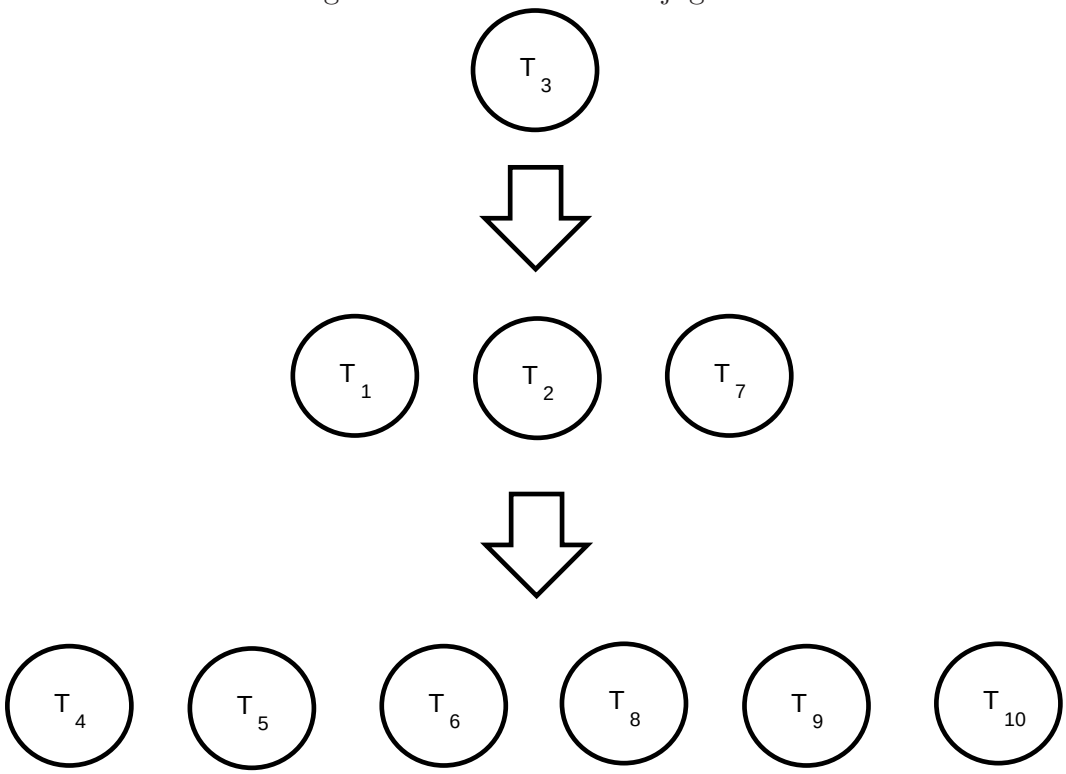
$$\dots$$

c) *Recomendar los mejores productos*

El objetivo final del sistema de recomendación es recomendar aquellos productos, T_j, que mejor se adaptan al perfil de usuario. Para ello, el modelo ordenará los productos T_j según su grado de no dominancia tal y como puede verse en la figura 4.8.

En está gráfica podemos ver que claramente el juguete T_3 es el que mayor grado tiene, y por lo tanto, es el que debería ser recomendado en primer lugar, seguido de los juguetes, T_1, T_2 y T_7.

Figura 4.8: Ordenación de juguetes



4.3. Conclusiones

Los sistemas de recomendación clásicos emplean información histórica del usuario para la generación de recomendación. Esta información está relacionada con las preferencias, necesidades, o gustos del usuario en el pasado y se utilizada para definir el perfil de usuario. Cuando un usuario expresa sus preferencias, se le suele requerir información en el dominio numérico, ya que, es el utilizado internamente por el sistema de recomendación para generar las recomendaciones.

En este capítulo hemos presentado un modelo de recomendación basado en contenido cuyas mejoras con respecto a los sistemas basados en contenido clásicos son las siguientes:

1. Se puede utilizar cuando no se tenga información histórica del usuario para generar las recomendaciones. Para ello el usuario proporcionará esta información en el momento en el que desee recibir las recomendaciones. Esta mejora permite utilizar este modelo en situaciones donde la información histórica, o bien no existe, es demasiado escasa, o bien no tienen sentido utilizarla ya que no está relacionada con las necesidades actuales.

2. Ofrece un marco lingüístico flexible de expresión a los usuarios del sistema. El sistema ofrecerá varias escalas lingüísticas para que sus usuarios empleen escalas adecuadas tanto a la naturaleza de las características evaluadas, como al grado de conocimiento que tienen sobre ellas.

Como resultado de estas mejoras, podremos utilizar este tipo de sistema de recomendación en entornos donde antes no era posible, debido a la falta de información histórica. Además, hemos mejorado la calidad de la información recogida del usuario sobre sus necesidades haciendo que mejoren las recomendaciones generadas por

el modelo.

Por último, hemos visto la aplicación de este modelo a un problema real de recomendación de juguetes.

Capítulo 5

Modelos de Recomendación Basados en Conocimiento

En nuestro proceso de investigación sobre sistemas de recomendación nos dimos cuenta que, aunque el modelo anterior era correcto y útil para su cometido, dentro de esta área de investigación se estaba avanzando con otros modelos de recomendación: los basados en conocimiento. Éstos fueron ideados para resolver este mismo tipo de problemas, la generación de recomendaciones cuando no existe o no es relevante la información histórica. Por esta razón dirigimos nuestra atención hacia ellos, teniendo en cuenta los resultados y propuestas del modelo anterior.

En este capítulo, presentamos dos modelos de sistemas de recomendación basados en conocimiento.

El primero de ellos, es un modelo de sistema de recomendación basado en conocimiento cuyos perfiles de usuario y descripciones de los productos a recomendar están definidos en un contexto lingüístico multigranular. Tal y como, ocurría en el modelo presentado en el capítulo anterior, queremos ofrecer al usuarios un contexto flexible donde puedan utilizar escalas lingüísticas más adecuadas para su grado de conocimiento.

Los objetivos que nos planteamos con este primer modelo fueron:

1. Facilitar el proceso de adquisición de las preferencias del usuario. El usuario proporciona un ejemplo, y en base a este ejemplo, se construye un perfil de usuario. Si este perfil no representa adecuadamente sus necesidades, el usuario tiene la oportunidad de refinar el perfil.

2. Mejorar el cálculo de similitud entre el perfil de usuario y los productos de la base de datos.

3. Reducir la carga computacional de la generación de recomendaciones. Los cálculos necesarios para esta generación de las recomendaciones son más simples.

El segundo modelo se presenta como una mejora del anterior. El objetivo principal de este segundo modelo es mejorar los mecanismos de construcción de perfil de usuario, de forma que, éste sea más fiel y no requiera mucha más información del usuario. El modelo solicitará un conjunto de ejemplos de las necesidades del usuario y una relación de preferencia sobre ellos, y a partir de esta información se construirá un perfil de usuario. Las ventajas que aportará serán:

1. Se ha eliminado la fase de refinamiento.

2. Hemos reducido la dependencia existente entre la calidad de las recomendaciones y la buena elección del ejemplo elegido para representar las necesidades del usuario.

A continuación presentaremos ambos modelos de recomendación. El primero de ellos, denominado modelo de recomendación basado en conocimiento con información lingüística multigranular (o RML) será presentado en la sección 5.1. El segundo, un sistema de recomendación basado en conocimiento basado en relacio-

nes de preferencia incompletas (o RRPI), lo veremos en la sección 5.2. Por último, en la sección 5.3 señalaremos algunas conclusiones.

5.1. RLM: Un modelo de recomendación basado en conocimiento con información lingüística multigranular

A continuación, presentamos un modelo de sistema de recomendación basado en conocimiento. Este tipo de sistemas de recomendación fueron estudiados en profundidad en el capítulo 2, y como hemos dicho al principio de este capítulo, fueron diseñados expresamente para generar recomendaciones cuando la información histórica sobre el usuario no existiera, no fuera suficiente o no fuera relevante para la recomendación.

El hecho de generar recomendaciones sin utilizar información histórica hace que este tipo de sistemas recomendación no presenten el problema del *nuevo usuario* ni del *nuevo producto*, tal y como, vimos en el capítulo 2. La mayor parte de los sistemas de recomendación basados en conocimiento emplean el razonamiento basado en casos [93, 111] para generar las recomendaciones.

En este modelo de recomendación, los productos de la base de datos están descritos por un vector de características donde cada elemento, normalmente, está relacionado con aspectos cualitativos del producto, con percepciones o con gustos. Por lo tanto, el dominio más adecuado para describirlas será el lingüístico. Además, se debe de tener en cuenta, que no sólo debemos usar una escala lingüística distinta para cada característica que estemos evaluando, sino que debemos tener en cuenta el grado de conocimiento de quien esté aportando esta información. Por

lo que, sería bueno modelar esta información mediante un contexto lingüístico multigranular.

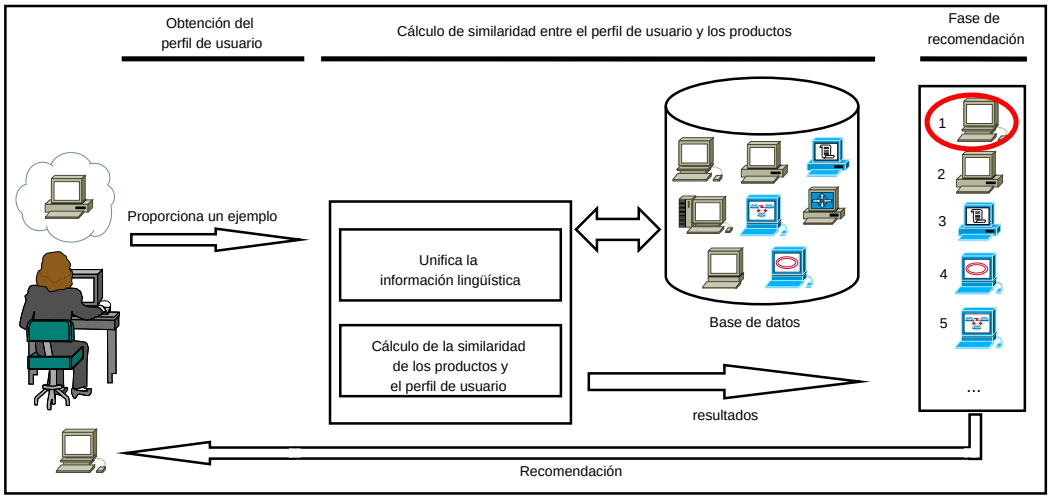
En esta propuesta, nos hemos centrado principalmente en los siguientes dos puntos:

- Crear un modelo de sistema de recomendación basado en conocimiento, cuyo funcionamiento esta basado en el razonamiento basado en casos, que permita obtener recomendaciones para los usuarios cuando la información histórica sobre ellos sea inexistente, escasa o no esté relacionada con la búsqueda actual.
- Ofrecer a los usuarios del modelo un marco de trabajo flexible, donde se pueden utilizar distintas escalas lingüísticas dependiendo de la característica que se este evaluando, y del grado de conocimiento que tenga la persona que lo está evaluando.

Este modelo desarrolla su actividad según el siguiente esquema (ver figura 5.1):

1. Creación de la base de datos de datos de productos.
2. Obtención del perfil de usuario. Esta fase se compone de dos pasos:
   a) Adquisición del ejemplo preferido por el usuario.
   b) Refinamiento ocasional de preferencias.
3. Filtrado de productos. Está compuesta de la siguientes fases:
   a) Unifica la información lingüística.
   b) Cálculo de la similitud de los productos y el perfil de usuario.
4. Fase de recomendación.

Figura 5.1: Modelo basado en conocimiento con información lingüística multigranular



A continuación presentaremos de forma detallada el marco de recomendación de este modelo, y seguidamente, presentaremos cada una de estas fases con más detalle.

5.1.1. Marco de recomendación

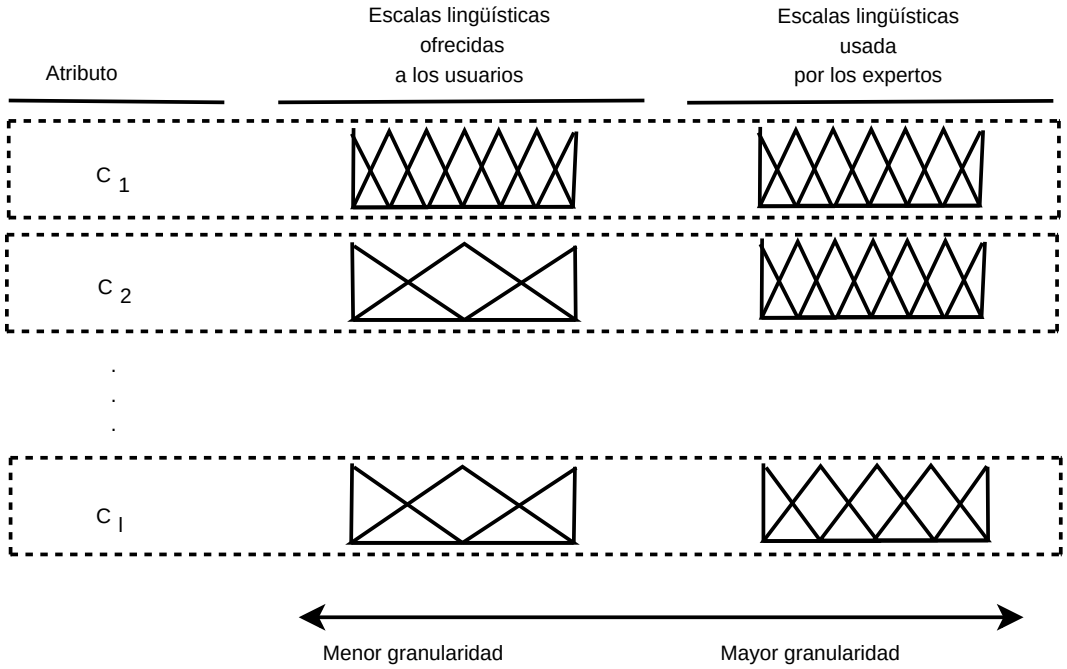
Al igual que en el capítulo 4, en esta sección, explicamos el contexto en el que se generarán las recomendaciones y qué herramientas empleamos para generarlas.

Como hemos introducido anteriormente, uno de los objetivos del modelo es ofrecer un marco de trabajo flexible (ver figura 5.2) donde:

1. Cada característica pueda ser evaluada mediante la escala lingüística más adecuada dependiendo de la naturaleza de ésta.

2. Se tenga en cuenta quien está proporcionando está información, si es un experto, la granularidad de la escala lingüística será mayor pues tiene un grado de conocimiento mayor que si fuera un cliente, el cual usará una escala lingüística con menor granularidad.

Figura 5.2: Marco de recomendación del sistema basado en conocimiento



Para llevar a cabo las recomendaciones, necesitaremos definir unas medidas de similitud que me permitan comparar la información almacenada en el perfil de usuario sobre las necesidades de éste, con la descripción de cada uno de los objetos que se puede recomendar. En este modelo, a diferencia del anterior, vamos a trabajar con medidas de similitud que necesitan que toda la información esté expresada en el mismo dominio, para reducir la complejidad computacional y obtener mejores resultados. Por lo tanto, en primer lugar, las valoraciones del perfil de usuario y de las descripciones de los productos se unificarán en un mismo dominio de expresión.

A continuación, revisaremos brevemente el proceso de unificación de información presentado en 3.4, y posteriormente, presentamos las medidas que utilizamos para calcular la similitud entre la información unificada.

Unificación información

En el modelo basado en contenido, el método de comparación del perfil de usuario con los productos era muy costoso tanto computacionalmente como en tiempo, aunque producía buenos resultados. Uno de los objetivos de este modelo es mejorar el rendimiento en la generación de las recomendaciones. Para ello, proponemos un nuevo método para calcular la similitud entre el perfil de usuario y cada objeto de la base de datos. En este método, se unifica la información en un único dominio de expresión, y el cálculo de la similitud se realiza cálculando la distancia entre dos conjuntos difusos pertenecientes al mismo dominio.

Para realizar el proceso de únificación realizaremos el proceso presentado en 3.4:

1. Seleccionar el CBTL o conjunto básico de términos lingüísticos.

2. Aplicar la función τ para unificar toda la información el CBTL. En este paso unificaremos el perfil de usuario,

$$P_e = \{p_1^e, ..., p_l^e\}$$

y las características

$$C = \{c_1, ..., c_k, ..., c_l\}$$

que describen cada uno de los productos, a_i de la base de productos,

$$A = \{a_1, ..., a_n\},$$

En este momento, las preferencias de los usuarios y las características de los productos están expresados mediante conjuntos difusos en el CBTL.

Cálculo de la similitud

Vimo que uno de los objetivos de este modelo es eliminar la complejidad en la comparación del perfil de usuario con las descripciones de los productos del modelo anterior para, de esta forma, generar recomendaciones con menor coste computacional y temporal.

Para ello, proponemos utilizar una medida de similitud basada en la distancia entre dos conjuntos difusos expresados en un mismo dominio lingüístico.

En [90] se hizo un estudio sobre medidas de similitud sobre este tipo de información. De él se desprende que el uso de valores centrales de los conjuntos difusos:

Definición 5.1. *Dado un conjunto difuso* $b^* = (\alpha_1, ..., \alpha_g)$ *definido sobre* $S = \{s_h\}$ *para* $h = 0, ..., g$ *obtenemos su valor central cv del siguiente modo:*

$$cv = \frac{\sum_{h=0}^{g} index\left(s_h\right) \alpha_h}{\sum_{h=0}^{g} \alpha_h},\ donde\ index\left(s_h\right) = h$$

cv representará la posición media o centro de la gravedad de la información contenida en el conjunto difuso b^**. El rango de este valor central es el intervalo cerrado* $[0, g]$.

Ésto permite definir una medida de similitud para conocer el grado de similitud, simple y util, entre el perfil de usuario, P_e, y un producto, a_j, de la base de datos, tal y como sigue:

Definición 5.2. *Para obtener la similitud total,* d_j*, entre* P_e *y* a_j *se utilizará la función, d, que definiremos a continuación:*

$$d_j = d\left(P_e, a_j\right) = \frac{1}{l} \sum_{k=1}^{l} sim\left(p_k^{*e}, v_k^{*j}\right),\ sim\left(p_k^{*e}, v_k^{*j}\right) \in V_{SIM}\left(P_e, a_j\right)$$

Donde $V_{SIM}(P_e, a_j) = \left\{sim\left(p_1^{*e}, v_1^{*j}\right), ..., sim\left(p_k^{*e}, v_k^{*j}\right), ..., sim\left(p_l^{*e}, v_l^{*j}\right)\right\}$*, es un vector que contiene la similitud de todos los atributos del perfil de usuario con respecto a la descripción del producto* a_j*.*

En este modelo, dichos valores se agregarán mediante una mediante un operador de agregación que podrá ser distinto dependiendo del problema.

La función *sim* calcula la similitud entre los valores de los atributos del perfil de usuario y el producto, a_j, y es definida como:

Esta función *sim* se define de la siguiente forma:

Definición 5.3. *Sea* $cv_k^{P_e}$ *el valor central del conjunto difuso* p_k^{*e}*, correspondiente al perfil de usuario,* P_e*, y* $cv_k^{a_j}$ *el valor central del conjunto difuso* v_k^{*j} *para del producto* a_j*, la función que calcula la similitud de ambos valores se define como:*

$$sim\left(p_k^{*e}, v_k^{*j}\right) = 1 - \left|\frac{cv_k^{P_e} - cv_k^{a_j}}{g}\right|$$

Una vez definido el marco de recomendación, presentaremos las distintas fases del modelo de recomendación.

5.1.2. Creación de la base de datos de productos

Como en todos los modelos vistos, éste también necesita una base de datos compuesta de las descripciones de los productos a recomendar. Dicha base de datos tendrá una estructura como la mostrada en la tabla 5.1. La base de datos puede generarse con distintas metodologías como por ejemplo:

- Mediante técnicas automáticas.

Tabla 5.1: Base de datos

	c_1	...	c_l
a_1	v_1^1	...	v_l^1
...	...	...	...
a_n	v_1^n	...	v_l^n

- De forma semiautomática o manual supervisada por un experto o un conjunto de expertos.

Esta base de datos contendrá n productos,

$$A = \{a_1, ..., a_n\},$$

donde cada producto, a_e, está descrito por un vector de valoraciones

$$F_e = \{v_1^e, ..., v_l^e\},$$

en donde cada elemento $v_k^e \in S_k$ es una valoración de la característica c_k del conjunto de características,

$$C = \{c_1, ..., c_k, ..., c_l\}$$

que describen los productos, siendo S_k la escala lingüística más adecuado para valorar la característica c_k.

5.1.3. Obtención del perfil de usuario

Una de las principales mejoras de este modelo, con respecto al anterior, está relacionada con la obtención del perfil del usuario. En el modelo anterior, se mostraban al usuario todas las características del perfil, y éste, valoraba cada una de las características para describir cuales eran sus gustos o necesidades. Esto conlleva una serie de desventajas que en este modelo se han intentado solucionar:

- El número de atributos que el usuario tiene que revisar para completar su perfil puede requerir demasiado tiempo y ser una de las causas por las que el usuario desista de su busqueda.

- El usuario no tiene porqué conocer todos los atributos que describen este perfil. Si no es un experto en la materia puede haber atributos que no pueda valorar, porque desconozca dicha característica. Por ejemplo, alguien no familiarizado con la automoción puede desconocer el significado del término EPS.

Existen varias alternativas para minimizar estos inconvenientes. Una de ellas sería mostrar sólo un conjunto reducido de las características más significativas. Sin embargo, esta solución conllevaría una perdida de expresividad para los usuarios. Otra alternativa que se propone en los modelos de recomendación basado en conocimiento es la de definir el perfil de usuario a partir de un ejemplo de las necesidades del usuario, y sólo modificar aquellas características que el usuario considere que no representan realmente sus necesidades. Esta solución evita conocer o valorar todas las características que definen el perfil de usario.

A continuación, explicaremos detenidamente los pasos que el modelo sigue para construir el perfil de usuario (ver figura 5.1).

Adquisición del ejemplo preferido por el usuario

El objetivo de esta fase es recoger la información inicial sobre las necesidades del usuario y construir un perfil de usuario inicial P_{e_0}.

El punto de partida para definir las necesidades del usuario se basa en la elección de un ejemplo, es decir, de un producto que el usuario expone como un caso de sus necesidades. Sea, a_e, el producto seleccionado como ejemplo de las

preferencias del usuario, u_e. Este producto estará descrito en la base de datos mediante un vector de utilidad

$$F_e = \{v_1^e, ..., v_l^e\},$$

donde, $v_k^e \in S_k$, es el valor asignado a dicho producto para la característica, c_k, expresado en S_k.

Este ejemplo, seleccionado por el usuario, genera un perfil de usuario inicial que notamos como

$$P_{e_0} = \{p_1^{e_0}, ..., p_l^{e_0}\}$$

donde, $p_k^{e_0} = v_k^e$, ya que, se asigna directamente los valores de la base de datos a los del perfil. En este perfil inicial, los conjuntos de términos lingüísticos son los mismos que los utilizados en la base de datos.

Refinamiento ocasional de preferencias

Esta fase tiene como objetivo ofrecer al usuario la posibilidad de refinar su perfil, debido a que éste ha podido elegir un ejemplo cercano a sus necesidades, pero que no refleje exactamente lo que necesita.

Para ello, el modelo ofrece al usuario la posibilidad de alterar algún valor de su perfil dando una valoración para una característica determinada expresada en una escala lingüística a su elección o la ya existente. En tal caso, para una característica, c_k, el usuario podrá asignarle una nueva valoración, $p_k^{e_1}$, expresada en otra escala lingüística, S_k^*, acorde con su grado de conocimiento y con la naturaleza de la característica que está evaluando.

Después de esta modificación, que es opcional, tenemos un perfil definitivo de usuario $P_e = \{p_1^e, ..., p_l^e\}$ donde $p_k^e \in S_k^e$ obtenidos del siguiente modo:

- $p_k^e = p_k^{e_0}, p_k^e \in S_k^e = S_k$ si la característica c_k no ha sido modificada
- $p_k^e = p_k^{e_1}, p_k^e \in S_k^e = S_k^*$ en caso contrario.

En esta fase de nuestro modelo, se ofrece a los usuarios la posibilidad de describir sus necesidades utilizando sus propios conjuntos de términos lingüísticos, S_k^*, acordes con su grado de conocimiento y con la naturaleza de la característica, sin necesidad de tener que adaptarse a una escala única fijada a priori.

5.1.4. Filtrado de productos

Al igual que en otros modelos anteriores. Este modelo ha de filtrar los productos de la base de datos con respecto al perfil del usuario, P_e, para encontrar cuáles son los productos más adecuados para el usuario.

En este modelo, para realizar este proceso se propone calcular la similitud entre los productos y el perfil de usuario, P_e, siguiendo el siguiente proceso:

1. Unificación la información lingüística: en primer lugar y debido a que el marco de definición del perfil de usuario, P_e y de los productos, $A = \{a_1, ..., a_n\}$, es multigranular, no podemos operar directamente sobre ella por lo que deberemos unificarla en un único dominio de expresión [76].

2. Calculo de la similitud entre cada producto y el perfil de usuario: una vez unificada la información calcularemos la similitud entre el perfil de usuario, P_e y cada uno de los productos, a_j de la base datos, A, mediante la medida de similitud que se presentó en la sección 5.1.1:

$$d_j = d\left(P_e, a_j\right) = \frac{1}{l}\sum_{k=1}^{l} sim\left(p_k^{*e}, v_k^{*j}\right),\ sim\left(p_k^{*e}, v_k^{*j}\right) \in V_{SIM}\left(P_e, a_j\right)$$

5.1.5. Recomendación

Una vez calculada la similitud entre el perfil de usuario y todos los productos de la base de datos, nuestro objetivo es ordenar los productos según la similitud obtenida y que están representadas en el siguiente vector de similitud

$$D = (d_1, ..., d_n)$$

Los mejores serán aquellos que mejor satisfagan las necesidades del perfil del usuario (con la similitud mayor). Así pues, dado el conjunto de productos

$$A = \{a_1, ..., a_n\}$$

el sistema recomendará al usuario un conjunto de k productos, $A^B = \{a_1, \cdots, a_k\}$, que verifica:

1. Si $a_i \in A^B$ entonces $a_i \neq a_e$.

2. $a_i \in A^B$ son los k productos con los mayores valores de similaridad, d_i.

Como puede comprobarse, la generación de recomendaciones de este modelo requiere mucha menos carga computacional. Esto hace que, este modelo sea capaz de manejar bases de datos de productos con un mayor número de productos y de características que el modelo presentado en capítulo 4.

5.1.6. Ejemplo de aplicación del modelo RLM

Al igual que con el modelo propuesto en el capítulo 4, vamos a mostrar un ejemplo general simple del funcionamiento del modelo presentado.

1. *Creación de la base de datos de productos.*

   Supongamos una base de datos de productos

$$A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

descritos por un conjunto de atributos

$$C = \{c_1, c_2, c_3, c_4\}$$

En sistemas reales estas bases de datos podrían tener almacenados miles de productos y estarán descritos por más atributos. Dado que el modelo se define en un contexto multigranular, los atributos c_1 y c_2 se valorarán usando las escala lingüística $S_{1,2}$ (ver figura 5.3) y para los atributos c_3 y c_4 en la escala lingüística $S_{3,4}$ (ver figura 5.4). Estas escalas lingüísticas se definirán mediante las siguientes funciones de pertenencia:

$s_0^{1,2} = Infimo = (0, 0, 0.125)$ $s_1^{1,2} = Muy\,bajo = (0, 0.125, 0.25)$

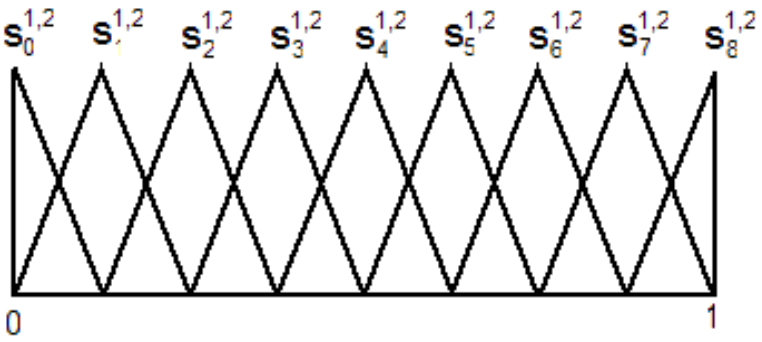
$s_2^{1,2} = Bajo = (0.125, 0.25, 0.375)$ $s_3^{1,2} = Un\,poco\,bajo = (0.25, 0.375, 0.5)$

$s_4^{1,2} = Medio = (0.375, 0.5, 0.625)$ $s_5^{1,2} = Un\,poco\,alto = (0.5, 0.625, 0.75)$

$s_6^{1,2} = Alto = (0.625, 0.75, 0.875)$ $s_7^{1,2} = Muy\,alto = (0.75, 0.875, 1)$

$s_8^{1,2} = Extremo = (0.875, 1, 1)$

Figura 5.3: Conjunto de etiquetas $S_{1,2}$ empleada en la base de datos de productos



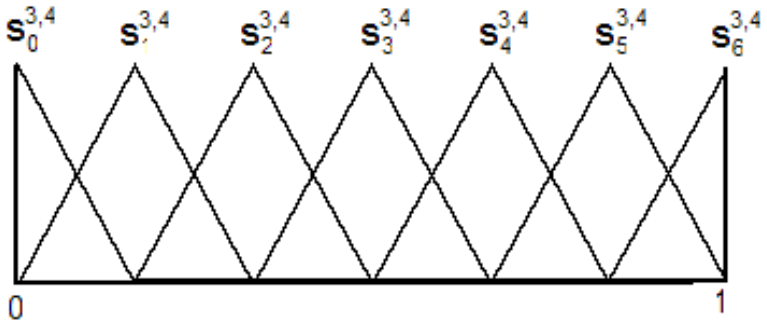
$s_0^{3,4} = Despreciable = (0, 0, 0.16)$ $s_1^{3,4} = Muy\,inferior = (0, 0.16, 0.33)$

$s_2^{3,4} = Inferior = (0.16, 0.33, 0.5)$ $s_3^{3,4} = Normal = (0, 33, 0.5, 0.66)$

$s_4^{3,4} = Elevado = (0.5, 0.66, 0.83)$ $s_5^{3,4} = Muy\,elevado = (0.66, 0.83, 1)$

$s_6^{3,4} = Considerable = (0.83, 1, 1)$

Figura 5.4: Conjunto de etiquetas $S_{3,4}$ empleada en la base de datos de productos



Las descripciones de los productos, de la vista de la base de datos utilizado para mostrar el ejemplo, la podemos ver en la tabla 5.2

Tabla 5.2: Base de datos de productos

	c_1	c_2	c_3	c_4
a_1	$s_0^{1,2}$	$s_3^{1,2}$	$s_3^{3,4}$	$s_2^{3,4}$
a_2	$s_5^{1,2}$	$s_2^{1,2}$	$s_1^{3,4}$	$s_4^{3,4}$
a_3	$s_7^{1,2}$	$s_4^{1,2}$	$s_0^{3,4}$	$s_5^{3,4}$
a_4	$s_5^{1,2}$	$s_6^{1,2}$	$s_2^{3,4}$	$s_6^{3,4}$
a_5	$s_8^{1,2}$	$s_0^{1,2}$	$s_4^{3,4}$	$s_6^{3,4}$
a_6	$s_1^{1,2}$	$s_8^{1,2}$	$s_3^{3,4}$	$s_1^{3,4}$

2. *Obtención del perfil de usuario*

   Si un usuario, u_e, desea recibir una recomendación del sistema, deberá proveer información al sistema de sus necesidades para que éste cree su perfil de usuario. Para obtener dicho perfil, se siguen los siguientes pasos:

   a) *Adquisición del ejemplo preferido por el usuario*

   El producto, a_1, es el ejemplo que el usuario dará para expresar sus necesidades actuales y a partir de éste se construye un perfil inicial del usuario:

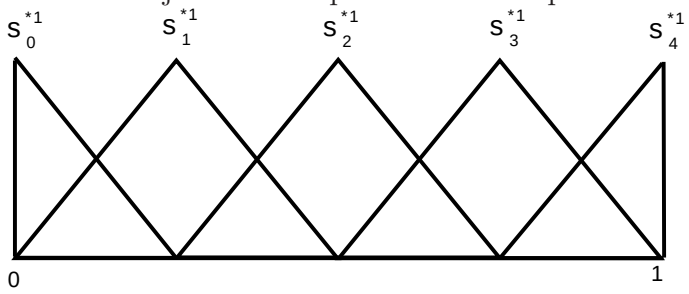
$$P_e = \left\{ s_0^{1,2}, s_3^{1,2}, s_3^{3,4}, s_2^{3,4} \right\}$$

   b) *Refinamiento ocasional de preferencias*

   Supongamos que el usuario quiere aumentar el valor de la característica, c_1, pero la escala lingüística utilizado por el sistema para describir esta característica es demasiado precisa para el nivel de conocimiento que tiene. El usuario decide utilizar la escala lingüística S_1^* que definimos a continuación (ver figura 5.5):

$$s_0^{*1} = Muy\ alto = (1, 1, 0.75) \qquad s_1^{*1} = Alto = (1, 0.75, 0.5)$$
$$s_2^{*1} = Medio = (0.75, 0.5, 0.25) \quad s_3^{*1} = Bajo = (0.5, 0.25, 0)$$
$$s_4^{*1} = Muy\ bajo = (0.25, 0, 0)$$

Figura 5.5: Conjunto de etiquetas utilizado por el usuario



El usuario le asigna el valor s_2^{*1} a la característica c_1. Por lo tanto, después de este paso su perfil de usuario será el siguiente:

$$P_e = \left\{s_2^{*1}, s_3^{1,2}, s_3^{3,4}, s_2^{3,4}\right\}$$

3. *Filtrado de productos*

El siguiente paso en nuestro ejemplo es el cálculo de la distancia entre el perfil de usuario y los productos de la base de datos de productos. Para ello debemos realizar los siguientes pasos:

a) *Unificar la información lingüística:* según las condiciones enumeradas en el capítulo 3, sección 3.4, el CBTL elegido para unificar la información será $S_{1,2}$. El perfil, P_e, unificado en $S_{1,2}$ es:

$$\begin{aligned} P_e = \ & \{(0,0,0.33,0.66,1,0.66,0.33,0,0), (0,0,0,1,0,0,0,0,0) \\ & (0,0,0.14,0.57,1,0.57,0.14,0,0), (0,0.28,0.71,0.85,0.42,0,0) \\ & ,0,0) \end{aligned}$$

Y la base de datos de productos unificada en el CBTL está en la tabla 5.3.

Tabla 5.3: Descripción de los productos en el CBTL

	c_1	c_2	c_3	c_4
a_1	$(1,0,0,0,0,0,0,0,0)$	$(0,0,0,1,0,0,0,0,0)$	$(0,0,.14,.57,1,.57,.14,0,0)$	$(0,.28,.71,.85,.42,0,0,0,0)$
a_2	$(0,0,0,0,0,1,0,0,0)$	$(0,0,1,0,0,0,0,0,0)$	$(.42,.85,.71,.28,0,0,0,0,0)$	$(0,0,0,0,.42,.85,.71,.28,0)$
a_3	$(0,0,0,0,0,0,0,1,0)$	$(0,0,0,0,1,0,0,0,0)$	$(1,.57,.14,0,0,0,0,0,0)$	$(0,0,0,0,0,.28,.71,.85,.42)$
a_4	$(0,0,0,0,0,1,0,0,0)$	$(0,0,0,0,0,0,1,0,0)$	$(.28,.71,.85,.42,0,0,0,0,0)$	$(0,0,0,0,0,0,.14,.57,1)$
a_5	$(0,0,0,0,0,0,0,0,1)$	$(1,0,0,0,0,0,0,0,0)$	$(0,0,0,0,.42,.85,.71,.28,0)$	$(0,0,0,0,0,0,.14,.57,1)$
a_6	$(0,1,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,1)$	$(0,0,.14,.57,1,.57,.14,0,0)$	$(.42,.85,.71,.28,0,0,0,0,0)$

b) *Cálculo de la distancia entre dos objetos:* en esta fase el modelo calculará la distancia entre el perfil de usuario y el resto de objetos

1) Cálculo de los valores centrales tanto de los productos de la base de datos como del perfil de usuario (ver tabla 5.4 y 5.5)

Tabla 5.4: Valores centrales de los productos

	c_1	c_2	c_3	c_4
a_1	0	3	4	2.62
a_2	5	2	1.37	5.37
a_3	7	4	1.19	7.05
a_4	5	6	2.6	7.5
a_5	8	0	5.37	7.5
a_6	1	8	4	1.37

Tabla 5.5: Valores centrales del perfil del usuario

c_1	c_2	c_3	c_4
4	3	4	2.62

Un ejemplo de cálculo de estos valores es:

$$cv_{c_1}^{P_e}(0,0,0.33,0.66,1,0.66,0.33,0,0) =$$

$$= \frac{0\cdot 0+1\cdot 0+2\cdot 0.33+3\cdot 0.66+4\cdot 1+5\cdot 0.66+6\cdot 0.33+7\cdot 0+8\cdot 0}{0+0+0.33+0.66+1+0.66+0.33+0+0} = 4$$

2) Cálculo de la distancia entre los atributos del perfil de usuario y la base de datos de productos (ver tabla 5.6):

Tabla 5.6: Distancia entre los atributos de la base de datos de productos y el perfil de usuario

	c_1	c_2	c_3	c_4
a_1	0.5	1	1	1
a_2	0.875	0.875	0.67	0.65
a_3	0.625	0.875	0.65	0.44
a_4	0.875	0.625	0.82	0.39
a_5	0.5	0.625	0.82	0.39
a_6	0.625	0.375	1	0.84

$$sim\left(p_1^{*e}, v_1^{*1}\right) = 1 - \left|\frac{cv_1^{P_e} - cv_1^{a_1}}{8}\right| = 1 - |0.5| = 0.5$$

3) El último paso de esta fase es el cálculo de la similitud entre los productos de la base de datos y el perfil de usuario obteniendo los siguientes

resultados (ver tabla 5.7).

Tabla 5.7: Distancia entre los productos y el perfil de usuario

a_1	a_2	a_3	a_4	a_5	a_6
0.87	0.76	0.64	0.67	0.58	0.71

$$d_1 = d\left(P_e, a_1\right) = \frac{0.5 + 1 + 1 + 1}{4} = 0.87$$

4. *Recomendación*

   En la última fase del modelo, se recomendará aquellos productos que más se acerquen a las necesidades del usuario. Si nos damos cuenta el producto que más se acerca al perfil de usuario es, a_1, lo cual es bastante lógico teniendo en cuenta que este fue el ejemplo que dio el usuario. Por esta razón, a_1, no es recomendado, ya que, el usuario está buscando productos similares a a_1, pero que no sean a_1. Por lo tanto, el vector ordenado que obtenemos es

   $$\left\{a_2, a_6, a_4, a_3, a_5\right\}.$$

   Si suponemos que nuestro sistema de recomendación sólo recomienda los dos productos más cercanos. Las recomendaciones recibidas por el usuario serían:

   $$a_2, a_6$$

5.2. RRPI: Sistema de recomendación basado en conocimiento con relaciones de preferencia incompletas

En nuestra búsqueda de mejorar los procesos de recomendación, en aquellas situaciones donde hay falta de información. Vamos a presentar un nuevo modelo, basado en conocimiento, que mejora el proceso de adquisición de información del usuario con respecto a los modelos anteriores.

Para ello, este modelo denominado RRPI, partirá de un conjunto reducido de ejemplos que expresan las necesidades del usuario y de una relación de preferencia incompleta sobre dichos ejemplos. A partir de esta información, construirá el perfil de usuario evitando, de esta forma, la fase de refinamiento del modelo RML.

El modelo RRPI se centrará en cumplir los siguientes objetivos:

1. Ofrecer al usuario una herramienta que permita generar su perfil de forma fiel a sus necesidades sin aportar mucha información, y que además lo guíe mejor hacia aquellos productos que le pueden interesar más. Como comentamos en la introducción, este modelo define el perfil de usuario a partir de varios ejemplos, y no de uno, y por lo tanto, es menos susceptible a lo bien han sido escogidos éstos para representar las necesidades del usuario.

2. Conseguir que el usuario no tenga que declarar de forma explícita cada uno de las características de su perfil, tal y como, ocurría con el modelo presentado en el capítulo 4, ni que este necesite refinar las características del perfil como ocurre en el modelo RML.

El modelo seguirá el siguiente esquema de funcionamiento para generar las

recomendaciones (ver figura 5.6):

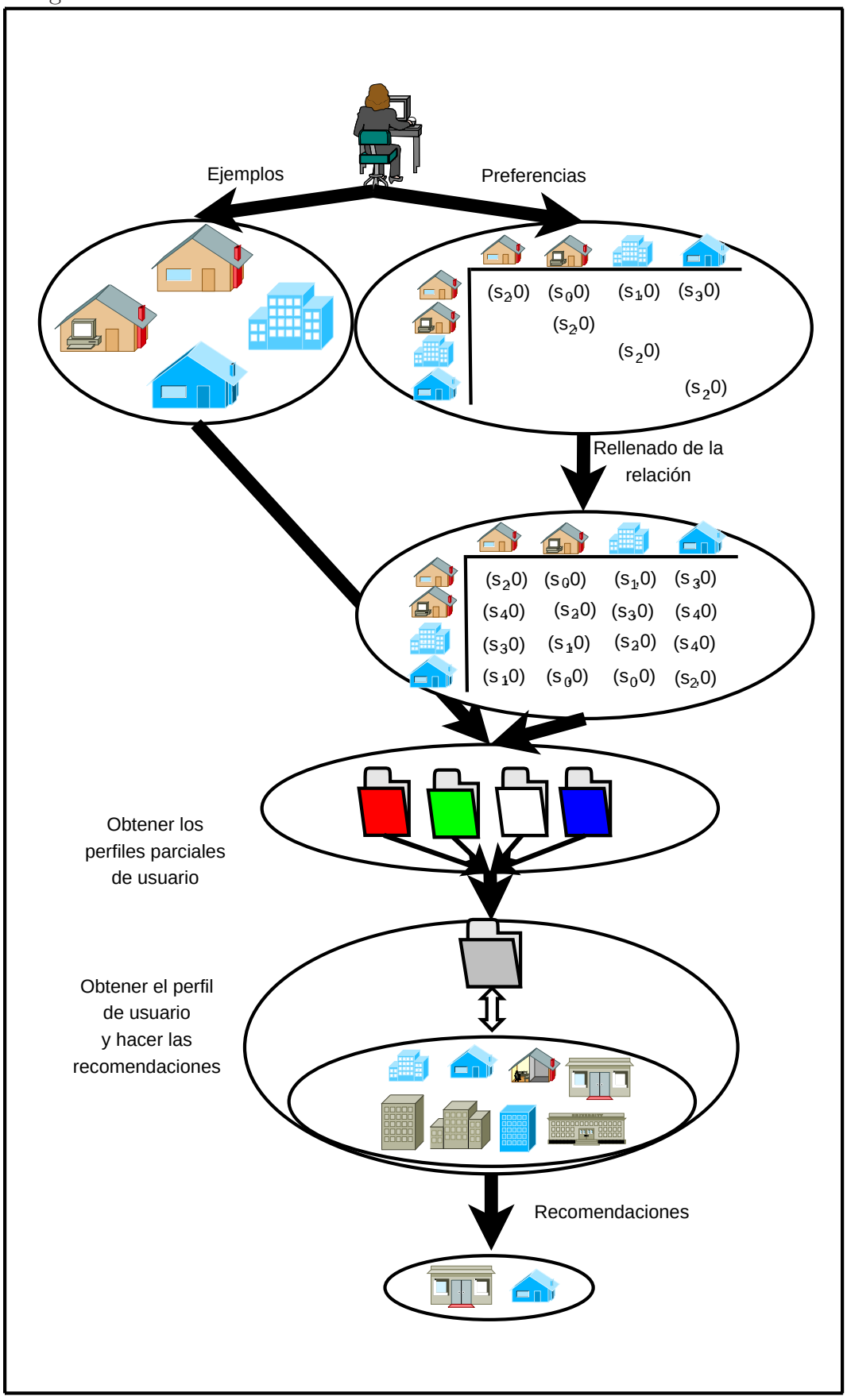
1. *Creación de la base de datos de productos.*
2. *Obtención del perfil de usuario.*
   a) *Adquisición de la información de preferencia del usuario:* Esta fase es un proceso de dos pasos:
      1) *Establecer los ejemplos favoritos y una relación de preferencia incompleta sobre ellos.*
      2) *Rellenado automático de la relación de preferencia incompleta.*
   b) *Construcción de un perfil de usuario:*
      1) *Construcción de perfiles de usuario parciales.*
      2) *Obtención del perfil de usuario.*
3. *Filtrado de productos.*
4. *Recomendación.*

A continuación, vereremos el marco de recomendacion de este modelo y explicaremos cada una de las fases que lo componen. Por último, veremos un ejemplo de cómo funciona en la sección 5.2.6.

5.2.1. Marco de recomendación.

El marco de recomendación que proponemos para el modelo RRPI, modelará la información de los usuarios mediante relaciones de preferencia. Nuestra propuesta de reducir el tiempo de obtención de dicha información nos lleva a requerir del usuario únicamente una relación de preferencia incompleta.

Figura 5.6: Modelo de sistema de recomendación basado en conocimiento



Sin embargo, para poder sugerir recomendaciones adecuadas, necesitamos la máxima información posible sobre los gustos y preferencias del usuario. Para ello, proponemos completar la relación de preferencia incompleta dada por el usuario, basándonos en la propiedad de transitividad aditiva. De esta forma, podemos obtener la máxima información posible, evitando inconsistencias, ya que, si éstas no se evitaran, las soluciones (recomendaciones) obtenidas por el modelo podrían no ser satisfactorias. Existen varios algoritmos para llevar a cabo esta tarea. En el apéndice C hemos revisado los algoritmos más utilizados en la literatura para rellenar relaciones de preferencia incompletas, y presentamos nuestra propuesta de algoritmo de rellenado de relaciones de preferencia lingüísticas, utilizado en este modelo.

Este marco de evaluación no se centra en los dominios de expresión que pueden utilizarse en el modelo, ya que, existen en la literatura herramientas para extender su uso en diferentes contextos de definición. En principio, proponemos que la información dada por el usuario, se exprese en un dominio lingüístico, y la información de los productos en un dominio numérico. Aunque en un futuro, puedan ser extendidos a otros dominios.

5.2.2. Creación de la base de datos de productos

Cada uno de los productos de la base de datos (ver tabla 5.8),

$$A = \{a_1, a_2, \ldots, a_m\},$$

a recomendar se describen por un conjunto de características

$$C = \{c_1, ..., c_t\}$$

Tabla 5.8: Base de datos

	c_1	...	c_l
x_1	v_1^1	...	v_l^1
...	...	...	...
x_m	v_1^n	...	v_l^n

y cada producto, a_i, está descrito con una valoración en cada una de estas características,

$$a_i = \left\{c_i^1, \ldots c_i^t\right\}.$$

En este caso, cada uno de estos valores podrá ser un número en el intervalo $[0, 1]$ o una etiqueta lingüística. En el ejemplo que mostraremos al final del capítulo, sólo hemos empleado valoraciones numéricas, pues nuestro objetivo principal en este modelo es mostrar cómo se construiría el perfil de usuario a partir de la información proporcionada por el usuario. Como hemos indicado, su extensión a otros dominios de expresión es simple.

Aunque la base de datos podría crearse mediante técnicas recuperación de información de forma automática, normalmente la creación de esta base de datos será semiautomática o manual, ya que, es muy probable que se requiera la supervisión de un experto o un conjunto de expertos para definir:

- ¿Qué características son importantes?
- ¿Qué valores utilizarán cada una de estas características?
- Si existe algún tipo de relación entre distintos valores
- ...

5.2.3. Obtención del perfil de usuario

Es en esta fase, se desarrollan las principales mejoras de nuestra propuesta. En el modelo RML, para construir el perfil de usuario éste debía:

1. Dar un ejemplo de las necesidades del usuario.
2. Usar la fase de refinamiento si quería modificar su perfil para adecuarlo lo mejor posible a sus necesidades.

En el modelo RRPI, nuestro objetivo es evitar esta fase de refinamiento y construir un perfil fiel a sus necesidades. Para ello, el modelo realizará los siguientes pasos:

1. Adquisición de la información de preferencia del usuario:
   a) Establecer los ejemplos favoritos y una relación de preferencia incompleta sobre ellos.
   b) Rellenado automático de la relación de preferencia incompleta.
2. Construcción de un perfil de usuario:
   a) Construcción de perfiles de usuario parciales.
   b) Obtención del perfil de usuario.

Adquisición de la información de preferencia del usuario

El objetivo de esta fase es obtener información sobre las preferencias del usuario. Primero, el usuario debe escoger algunos productos (cuatro o cinco) como ejemplo de sus preferencias, gustos o necesidades. La principal dificultad para el usuario, sería cómo explorar la base de datos de productos para encontrar estos cuatro o cinco productos, ya que, puede ser inmensa. Para facilitar la tarea de

selección, el sistema construye un conjunto de productos representativo para este usuario.

Sea $A = \{a_1, a_2, \ldots, a_m\}$ el conjunto de productos que pueden ser recomendados y cada uno de ellos esta descrito por un vector de características

$$a_i = \left\{c_i^1, \ldots c_i^t\right\}.$$

Cuando un usuario visita el sistema se le ofrecerá un subconjunto de productos

$$A_r = \{a_1^r, a_2^r, \ldots, a_{m'}^r\}\,(m' \leq m).$$

Este subconjunto debería de ser suficientemente grande como para contener productos que representen cualquier tipo de necesidad del usuario y además estos productos tienen que ser conocidos por el usuario. Sin embargo, debemos tener cuidado en no ofrecer un conjunto de productos demasiado grande ya que podría hacer desistir al usuario del uso del sistema de recomendación.

Existen varias alternativas para obtener este subconjunto, la mas fácil es la utilización de listas de productos creadas por expertos, listas de productos más vendidos,... Por ejemplo, CDNOW (www.cdnow.com) ofrece a sus usuarios listas de CDs por género creados por expertos en música o les permite consultar los CDs más vendidos en ese momento. Sería fácil, a partir de estas listas construir el subconjunto A_r. Además este subconjunto no tiene porque ser único, se le puede ofrecer al usuario un subconjunto distinto dependiendo de lo que esté buscando (no es lo mismo buscar música rock, que clásica).

Nota 5.1. No existe ninguna correlación entre conocidos y preferidos. En este subconjunto A_r que se le ofrece al usuario, éste podrá encontrar tanto productos interesantes, como productos que al usuario no le interesen. Por ejemplo, si estuviéramos recomendando hoteles, los hoteles *Hilton* serían muy conocidos pero no tienen porque ser del agrado del usuario.

i. Establecer los ejemplos favoritos y una relación de preferencia incompleta sobre ellos.

El conjunto, A_r, es mostrado al usuario para que escoja el conjunto de ejemplos que representan sus necesidades,

$$A_u = \{a_1^u, \ldots, a_n^u\}.$$

Después, el sistema le pedirá al usuario que exprese sus preferencias entre los elementos de A_u por medio de una relación de preferencia lingüística, en la cual, se utilizará la escala lingüística

$$S = \{s_0, \ldots, s_g\}.$$

El usuario solo tiene que rellenar una fila de la relación de preferencia, ya que, el modelo de recomendación obtendrá una relación de preferencia completa a partir del "algoritmo de rellenado de relaciones de preferencia incompletas con tendencia al valor de indiferencia" que hemos presentado en el apéndice C.

De esta forma, se obtienen tres ventajas:

1. El proceso de adquisición de información de preferencia es rápido y fácil: el usuario proporciona la información mínima necesaria para que, el sistema pueda empezar a generar un perfil de usuario útil para el cálculo de las recomendaciones.

2. El algoritmo de rellenado de preferencias nos permite trabajar con relaciones de preferencias lingüísticas completas.

3. Debido a que el modelo parte de un conjunto de ejemplos pequeño y una relación de preferencia, las recomendaciones son menos dependientes de lo

bien seleccionado que sean estos ejemplos, cosa que no ocurre en los sistemas de recomendación basados en conocimiento clásicos. Ya que en ellos, las recomendaciones son guiadas por un solo ejemplo escogido por el usuario, si no es el adecuado, las recomendaciones difícilmente serán acertadas. Cuando las recomendación son guiadas por varios ejemplos, es más probable que se puedan obtener buenas recomendaciones siempre y cuando tengamos algunos ejemplos que representen de forma adecuada las necesidades del usuario. Además, el uso de las relaciones de preferencia nos da la oportunidad de averiguar que ejemplos son más cercanos o más lejanos de lo que quiere realmente el usuario. A partir de esta información, podemos obtener un perfil de usuario más refinado que el obtenido en sistemas de recomendación clásicos.

ii. Rellenado automático de la relación de preferencia incompleta.

Una vez el usuario nos ha proporcionado el conjunto de ejemplos de sus necesidades y una relación de preferencia lingüística incompleta sobre ellos, para calcular el perfil de usuario necesitamos rellenar esta relación de preferencia, utilizando un algoritmo de rellenado de relaciones de prefencia incompletas.

En esta fase partiremos de una relación de preferencia similar a la siguiente:

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ ? & p_{22} & ? & ? \\ ? & ? & p_{33} & ? \\ ? & ? & ? & p_{44} \end{pmatrix}$$

Donde "?" representa valoraciones desconocidas, que algoritmo deberá estimar. Hay que señalar que, p_{ii}, por definición tomarán el valor de indiferencia, y p_{12}, p_{13},

p_{14} son las valoraciones proporcionadas por el usuario. Tras aplicar el algoritmo de rellenado de relaciones de preferencia, obtendremos la siguiente relación:

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p'_{21} & p_{22} & p'_{23} & p'_{24} \\ p'_{31} & p'_{32} & p_{33} & p'_{34} \\ p'_{41} & p'_{42} & p'_{43} & p_{44} \end{pmatrix}$$

donde cada elemento de la relación estará expresado con términos lingüísticos de S, representados mediante 2-tuplas, y en donde p'_{ij} es un elemento calculado mediante dicho algoritmo.

Construcción de un perfil de usuario

A partir de la relación de preferencia completa, se construye un perfil de usuario que el sistema utilizará para comparar las necesidades del usuario con las características de todos los productos almacenados en la base de datos de productos. El sistema calculará el perfil de usuario, partiendo de las descripciones de los productos elegidos como ejemplo de las necesidades del usuario y utilizando la relación de preferencia completada obtenida. El perfil del usuario se construye como sigue:

i. Construcción de los perfiles de usuario parciales

Partiendo de cada columna de la relación de preferencia, $(p_{1j}, p_{2j}, \ldots, p_{nj})$, se obtiene un perfil de usuario parcial que representa las preferencias del usuario con respecto al producto a_j^u. El proceso para obtener este perfil parcial, consiste en la agregación de los vectores de características de los otros productos distintos de, a_j^u. Para construir el perfil parcial utilizaremos el operador IOWA propuesto

por Yager en [213]. Con él, agregaremos el vector de características de los otros productos distintos de a_j^u, utilizando los elementos conocidos de la relación de preferencia $(p_{1j}, p_{2j}, ..., p_{nj})$ que utilizaremos como variables de inducción de orden en la agregación.

El operador IOWA se utiliza para agregar tuplas de la forma (υ_i, α_i), donde, υ_i, es conocido como la variable de inducción de orden y α_i es el valor a agregar

$$F_W\left(\langle \nu_1, a_1\rangle, ..., \langle \nu_n, a_n\rangle\right) = W^T B_v,$$

siendo $B_v = (b_1, ..., b_l)$ el resultado de ordenar el vector $\Lambda = (\alpha_1, ..., \alpha_l)$ de acuerdo a los valores de las variables de inducción de orden y W^T es un vector de pesos que cumple las siguientes condiciones:

$$W = (w_1, ..., w_l)$$

$$w_i \in [0,1]\, \forall i, \sum_{i=1}^{l} w_i = 1$$

El objetivo de este paso es calcular n perfiles parciales $\left\{pp_j = \left(c_{pp_j}^1, \ldots, c_{pp_j}^t\right)\right\}$, uno por cada producto a_j^u. El perfil parcial, pp_j, para cada producto a_j^u es un vector cuyos valores indican las preferencias del usuario de acuerdo a sus necesidades sobre el producto, a_j^u. El perfil parcial, pp_j, se obtiene agregando los vectores $\{(c_i^1, \ldots c_i^t), \forall i \neq j\}$ que describen al ítem a_i^u. Cada elemento, $c_{pp_j}^k$, se obtiene por la agregación de los $n-1$ elementos $\left\{c_i^k, \forall i \neq j\right\}$. Para llevar a cabo esta agregación usaremos el operador IOWA. Así, para cada elemento,$c_{pp_j}^k$, las variables de inducción de orden son obtenidas a partir de la columna j de la relación de preferencia $\{p_{ij}, \forall i \neq j\}$. Para cada característica aplicaremos la siguiente función:

$$c^k_{pp_j} = F_W\left(\left\langle p_{1j}, c^k_1\right\rangle, ..., \left\langle p_{nj}, c^k_n\right\rangle\right) = W^T B_v,$$

donde el vector $B_v = (b_1, \ldots b_{n-1})$ es dado por un orden de mayor a menor de los elementos del conjunto $\left\{c^k_i, \forall i \neq j\right\}$ de acuerdo con el orden inducido por las variables $(p_{1j}, \ldots, p_{nj})$ donde, p_{ij}, representa las preferencias de el ejemplo, a^u_i, sobre el ejemplo a^u_j.

Nota 5.2. Si estuviéramos operando con características lingüísticas, entonces el operador de agregación que se debe utilizar en este caso es el IOWA para modelos lingüísticos [209].

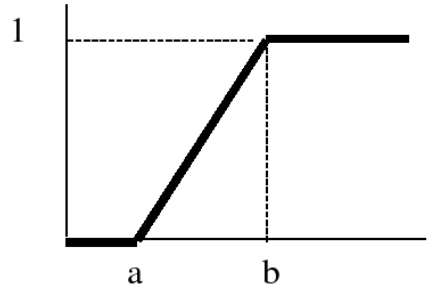
Hay diferentes métodos para construir el vector de pesos $W = (w_1, ..., w_{n-1})$. Podríamos asociarlo con un cuantificador lingüístico [210] o resolver un problema matemático tal y como se explica en [150]. En nuestro modelo, hemos optado por el método sugerido por Yager para calcular el peso de los operadores de agregación OWA mediante cuantificadores lingüísticos no decrecientes. Un cuantificador lingüístico no decreciente debe satisfacer las siguientes tres propiedades:

1. $Q(1) = 1$
2. $Q(0) = 0$
3. $Q(r_1) \geq Q(r_2)$ si $r_1 \geq r_2$

En el caso de un cuantificador no decreciente proporcional Q (ver figura 5.7), los pesos para el elemento i con $i = 1, \ldots n-1$ se calcula mediante la siguiente expresión:

$$w_i = Q\left(\frac{i}{n-1}\right) - Q\left(\frac{i-1}{n-1}\right)$$

Figura 5.7: Cuantificador proporcional no decreciente



donde la función de pertenencia de este cuantificador proporcional no decreciente Q es la siguiente:

$$Q(x) = \begin{cases} 0 & si\ x < a \\ \frac{x-a}{b-a} & si\ a \leq x \leq b \\ 1 & si\ x > b \end{cases}$$

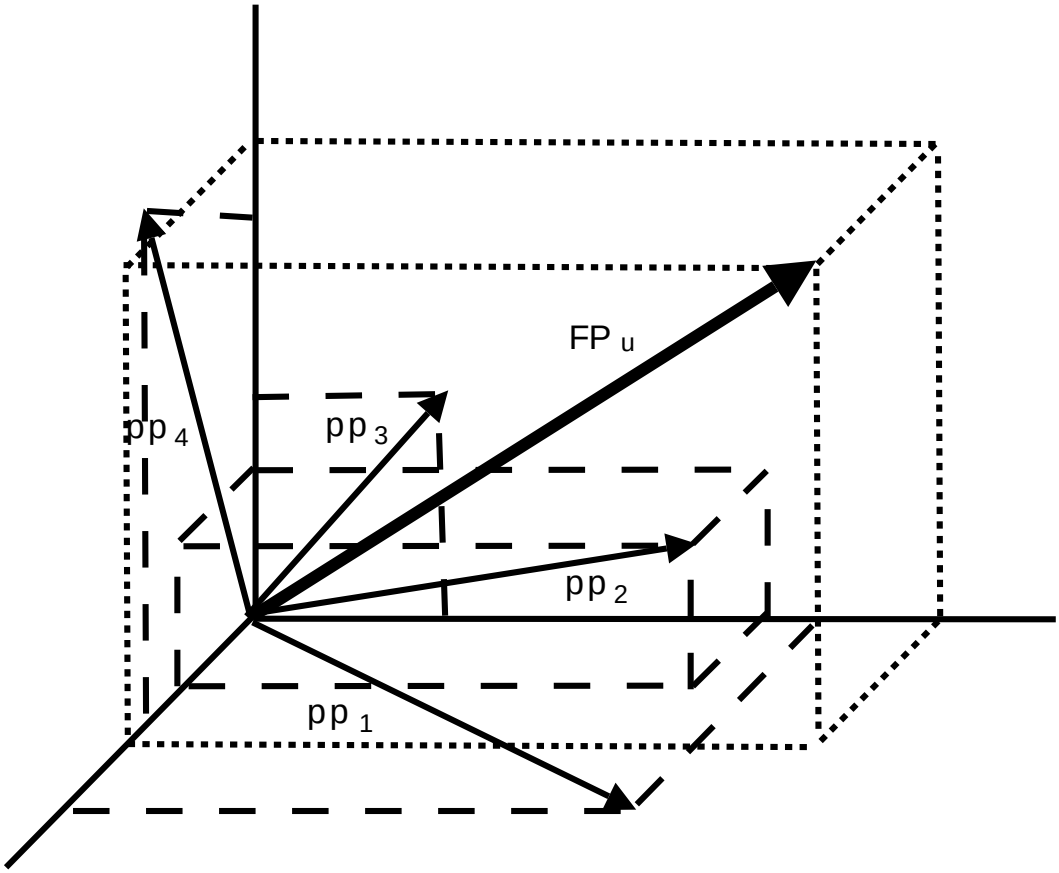
ii. Obtención del perfil de usuario

El sistema calculará el perfil final de usuario combinando los perfiles parciales (ver figura 5.8), $pp_1, \ldots, pp_n$ obtenidos anteriormente. Para este proceso de agregación, el sistema también utilizará el operador IOWA, ya que, se encuentra en una situación similar. En este paso vamos a agregar todos los perfiles parciales, $pp_j = \left(c^1_{pp_j}, \ldots, c^t_{pp_j}\right)$, obtenidos para cada producto a^r_j. Utilizaremos la siguiente función para obtener cada uno de los valores del perfil de usuario:

$$c^k_{fp} = F'_W\left(\left\langle p_1, c^k_{pp_1}\right\rangle, ..., \left\langle p_n, c^k_{pp_n}\right\rangle\right) = W'^T B'_v,$$

donde el vector, $B'_v = (b'_1, ..., b'_l)$, se obtiene mediante una ordenación creciente de los elementos del conjunto, $\left\{c^k_{pp_i}, i = 1, ..., n\right\}$, tomando como variables de ordenación las variables $(p_1, ..., p_n)$. El vector de pesos, W', se obtendrá siguiendo uno

Figura 5.8: Construcción del perfil de usuario



de los métodos explicados en el paso anterior.

Si estuviéramos operando con características lingüísticas, entonces el operador de agregación será el mismo que indicamos en la *nota 5.2*.

Para calcular los valores de ordenación p_j, utilizaremos la siguiente función que calcula el grado de dominancia de cada alternativa, p_j, sobre el resto de alternativas

$$p_j = \frac{1}{n-1} \sum_{j=0|j\neq i} (\beta_{ji}),$$

donde $\beta_{ji} = \Delta^{-1}(p_{ji})$ y p_{ji} es una 2-tupla lingüística que representa la preferencia de la alternativa j sobre la alternativa i.

La alternativa más cercana a las preferencias del usuario (la preferida) tendrá el valor más alto, la segunda más cercana el segundo más alto y así sucesivamente.

El perfil final de usuario tendrá la siguiente forma:

$$FP_u = \left\{c_{fp}^1, ..., c_{fp}^t\right\}.$$

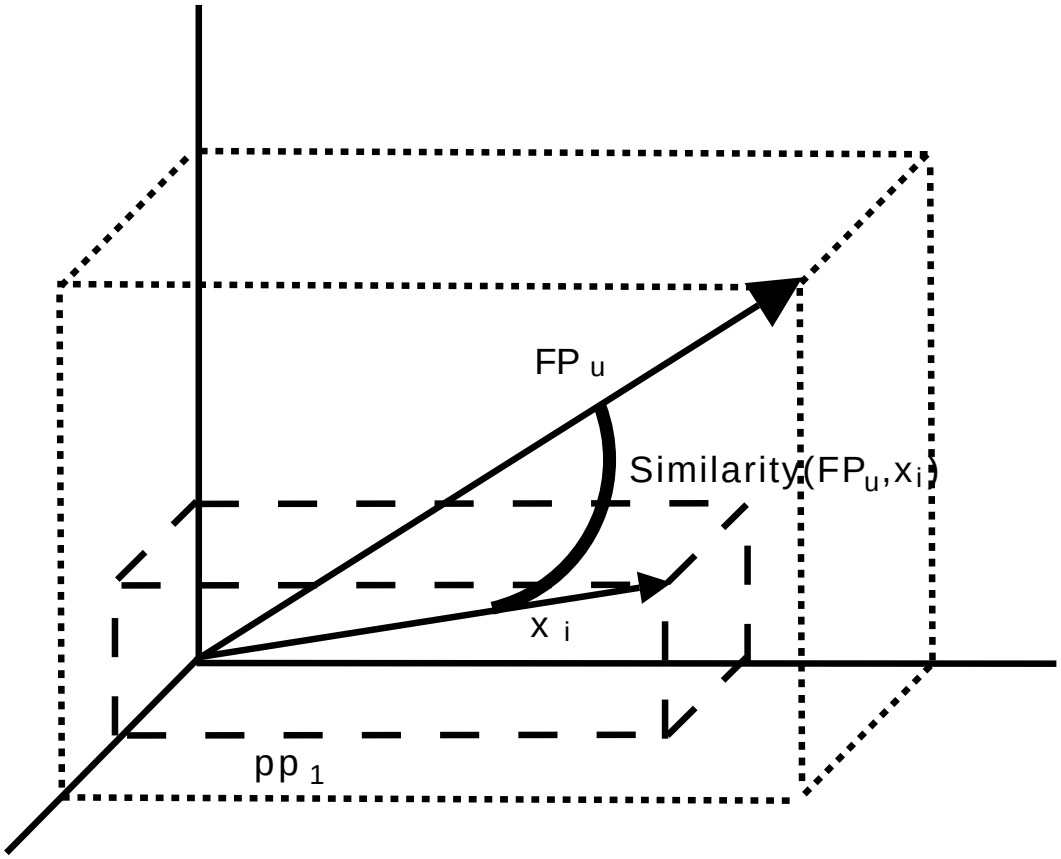
5.2.4. Filtrado de productos

Una vez que tenemos el perfil de usuario, $FP_u = \left\{c_{fp}^1, ..., c_{fp}^t\right\}$, hay que filtrar los productos de la base de datos, para buscar los productos más cercanos a las necesidades del usuario.

La base de datos de productos $A = \{a_1, a_2, \ldots, a_m\}$ contiene todos los productos donde cada a_i es descrito por un conjunto de características $a_i = \{c_i^1, \ldots c_i^t\}$.

El proceso para encontrar el producto más similar al perfil de usuario consiste en la comparación de las descripciones de estos productos con el perfil de usuario final. Para hacerlo, el sistema calculará una valoración que medirá la similitud

Figura 5.9: Similitud entre el perfil de usuario y un producto



entre un producto a_i y el perfil del usuario. Nosotros proponemos el uso de una medida basada en el coseno de vectores [106, 190, 215]. Para llevar a cabo este cálculo, trataremos el perfil de usuario y las descripciones de los productos como vectores compuestos por t características definidas en un espacio t-dimensional. La función de similitud basada en el coseno de dos vectores se define de la siguiente forma (ver figura 5.9):

Definición 5.4. *La similitud entre el perfil de usuario, FP_u, y el producto a_i se obtiene como:*

$$Similitud\,(FP_u, a_i) = \cos\left(\vec{FP_u}, \overrightarrow{a_i}\right) = \frac{\vec{FP_u} \cdot \overrightarrow{a_i}}{\|FP_u\| \cdot \|a_i\|}$$

Nota 5.3. Si alguna de las características esta evaluada mediante una escala lingüística se operaría con otras medidas de similitud como las presentadas en los modelos anteriores.

5.2.5. Recomendación

Finalmente, el sistema recomendará al usuario un conjunto de k productos, $A^B = \{a_1, \cdots, a_k\}$, que verifica:

1. Si $a_i \in A^B$ entonces $a_i \notin A_u$.
2. $Similitud\,(FP_u, a_i^r)$, $a_i \in A^B$ son los k productos con los mayores valores de similaridad.

5.2.6. Ejemplo de aplicación del modelo RRPI

A continuación, aplicaremos nuestro modelo a un problema donde un usuario desea obtener alguna recomendación. De acuerdo al esquema de funcionamiento:

1. *Creación de la base de datos de productos*

   Nuestro sistema ha almacenado una base de productos,

   $$A = \{a_1, a_2, \ldots, a_m\},$$

   donde cada producto es descrito por un conjunto de características valoradas en $[0, 1]$ (ver tabla 5.9). Estas características no son usualmente visibles para el usuario ya que son difíciles de entender. Aunque internamente el sistema las usará para calcular las recomendaciones, lo que verá el usuarios será una descripción precisa, breve y fácil de entender sobre ellos. Por ejemplo, en este ejemplo recomendaremos CDs de música (aunque podría ser aplicado a otros productos complejos como hoteles, restaurantes, etc.).

Tabla 5.9: Base de datos de productos

	Descripción									
Producto	c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
Producto a_1	1.0	0.2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Producto a_2	1.0	0.3	1.0	1.0	0	0	1.0	0	1.0	1.0
Producto a_3	0.5	0.1	0.4	0.8	1.0	1.0	1.0	0.4	0.9	0.9
Producto a_4	0.1	0.3	0.3	0.3	1.0	0	0.78	0	0.85	0.95
Producto a_5	0.74	0.37	0.26	0.41	0.39	0.86	0.22	0.05	0.62	0.62
Producto a_6	0.36	0.52	0.74	0.28	0.42	0.14	0.76	0.12	0.36	0.59
...	...	...	...	...	...	...	...	...	...	...
Producto a_8	0.55	0.012	0.81	0.88	0.45	0.97	0.13	0.60	0.88	0.49
...	...	...	...	...	...	...	...	...	...	...
Producto a_{11}	0.20	0.18	0.61	0.93	0.28	0.49	0.78	0.88	0.49	0.67
...	...	...	...	...	...	...	...	...	...	...
Producto a_{21}	0.82	0.30	0.89	0.46	0.38	0.12	0.26	0.27	0.57	0.49
Producto a_m	...	...	...	...	...	...	...	...	...	...

A continuación mostraremos los pasos que se tienen que seguir para obtener las recomendaciones.

2. *Obtención del perfil de usuario*

   Para obtenerlo, se realizarán las siguientes fases:

   a) *Adquisición de la información de preferencia del usuario*

      El sistema mostrará el conjunto de los productos más representativos

del sistema

$$A_r = \{a_1^r, a_2^r, \ldots, a_{m'}^r\}\,(m' \leq m)$$

y el usuario seleccionará los cuatro productos más cercanos a lo que el necesita (ver tabla 5.10).

Tabla 5.10: Ejemplos dados

		Descripción (oculta)									
CD	ID	c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
...	...	...	...	...	...	...	...	...	...	...	...
Spirit	1	1.0	0.2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0304	2	1.0	0.3	1.0	1.0	0	0	1.0	0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...
Cry	3	0.5	0.1	0.4	0.8	1.0	1.0	1.0	0.4	0.9	0.9
...	...	...	...	...	...	...	...	...	...	...	...
No Angel	4	0.1	0.3	0.3	0.9	1.0	0	0.78	0	0.85	0.95
...	...	...	...	...	...	...	...	...	...	...	...
Producto $x_{m'}^r$	...	...	...	...	...	...	...	...	...	...	...

1) *Establecer los ejemplos favoritos y una relación de preferencia incompleta sobre ellos.*

   En este ejemplo, el usuario ha escogido los CDs, Spirit, 0304, Cry y No Angel como ejemplo de sus necesidades. El usuario da sus preferencias sobre el conjunto de ejemplos dados. Para expresar la preferencia de la alternativa i sobre la alternativa j utilizamos la siguiente escala lingüística:

$$S = \{s_0 = VL, s_1 = L, s_2 = I, s_3 = H, s_4 = VH\}$$

donde VL, L, I, H, VH representan "Muy bajo", "Bajo", "Indiferente" y "Alto" y "Muy alto" respectivamente.

En nuestro caso, el usuario proporciona las preferencias del primer productos sobre todos los demás. Expresa su opinión por medio de términos lingüísticos que se transformarán en 2-tuplas y se obtendrá la siguiente matriz de preferencia:

$$P = \begin{pmatrix} (s_2,0) & (s_0,0) & (s_1,0) & (s_3,0) \\ & (s_2,0) & & \\ & & (s_2,0) & \\ & & & (s_2,0) \end{pmatrix}$$

2) *Rellenado automático de la relación de preferencia incompleta.*

El sistema rellenará la relación de preferencia utilizando el "algoritmo de rellenado de relaciones de preferencia con tendencia al valor de indiferencia" y obtendremos la siguiente relación de preferencia completa:

$$P'' = \begin{pmatrix} (s_2,0) & (s_0,0) & (s_1,0) & (s_3,0) \\ (s_3,0) & (s_2,0) & (s_3,0) & (s_4,0) \\ (s_3,0) & (s_1,0) & (s_2,0) & (s_4,0) \\ (s_2,0) & (s_0,0) & (s_0,0) & (s_2,0) \end{pmatrix}$$

b) *Construcción de un perfil de usuario*

El objetivo de esta fase es obtener el perfil de usuario final. Para ello:

1) *Construcción de los perfiles de usuario parciales*

Lo primero que hay que hacer es calcular los vectores de pesos,

W y W'. Para obtener estos pesos, utilizaremos la fórmula que aparece en la sección 5.2.3 con $a = 0$ y $b = 0.5$. Dependiendo de las características de cada problema, se podrá utilizar un valor de a y b distinto. El primer vector es $W = (0.67, 0.33, 0)$ y el segundo $W = (0.5, 0.5, 0, 0)$.

Con esos pesos y utilizando la relación de preferencia que nos ha dado el usuario, obtendremos los perfiles parciales del usuario. Por ejemplo, para obtener el primer valor del perfil parcial relacionado con el primer ejemplo tenemos que hacer el siguiente cálculo:

$$c^1_{pp_1} = F_W\left(\langle(s_3, 0), 1\rangle, \langle(s_3, 0), 0.5\rangle, \langle(s_2, 0), 0.1\rangle\right) =$$

$$= (0.67, 0.33, 0) \begin{pmatrix} 1 \\ 0.5 \\ 0.1 \end{pmatrix} = 0.67 + 0.33 \cdot 0.5 = 0.83$$

Los perfiles parciales que obtenemos se pueden ver en la tabla 5.11.

Tabla 5.11: Perfiles parciales

	Descripción									
Perfil parcial	c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
pp_1	0.83	0.23	0.8	0.93	0.33	0.33	1	0.13	0.97	0.97
pp_2	0.67	0.13	0.6	0.87	1	1	1	0.6	0.93	0.93
pp_3	1	0.27	1	1	0.33	0.33	1	0.33	1	1
pp_4	0.83	0.23	0.8	0.93	0.33	0.33	1	0.13	0.97	0.97

2) *Obtención del perfil de usuario*

A continuación, se agregan los perfiles parciales utilizando el vector

de pesos W'. Las variables de inducción, (p_1, p_2, p_3, p_4), se obtienen a partir de la relación de preferencia tal y como se indicó en la sección 5.2.3 de la siguiente manera:

$$p_1 = \frac{1}{4-1} \sum_{j=0|j\neq 1}^{n} \beta_{j1} = \frac{1}{3}(3+3+2) = 2.66$$

$$p_2 = \frac{1}{4-1} \sum_{j=0|j\neq 2}^{n} \beta_{j2} = \frac{1}{3}(0+1+0) = 0.33$$

$$p_3 = \frac{1}{4-1} \sum_{j=0|j\neq 3}^{n} \beta_{j3} = \frac{1}{3}(1+3+0) = 1.33$$

$$p_4 = \frac{1}{4-1} \sum_{j=0|j\neq 4}^{n} \beta_{j4} = \frac{1}{3}(3+4+4) = 3.66$$

Para calcular el primer valor del perfil de usuario realizaremos los siguientes cálculos:

$$c^1_{fp} = F'_W\left(\langle 2.54, 0.83\rangle, \langle 0.33, 0.67\rangle, \langle 1.61, 1\rangle, \langle 3.39, 0.83\rangle\right) =$$

$$= (0.5, 0.5, 0, 0) \begin{pmatrix} 0.83 \\ 0.83 \\ 1 \\ 0.67 \end{pmatrix} =$$

$$= 0.83 \cdot 0.5 + 0.83 \cdot 0.5 = 0.83$$

Después de calcular todos los valores, obtendremos el perfil de usuario que se puede ver en la tabla 5.12.

Tabla 5.12: Perfil de usuario FP_u

FP_u									
c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
0.83	0.23	0.8	0.93	0.33	0.33	1	0.13	0.97	0.97

3. *Filtrado de productos*

   En esta fase, el modelo compara el perfil de usuario con cada producto y recomienda aquellos productos de la base de datos. Los resultados obtenidos se pueden ver en la tabla 5.13.

Tabla 5.13: Recomendaciones

Producto	similitud
Producto x_2	0.97
Producto x_1	0.90
Producto x_{21}	0.89
...	...

4. *Recomendación*

   El sistema recomendará los k productos más cercanos al perfil de usuario de acuerdo con las reglas introducidas en la sección 5.2.5 (ver tabla 5.13).

   Si que $k = 3$, entonces los productos que el sistema recomienda son $X^B = \{x_2, x_1, x_{21}\}$.

5.3. Conclusiones

En esta capítulo, hemos presentado dos modelos de sistemas de recomendación basados en conocimiento el RML y el RPPI, cuyos objetivos son:

1. Facilitar el proceso de adquisición de las preferencias del usuario.

2. Reducir la carga computacional de los modelos.

El primero de ellos, el RML, es un sistema de recomendación basado en conocimiento, en el cual, el perfil de usuario se construye a partir de un producto elegido por el usuario como ejemplo de sus necesidades. Este modelo de recomendación explota la información disponible sobre el usuario empleando técnicas de razonamiento basado en casos. Además, este modelo de recomendación ofrece un contexto de trabajo más adecuado que los clásicos debido a que maneja información lingüística para modelar la información sobre los gustos, necesidades u opiniones de los usuarios. También permite que éstos se expresen con escalas lingüísticas más adecuadas al grado de conocimiento que tengan sobre la característica. Además, emplea una medida de distancia que mejora la precisión de los resultados y reduce de forma significativa la carga computacional de la fase de recomendación, permitiendo que se puedan utilizar con base de datos de productos de mayor tamaño con respecto al modelo presentado en el capítulo 4.

La segunda propuesta, RRPI, presenta una serie de mejoras sobre el modelo anterior y sobre otros sistemas de recomendación basados en conocimiento clásicos. Primero, evita la necesidad de una fase de refinamiento del perfil de usuario. Esta fase suele ser muy difícil de diseñar, implementar y adaptar a los usuarios, pudiendo presentar limitaciones que den lugar a recomendaciones no satisfactorias. Pero quizás, la ventaja más importante que presenta, RRPI, es la de construir un

perfil más preciso que en los modelos anteriores, sin que el usuario tenga que aportar mucha más información o tenga que tener un grado de conocimiento elevado de los productos, ya que, el perfil de usuario es construido a partir de varios ejemplos y de una relación de preferencia incompleta sobre dichos ejemplos. A partir de esta información se construirá un perfil de usuario, sin que esté tenga la necesidad de refinarlo, y se generarán las recomendaciones.

Capítulo 6

REJA. Sistema de Recomendación de Restaurantes de Jaén

En este capítulo, presentamos la implementación de un sistema de recomendación para restaurantes de la provincia de Jaén.

Este sistema de recomendación se ha implementado dentro del proyecto sistema de recomendación de restaurantes basado en lógica difusa REJA (REstaurantes de JAén) de la Junta de Andalucía (JA031/06) y gracias a la beca de la Consejería de Comercio, Turismo y Deporte de la Junta de Andalucía, publicada en la Resolución del 21 de Marzo de 2006 y concedida en la Resolución 16 de agosto 2006.

El sistema que presentamos, es un sistema de recomendación híbrido compuesto de dos sistemas de recomendación, uno colaborativo, y uno basado en conocimiento. El basado en conocimiento es una implementación del modelo RRPI que vimos en el capítulo 5.

Nuestro objetivo fue crear un sistema de recomendación que se pudiera aplicar en situaciones en donde a los usuarios le gustaría recibir recomendaciones, pero en donde, debido a las características del problema, no se pueden aplicar técnicas

clásicas de recomendación. Por ejemplo, es muy dificil de implementar sistemas de recomendación de restaurantes, lugares turísticos o sitios de ocio... ya que, presentan problemas como los que enumeramos a continuación:

1. Muchos de los usuarios que interaccionán con el sistema, son usuarios casuales que nunca han usado el sistema de recomendación o que lo han usado de forma esporádica y no piensan utilizarlo de forma habitual. En el capítulo 2 vimos que las técnicas clásicas de recomendación sufren del problema del *nuevo usuario* y por lo tanto no son capaces de generar recomendaciones cuando se encontraban en estas situaciones.

2. Un número importante de usuarios tendrá un conocimiento basado en expectativas sobre el servicio o producto que quieren recibir y es muy probable que no sepan expresar de forma clara y precisa las características del tipo de restaurante que desean visitar.

3. Es habitual que en este tipo de situaciones, existan usuarios que quieran recibir recomendaciones puntuales que no tengan nada que ver con lo que han hecho en el pasado. Por lo tanto, en estos casos, la información histórica no será relevante y no debería ser utilizada en la generación de recomendaciones. Por ejemplo, si un cliente desea celebrar un cumpleaños en un restaurante, es muy probable que sea un hecho puntual y que no quiera que se use la información de los restaurantes que le han gustado en el pasado para generar estas recomendaciones.

En este capítulo, presentamos el funcionamiento de este sistema. En la sección 6.1 daremos una breve descripción de los sistemas de recomendacion que forman

parte de REJA, en la sección 6.2 veremos el funcionamiento del interfaz utilizado en REJA y por último las conclusiones.

6.1. Esquema de hibridación

En este sistema de recomendación, se ha empleado la conmutación como mecanismo de hibridación [31]. Este tipo de sistemas híbridos implementan varias técnicas de recomendación, y dependiendo de la situación en la que se encuentren, emplean una u otra para la generación de recomendaciones. Este tipo de hibridación presenta las siguientes ventajas:

- Es bastante fácil de implementar, ya que su funcionamiento es simple, ya que según la situación emplea una técnica u otra.
- Como las técnicas de recomendación son independientes unas de otras, no hay que alterarlas. Su implementación es la misma que si se quisiera utilizar de forma independiente.

Decidimos emplear este tipo de hibridación, ya que, después de estudiar las distintas alternativas, nos pareció una técnica adecuada a nuestras necesidades y que podía ofrecer buenos resultados.

Para que un usuario pueda recibir alguna recomendación, éste debe estar registrado en el sistema y se tiene que idenficar en el sistema. Dependiendo de cómo desee recibir las recomendaciones, y de la información que aporte y quiere que se utilice, se seleccionará una u otra técnica de recomendación:

- Así, se empleará la técnica colaborativa, si el usuario desea que se busquen restaurantes, basándose en la información histórica que el sistema tiene sobre aquellos que el usuario ha visitado.

- Se empleará una búsqueda basada en conocimiento, si éste quiere que las recomendaciones se basen en un ejemplo de un restaurante al que querría ir y no, en la información histórica de dicho usuario. Este sistema de recomendación está basado en el modelo RRPI presentado capítulo 5.

A continuación, explicaremos los módulos que de los que se componen esta hibridación

6.1.1. Módulo colaborativo

Este es el primer módulo de recomendación que vamos a revisar en REJA. El funcionamiento de este módulo es el siguiente (ver igura 6.1):

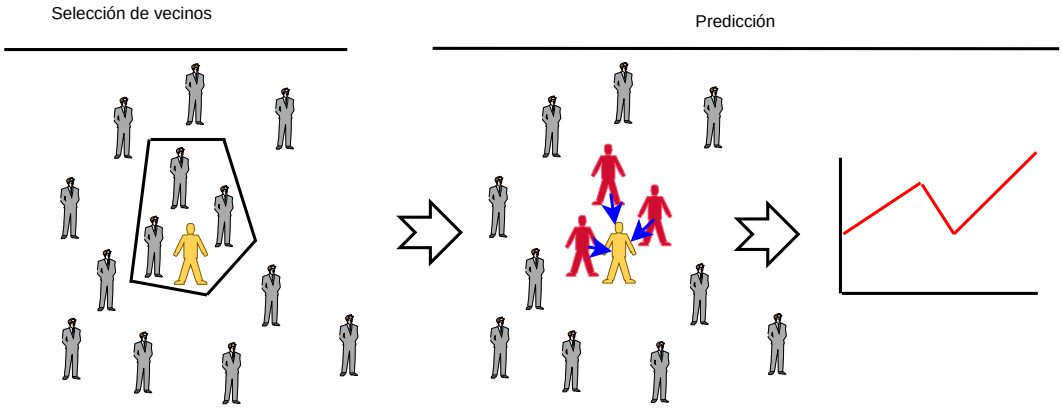
1. El sistema guarda un perfil de cada usuario con sus evaluaciones sobre los restaurantes.

2. Se mide el grado de similitud entre los distintos usuarios del sistema en base a sus perfiles y se crean grupos de usuarios con características afines.

3. El sistema usará toda la información obtenida en las fases anteriores para realizar las recomendaciones. A cada usuario, le recomendará restaurantes que no haya evaluado y que lo hayan sido de manera positiva por el resto de miembros de su grupo.

Como indicamos en el capítulo 2, este tipo de sistemas de recomendación no tienen en cuenta el contenido y las características de los restaurantes que recomiendan, sino que buscan usuarios con gustos similares al usuario actual. Este módulo generará las recomendaciones de restaurantes recomendando aquellos restaurantes que el usuario no haya visto, pero que, le hayan gustado al grupo de usuarios que son afines a él.

Figura 6.1: Esquema de funcionamiento de un sistema de recomendación colaborativo



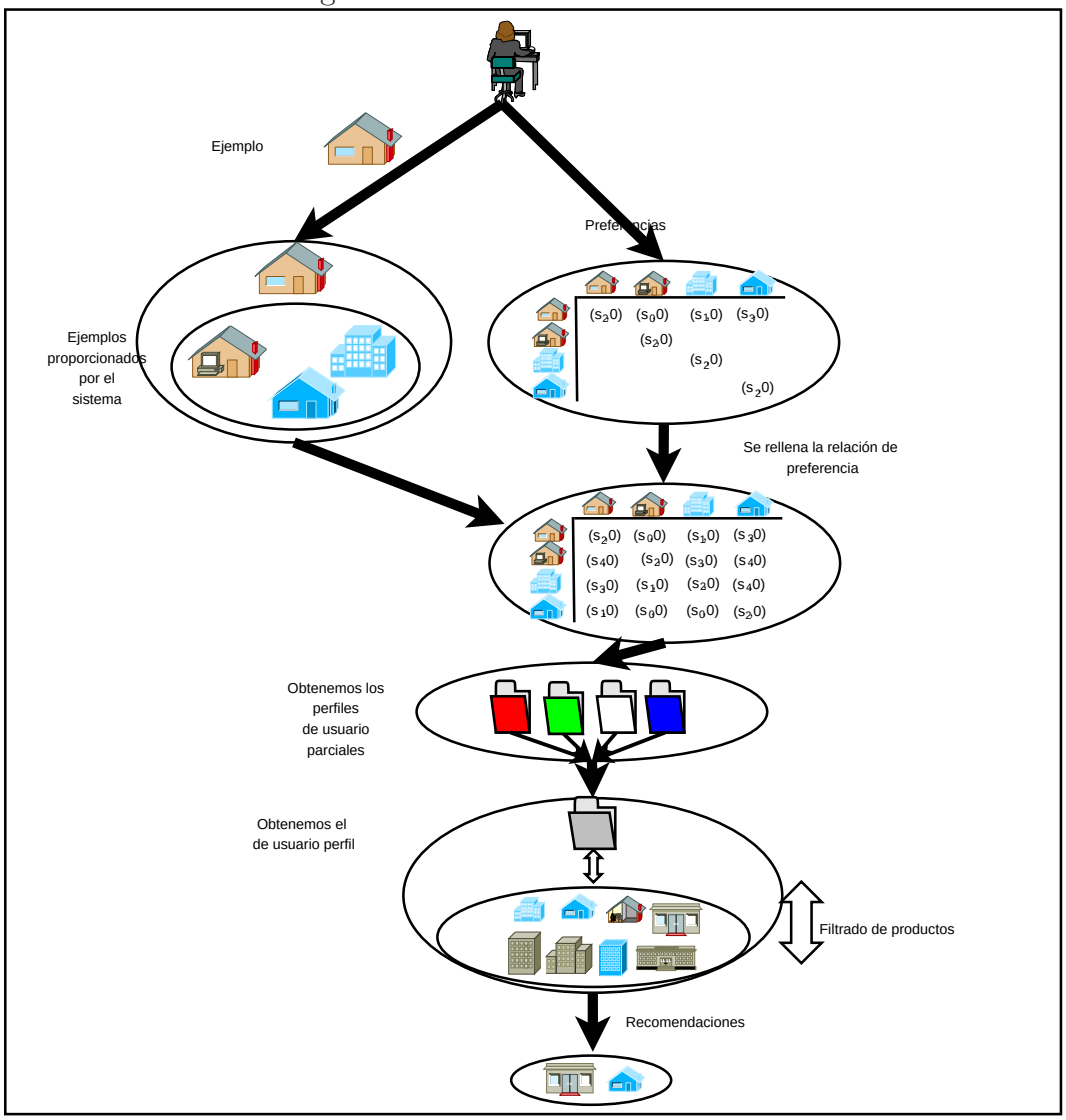
En REJA no se ha llegado a implementar ningún algoritmo de filtrado colaborativo, sino que se ha utilizado un motor de búsqueda colaborativa llamada CoFE (http://eecs.oregonstate.edu/iis/CoFE/), que trabaja sobre los datos que se le han proporcionado. Debido a que hay accesibles gran cantidad de sistemas de recomendación colaborativos en la literatura, y a que la implementación realizada es una implementación estándar basada en CoFE. No vamos a hacer una presentación en detalle del funcionamiento ya que puede encontrarse en el capítulo 2, donde vimos este tipo de sistemas de recomendación en profundidad.

6.1.2. Módulo basado en conocimiento RRPI

Este módulo está basado en el modelo de recomendación RRPI presentado en el capítulo 5. Las fases de este modelo las podemos ver en la figura figura 6.2.

La única diferencia significativa, con respecto al modelo presentado en el capítulo 5, es que mientras en el modelo teórico era el usuario el que escogía los cuatro productos ejemplo, en REJA el usuario escoge un restaurante que represente sus necesidades, y el sistema, escoge los otros tres. Estos tres restaurantes serán los dos últimos valorados por el usuario y uno que no se parezca al escogido

Figura 6.2: Modelo RRPI en REJA



por el usuario, pero bien conocido.

6.2. Aplicación

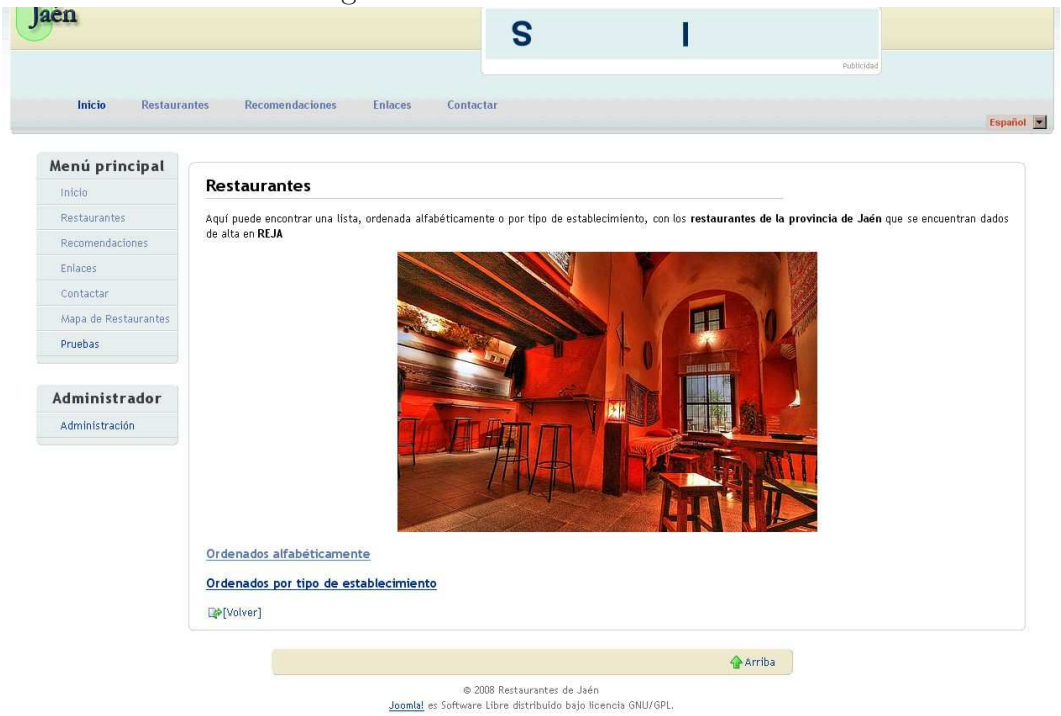
REJA es un sitio web con información sobre restaurantes que además implementa un sistema de recomendación para restaurantes de la provincia de Jaén.

A continuación, explicaremos brevemente el funcionamiento de REJA. La primera vez que conectemos con el sitio web que mantiene REJA veremos una pantalla como la que se puede ver en la figura 6.3.

Figura 6.3: REJA. Pantalla de presentación

Para obtener información sobre un restaurante lo único que tenemos hacer es pulsar sobre la opción *Restaurantes* que aparece a la izquierda de la pantalla principal. Una vez pulsada aparecerá una pantalla (ver figura 6.4) que nos permitir elegir los restaurantes de forma alfabética o por tipo de establecimiento.

Figura 6.4: REJA. Restaurantes



Una vez escogida cualquiera de las dos opciones, elegiremos un restaurante y aparecerá una pantalla con la información disponible sobre él (ver figura 6.5)

Figura 6.5: REJA. Información sobre un restaurante

Además, REJA también permite georeferenciar cualquier restaurante (ver figura 6.6). Con está característica REJA nos ofrece entre otras cosas, la posibilidad de mostrar en un mapa la localización de un restaurante, conocer otros lugares de interés cercanos a él, o el cálculo de que ruta se debe de seguir para llegar a él.

Figura 6.6: REJA. Georeferenciación

Como hemos explicado en este capítulo, REJA está compuesto por dos módulos, en donde, se implementa dos modelos de recomendación distintos, que serán utilizados mediante un mecanismo de hibridación por conmutación. En las siguientes subsecciones explicamos qué pasos seguimos para utilizar cada uno de los módulos.

Módulo Colaborativo

Para utilizar este módulo es necesario que el usuario este registrado en el sistema, se identifique, y debe de haber proporcionado suficiente información sobre los restaurantes que conoce, para que el módulo colaborativo pueda generar recomendaciones precisas.

Para identificarse tiene que introducir un nombre de usuario y una contraseña en las cajas de texto que aparecen a la derecha de la pantalla principal (figura 6.3). Este nombre de usuario y esta contraseña tienen que haber sido suministrados por los administradores del sistema.

Figura 6.7: REJA. Pantalla principal una vez identificado

Una vez identificados aparecerá una pantalla de bienvenida similar a la que vemos en la figura 6.7, en la cual podemos ver, que se le ofrecen dos opciones:

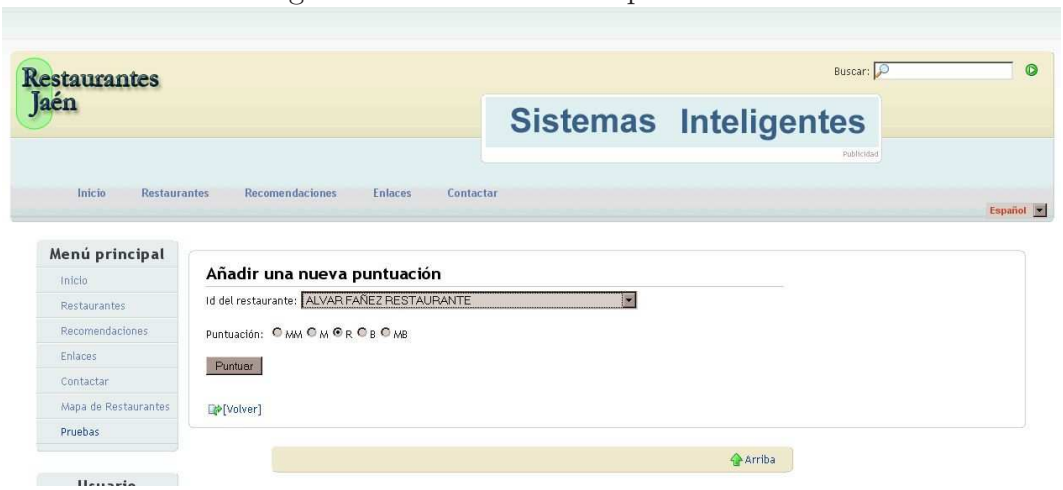
1. *Realizar una nueva puntuación:* en donde podremos proporcionar más información sobre los restaurantes que ha visitado y que opinión tiene sobre

ellos.

2. *Obtener recomendaciones:* en la cual recibiremos recomendaciones basadas en estas opiniones.

Si pulsa sobre la primera opción, *Realizar una nueva puntuación* podremos añadir una puntuación sobre un restaurante en el que hemos estado (ver figura 6.8). Para ello, se seleccionara el restaurante que queremos puntuar y luego le daremos la puntuación que deseemos: MM: Muy Mal, M: Mal, R: Regular, B: Bueno, MB: Muy Bueno. Esta información será utilizada para el módulo colaborativo para la generación de recomendaciones

Figura 6.8: REJA. Añadir puntuaciones

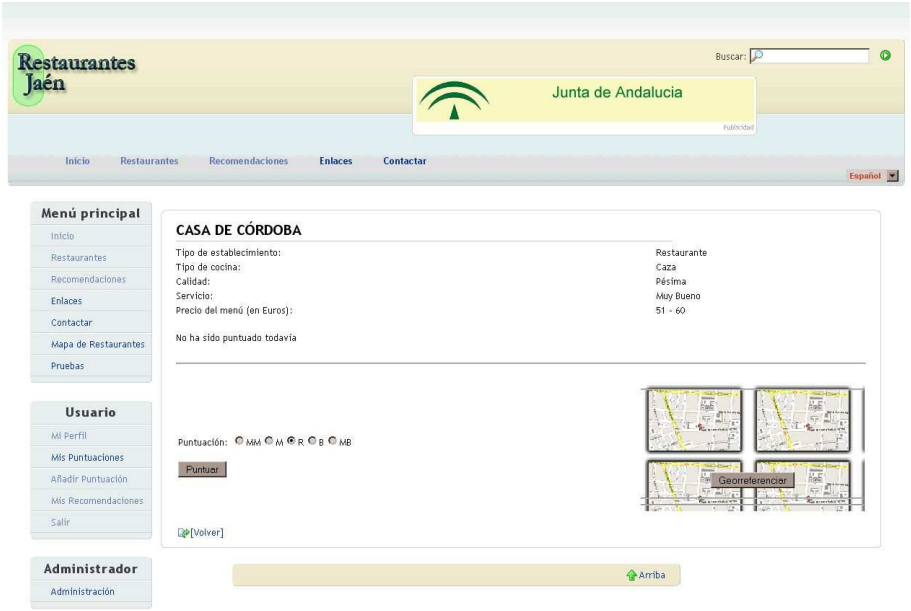


Una vez introducida toda la información necesaria, desde esta pantalla de bienvenida, es posible recibir recomendaciones colaborativas pulsando sobre la opción *Obtener recomendaciones* y aparecerá una pantalla similar a la que vemos en la figura 6.9.

Figura 6.9: REJA. Recomendaciones módulo colaborativo

Si se pulsa sobre una de las recomendaciones aparecerá otro pantalla (ver figura 6.10) en donde podremos ver información sobre este restaurante y en donde además podremos puntuarlo.

Figura 6.10: REJA. Recomendación hecha con el módulo colaborativo



Además REJA ofrece otra opción relacionada con el módulo colaborativo. La opción *Mis puntuaciones*, que podemos encontrar a la izquierda de la pantalla de bienvenida. Si pulsamos sobre ella, veremos el perfil de usuario utilizado por el módulo colaborativo, es decir, todas las puntuaciones que ha aportado el usuario sobre los restaurantes que conoce o ha visitado (ver figura 6.11).

Figura 6.11: REJA. Perfil de usuario del módulo colaborativo

Módulo RRPI

Para recibir las recomendación desde el módulo RRPI, desde la pantalla de bienvenida que aparece una vez el usuario se ha idenficado, se pulsará sobre la opción, *Recomendaciones*, que se encuentra a la izquierda de la pantalla principal. Una vez hecho ésto, veremos la primera pantalla del módulo RRPI (ver figura 6.12), en donde el usuario deberá elegir un restaurante parecido al que busca.

Figura 6.12: REJA. Recomendaciones identificado

Una vez el usuario ha elegido este restaurante ejemplo, el sistema le ofrecerá otros tres restaurantes que deberá comparar con el elegido (ver figura 6.14). Para ello, nos dirá la preferencia de este restaurante con los mostrados por el sistema utilizando la escala lingüística. En esta escala tenemos los siguientes valores lingüísticos: mucho mejor, mejor, igual, peor y mucho peor. En la figura 6.13 podemos ver en que fases del modelo de recomendación nos encontramos.

Figura 6.13: Modelo RRPI en REJA. El usuario da un ejemplo y una relación de preferencia sobre dicho ejemplo y otros tres restaurantes

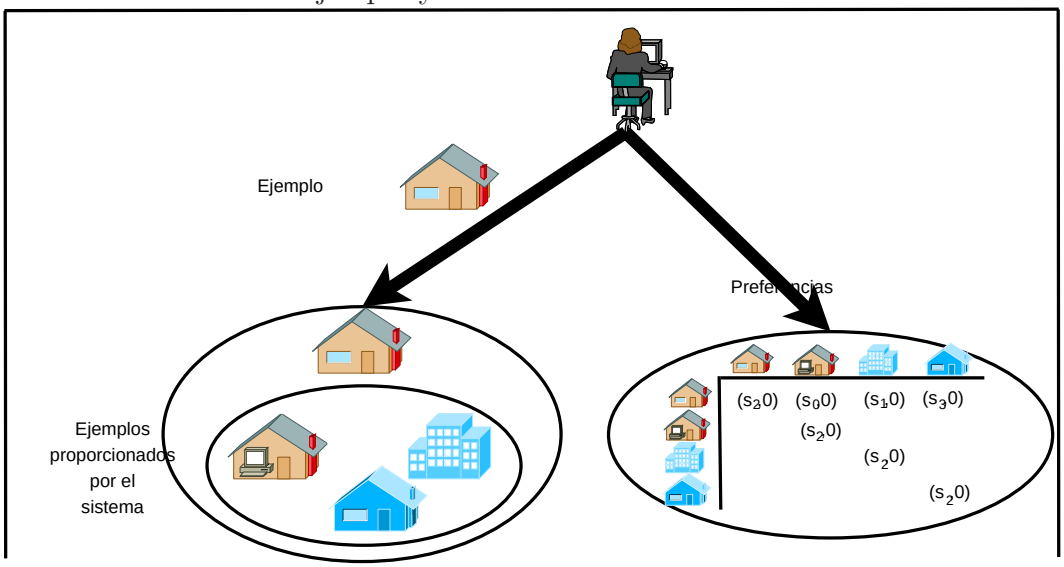
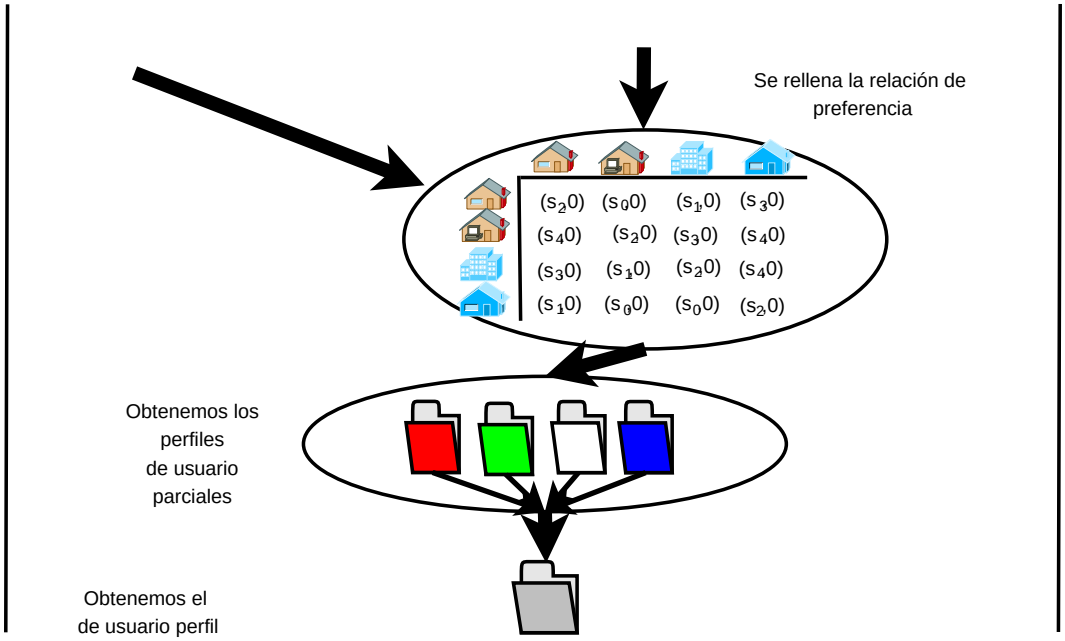


Figura 6.14: REJA. Aportando información de preferencia módulo RRPI

A partir de esta información obtendrá una relación de preferencia completa y posteriormente, el perfil de usuario (ver figura 6.15).

Figura 6.15: Modelo RRPI en REJA. Se obtiene el perfil de usuario



Con este perfil generará las recomendaciones (ver figura 6.16) y devolverá información similar a lo que vemos en la figura 6.17.

Figura 6.16: Modelo RRPI en REJA. Recomendación

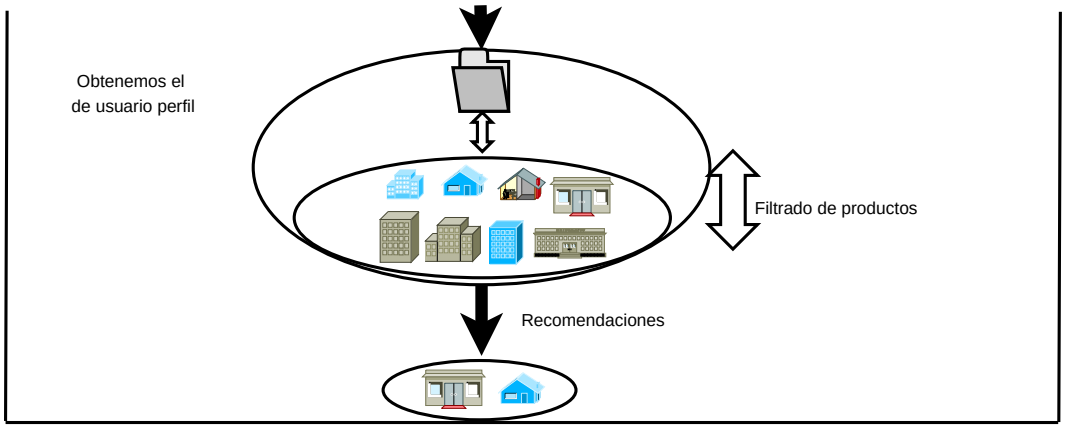
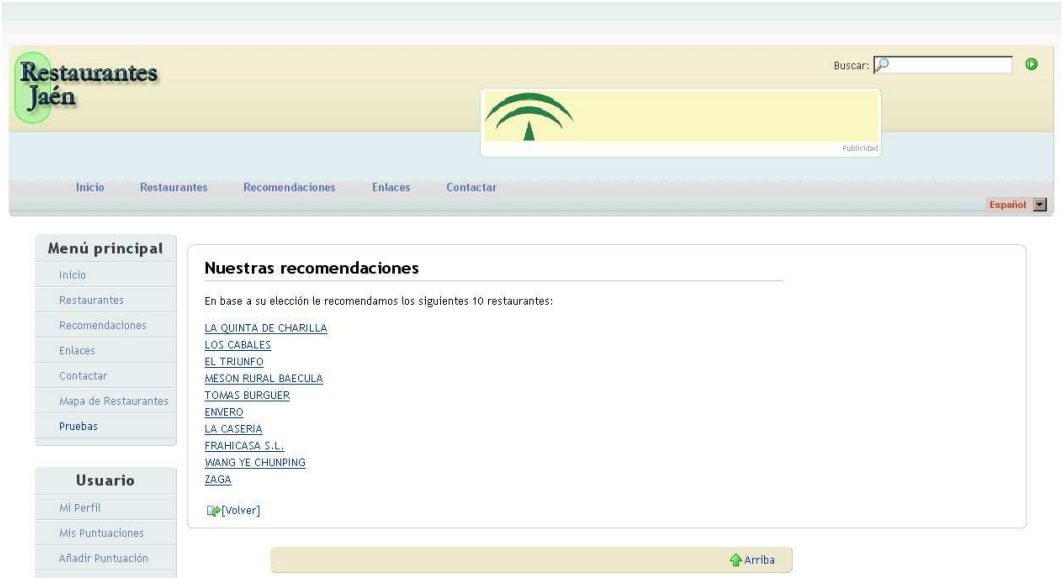


Figura 6.17: REJA. Recomendaciones obtenidas



Como ocurría en los anteriores módulos, si pulsa sobre el nombre del restaurante recomendado, el sistema mostrará la información que tenga sobre éste (ver figura 6.18).

Figura 6.18: REJA. Información sobre un restaurante

6.3. Conclusiones

En este capítulo, se ha presentado la implementación de un sistema de recomendación híbrido para restaurantes de la provincia de Jaén. Se ha empleado la hibridación por conmutación y los modelos de recomendación que componen este sistema se han dividido en dos módulos:

1. Un módulo colaborativo

2. Un módulo basado en conocimiento, que se basa en el modelo RRPI presentado en el capítulo 5.

Al principio de esta memoria establecimos como objetivo final el desarrollo de un prototipo para un sistema de recomendación.

En este capítulo hemos presentado dicho prototipo de un sistema de recomendación híbrido de restaurantes de la provincia de Jaén. Las principales aportaciones y características del prototipo presentado será:

- Ofrece una mayor flexibilidad que los sistemas clásicos, ya que, permite a los usuarios que expresan sus preferencias mediante valoraciones lingüísticas.

- Mejora los procesos de adquisición de información para la construcción de perfiles de usuario. El módulo basado en conocimiento partirá de una relación de preferencia incompleta sobre un conjunto de restaurantes para construir el perfil de usuario, ya que ésta puede ser reconstruida mediante procesos automáticos.

- Es capaz de realizar recomendaciones aún cuando no existe la información histórica o ésta no se tiene que utilizar. Aunque el sistema incluye un módulo

colaborativo, su uso no es ni obligatorio ni necesario para recibir recomendaciones. En el sistema se ha incluido un módulo basado en contenido para generar recomendaciones cuando no se quiera o no se pueda utilizar esta información histórica.

- Es un prototipo que genera recomendaciones en el sector turístico, facilitando así, la propagación de este tipo de herramientas en el mismo. Éstas se habían visto limitadas por diversas dificultades propias de las características de los productos del sector.

Conclusiones y Trabajos Futuros

A continuación revisaremos cuales han sido las principales propuestas y los resultados obtenidos a lo largo de esta memoria. Posteriormente presentamos las líneas de investigación y trabajos futuros que nos planteamos a partir de estos resultados.

Propuestas y resultados obtenidos

Los sistemas de recomendación ofrecen recomendaciones personalizadas a sus usuarios, haciendo que el proceso de compra en e-shops sea más ameno, rápido y personal. El objetivo de estos sistemas, es ofrecer a los usuarios los productos que más les interese o que más se adecuen a sus gustos o preferencias. Para aprender los gustos o preferencias de los usuarios muchos de estos sistemas utilizan la información histórica del usuario, que productos ha comprado o ha evaluado. Aunque este tipo de sistemas ha sido empleado en muchas situaciones con éxito, ya que, esta información estaba disponible, existen otras situaciones donde la información no está disponible o no es útil y, sin embargo, es deseable que se generen recomendaciones para los usuarios.

Los objetivos que establecimos al principio esta investigación fueron:

- Estudiar como modelar la información en este tipo de problemas.

- Mejorar los procesos de recogida de información del usuario en sistemas de recomendación.
- Diseñar modelos de sistemas de recomendación que generen recomendaciones en situaciones en donde la información histórica no esta disponible, bien porque no exista, sea escasa, o no sea relevante.
- Implementar un prototipo de sistema de recomendación que pueda ser aplicado al sector turístico.

Teniendo en cuenta estos objetivos, hemos presentado tres propuestas de modelos de Sistemas de Recomendación para alcanzarlos:

1. **Modelo basado en contenido con información lingüística multigranular:** las características fundamentales de este modelo son las siguientes:
   - Trabaja con información lingüística. El dominio lingüístico es más adecuado, ya que, estamos modelando información subjetiva, gustos o preferencias adquirida mediante percepciones, y por lo tanto, tienen asociado un alto grado de incertidumbre.
   - Genera recomendaciones sin necesidad de información histórica. Este modelo de recomendación permite a los usuarios que especifiquen su perfil, por lo que ya no es necesario el uso de la información histórica sobre éste.
   - Trabajan con información multigranular. Definimos un contexto multigranular que permitiera, tanto a los usuarios normales del sistema, como a los expertos que describen los productos, utilizar las escalas lingüísticas más adecuados con respecto a las características que están evaluando o al grado de conocimiento que tengan.

2. **Modelos basados en conocimiento:** este tipo de modelos fueron creados para utilizarse para generar recomendaciones cuando no existe información histórica. En nuestra memoria hemos presentado los siguientes modelos basados en conocimiento:

   a) *Modelo de recomendación basado en conocimiento con información lingüística multigranular (o RML):* en este modelo, el perfil de usuario se genera a partir de un producto ejemplo de las necesidades de este dado por el usuario. Frecuentemente, este ejemplo no representará de forma exacta las necesidades del usuario, y se requerirá una fase de refinamiento. Las principales mejoras con respecto a los modelos clásicos basados en conocimiento son:

   - Trabaja con información lingüística.
   - Trabaja con información multigranular.

   Además, ofrece algunas ventajas sobre el modelo basado en conocimiento presentado en esta memoria:

   - Es más fácil de construir el perfil de usuario, éste es más completo y por lo tanto es de esperar que se obtengan mejores recomendaciones
   - Los cálculos para generar las recomendaciones son más eficientes, y por lo tanto, podemos manejar bases de datos de mayor tamaño y generar recomendaciones en tiempos reducidos.

   b) *Modelo de recomendación basado en conocimiento basado en relaciones de preferencia lingüísticas (o RRPI):* uno de los inconvenientes de los sistemas de recomendación basados en conocimiento, es que el proceso de refinamiento del perfil de usuario puede ser largo y complejo. Debido a que nuestros primeros estudios se centraron en el modelado

de relaciones de preferencia y su reconstrucción, encontramos que éstas podían ser útiles para mejorar el proceso de adquisición de información en los sistemas de recomendación, y propusimos su uso en el modelo RRPI, cuyas principales mejoras son:

- No existe fase de refinamiento. El perfil de usuario se genera a partir de cuatro ejemplos de las necesidades del usuario y una relación de preferencia incompleta.
- Minimizamos la información necesaria para generar las recomendaciones: en situaciones reales, no podemos exigirle al usuario que proporcione una relación de preferencia completa, ya que, le debería dedicar mucho, y en la mayoría de los casos, esta sería la causa de que desistiera de sus búsquedas. En este modelo, hemos incluido las herramientas matemáticas necesarias para que los modelos sean capaces de reconstruirla a partir de la relación preferencia incompleta proporcioana por el usuario.
- Trabaja con información lingüística.

El modelo RRPI, fue utilizado para implementar un sistema de recomendación híbrido para restaurantes de la provincia de Jaén, REJA. Y así, lo dotamos de la capacidad de generar recomendaciones en situaciones en los que los sistemas de recomendación clásicos no podían.

En relación a la difusión y publicaciones de los resultados de nuestra investigación en este campo, destacaremos las siguientes publicaciones:

- Luis Martínez, Luis G. Pérez y Manuel Barranco, A Multi-granular Linguis-

tic Content Based Recommendation Model. International Journal of Intelligent Systems. 22:5, 2007 pp.419-434.

- Luis Martínez, Luis G. Pérez, Manuel Barranco y Macarena Espinilla. A Knowledge Based Recommender System with Multigranular Linguistic Information. International Journal of Computational Intelligence Systems,1 (2), 2008, pp. 148-158.

- Luis Martínez, Luis G. Pérez, Manuel Barranco y Macarena Espinilla. Improving the Effectiveness of Knowledge Based Recommender Systems Using Incomplete Linguistic Preference Relations. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (aceptado, en espera de la publicación).

- Luis Martínez, Luis G. Pérez y Manuel Barranco. A Knowledge Recommender System based on Fuzzy Consistent Preference Relations. En el libro: Intelligent Decision and Policy Making Support Systems. Serie: Studies in Computational Intelligence , Vol. 117. Editores: Da Ruan, Frank Hardeman y Klaas van der Meer (Springer, 2008)

- Luis G. Pérez, Manuel J. Barranco y Luis Martínez. Exploiting Linguistic Preference Relations in Knowledge Based Recommendation Systems. Eurofuse 2007. Jaén. España

- Luis G. Pérez, Manuel Barranco y Luis Martínez. Building User Profiles for Recommender Systems from incomplete preference relations. FUZZ-IEEE 2007, IEEE International Conference on Fuzzy Systems. Londres

- L. Martínez, M. Barranco, L. Pérez, M. Espinilla y F. Siles. A Knowledge Based Recommendation System with Multigranular Linguistic Information.

En el 2007 International Conference on Intelligent Systems and Knowledge Engineering (ISKE2007), Oct 15-16, 2007, Chengdu, China.

Trabajos futuros

Una vez obtenidos los resultados previos, en el futuro nos planteamos mejorar los mismos en las siguientes líneas de actuación:

1. Mayor flexibilidad en el modelado de preferencias. Para ello, estudiaremos como adaptar los modelos presentados en esta memoria, para que trabajen en contexto con información heterogénea: numérica, intervalar y lingüística.

2. Estudiar modelos de hibridación más avanzados que el propuesto, con el objetivo de facilitar la obtención de recomendaciones.

3. Estudiar en mayor profundidad, el tratamiento de la información incompleta, para mejorar los procesos de adquisición de información en los sistemas de recomendación.

4. Avanzar en la implementación de REJA, para obtener a partir del prototipo actual, un sistema de recomendación completamente funcional y de uso comercial.

Apéndice A

Nociones y Conceptos Básicos de la Teoría de Conjuntos Difusos

En 1960, L.Zadeh propuso la teoría de conjuntos difusos [219] con el objetivo de resolver problemas en donde los enfoques clásicos no podían aplicarse, o no producían soluciones satisfactorias. En esta teoría, Zadeh generaliza la noción clásica de conjunto definiendo un nuevo tipo de conjunto, llamado "conjunto difuso", cuya frontera no es precisa. Los conjuntos difusos se definieron como una nueva forma de representación de la imprecisión y la incertidumbre [108, 224], distinta al tratamiento que se había llevado hasta ese momento mediante la Teoría Clásica de Conjuntos y la Teoría de la Probabilidad. Dentro de la Teoría de Conjuntos Difusos podemos discernir dos vertientes principales [157]:

1. *Una teoría matemática formal [95, 147]*, ampliando conceptos e ideas de otras áreas de la matemática como el álgebra, la Teoría de Grafos, la Topología, etc.

2. *Y una potente herramienta para tratar situaciones del mundo real en las que aparece incertidumbre (imprecisión, vaguedad, inconsistencia, etc.)*, esta se adapta con facilidad a diferentes contextos y problemas en donde la incerti-

dumbre juega un papel fundamental: teoría de sistemas [29, 156], teoría de la decisión [63, 58], bases de datos [25, 216], etc.

A.1. Conjuntos difusos y función de pertenencia

Usando la noción de conjunto clásica, podemos representar una agrupación en colecciones de objetos que cumplen unas determinadas características. De esta forma, en un universo de discurso X, cada uno de elementos pertenecientes a este universo se le asignará un valor del conjunto $\{0, 1\}$ para indicar si pertenece o no a dicho conjunto. Si un elemento cumple todas propiedades para pertenecer a dicho conjunto, se le asignará el valor 1, si por el contrario, no cumple todas las propiedades, se le asignará el valor 0 para indicar de esta forma que no pertenece a dicho conjunto.

Definición A.1. *Sea A un conjunto en el universo X, la función característica asociada a A, $A(x) \in X$, se define como:*

$$A(x) = \begin{cases} 1 \; si \; x \in A \\ 0 \; si \; x \notin A \end{cases}$$

La función $A : X \to \{0, 1\}$ induce una restricción, con un límite bien definido, sobre los objetos del universo X que pueden ser asignados al conjunto A.

Zadeh generalizó está definición permitiendo valores intermedios en la función característica asociada, denominándola a partir de ese momento función de pertenencia. Gracias a los conjuntos difusos, podemos manejar adecuadamente ciertas categorías de objetos del mundo real, que antes difícilmente se podían representar, pues éstas no tenían unos límites claros o bien definidos. Por ejemplo, si hablamos

del conjunto de ordenadores potentes, productos con buen sabor, coches rápidos,... podemos encontrarnos objetos que claramente pertenecen a dichos conjuntos, que sin lugar a dudas no pertenecen a dichos conjuntos, pero también podemos encontrarnos con objetos que pertenecen en mayor o menor medida. El grado de pertenencia de dichos objetos se puede expresar mediante un número real en el intervalo $[0, 1]$, cuanto más cercano a 1 sea el grado, mayor será el grado de pertenencia a dicha categoría y cuanto más cercano a 0 menor será.

Definición básica de conjunto difuso

A partir del punto anterior, intuitivamente se puede afirmar, que un conjunto difuso es una colección de objetos con valores de pertenencia entre 0 y 1. Los valores de pertenencia expresan los grados de pertenencia a dicho conjunto, o lo que es lo mismo, con qué grado cumplen las propiedades necesarias para pertenecer a dicho conjunto. Si este grado es 0, el objeto no pertenecerá el conjunto (exclusión completa), y si por el contrario es 1, cumple completamente todas las propiedades necesarias para pertenecer a dicho conjunto (pertenencia completa). Formalmente podemos definir los conjuntos difusos de la siguiente forma [219]:

Definición A.2. *Un conjunto difuso $\tilde{A}$ sobre X está caracterizado por una función de pertenencia que transforma los elementos de un dominio, espacio, o universo del discurso X en el intervalo* $[0, 1]$.

$$\mu_{\tilde{A}} : X \rightarrow [0, 1]$$

Así, un conjunto difuso $\tilde{A}$ en X puede representarse como un conjunto de pares ordenados de un elemento genérico x, $x \in X$ y su grado de pertenencia $\mu_{\tilde{A}}(x)$:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) / x \in X, \mu_{\tilde{A}}(x) \in [0,1]\}$$

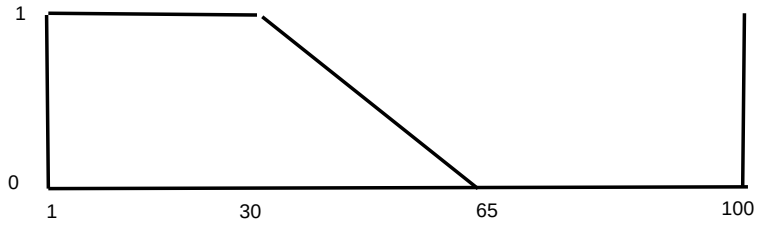
Como podemos observar, un conjunto difuso es una generalización del concepto de conjunto clásico cuya función de pertenencia sólo tomaba dos valores, 0 si no pertenecía a dicho conjunto, 1 si pertenecía.

Ejemplo

Supongamos que queremos modelar el concepto de "persona joven" con el objetivo de clasificar un conjunto de personas cuya edad oscila entre 0 y 100 años. Una persona menor o igual a 30 años se considerará joven y, por lo tanto, se le asignar un valor 1 a su grado de pertenencia al conjunto difuso de personas jóvenes. Una persona con una edad igual o superior a 65 años no puede considerarse como una persona joven y de ahí que se le asigne el valor 0 al grado de pertenencia al conjunto difuso de personas jóvenes. La cuantificación del resto de valores puede llevarse a cabo mediante una función de pertenencia $\mu_{\tilde{J}} : E \rightarrow [0,1]$ que caracteriza el conjunto difuso $\tilde{J}$ de personas jóvenes en el universo $E = [1, 100]$.

$$\mu_{\tilde{J}}(x) \begin{cases} 1 & x \in [1, 30] \\ 1 - \frac{x-30}{35} & x \in [30, 65] \\ 0 & x \in [65, 100] \end{cases}$$

Figura A.1: Función Edad



Los conjuntos difusos pueden ser definidos sobre universos discretos o continuos

usando distintas notaciones. Si un universo X es discreto y finito, con cardinalidad n, el conjunto difuso puede expresarse con un vector $n-dimensional$ cuyos valores son los grados de pertenencia de los correspondientes elementos de X. Por ejemplo, si $X = \{x_1, ..., x_n\}$, entonces un conjunto difuso $\tilde{A} = \left\{ \left(\frac{a_i}{x_i} \right) | x_i \in X \right\}$, donde $a_i = \mu_{\tilde{A}}(x_i), i = 1, ..., n$, puede notarse por [108]:

$$\tilde{A} = \frac{a_1}{x_1} + \frac{a_2}{x_2} + ... + \frac{a_n}{x_n} = \sum_{i=1}^{n} \frac{a_i}{x_i}$$

Cuando el universo X es continuo, para representar un conjunto difuso usamos la siguiente expresión:

$$\tilde{A} = \int_x \frac{a}{x}$$

donde $a = \mu_{\tilde{A}}(x)$ y la integral debería ser interpretada de la misma forma que el sumatorio en el universo finito.

A continuación, introducimos otros conceptos básicos a la hora de trabajar con conjuntos difusos, como son el soporte, el núcleo y el $\alpha-corte$ de un conjunto difuso:

Definición A.3. *El soporte de un conjunto difuso $\tilde{A}$, Soporte $\left(\tilde{A}\right)$, es el conjunto de todos los elementos de $x \in X$, tales que, el grado de pertenencia sea mayor que cero.*

$$Soporte\left(\tilde{A}\right) = \{x \in X \,|\mu_{\tilde{A}}(x) > 0\}$$

Si esta definición la aplicamos solo a aquellos elementos del universo X con grado de pertenencia igual a 1, tendríamos el núcleo del conjunto difuso.

Definición A.4. *El núcleo de un conjunto difuso $\tilde{A}$, $Nucleo\left(\tilde{A}\right)$, es el conjunto de todos los elementos de $x \in X$, tales que el grado de pertenencia es igual a 1.*

$$Nucleo\left(\tilde{A}\right) = \{x \in X \,|\mu_{\tilde{A}}(x) = 1\}$$

A veces puede ser más interesante conocer el conjunto de aquellos elementos que pertenecen a un conjunto con un grado menor, igual o mejor que un umbral determinado α que el grado al que pertenece un elemento a dicho conjunto. Estos conjuntos se denominan $\alpha - cortes$.

Definición A.5. *Sea $\tilde{A}$ un conjunto difuso sobre el universo X, dado un número $\alpha \in [0,1]$. Se define el $\alpha -$ corte sobre $\tilde{A}$, ${}^{\alpha}A$, como un conjunto que contiene todos los valores del universo X cuya función de pertenencia en $\tilde{A}$ sea mayor o igual al valor α:*

$${}^{\alpha}A = \{x \in X \,|\mu_{\tilde{A}}(x) \geq \alpha\}$$

A.2. Tipos de funciones de pertenencia

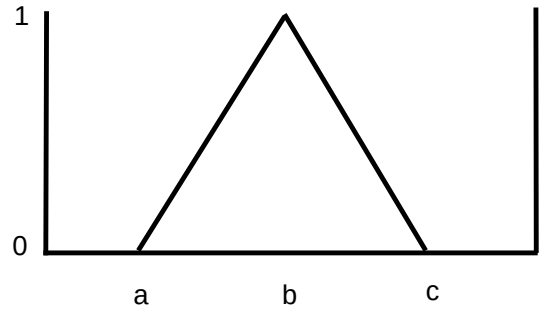
En principio cualquier función $\mu_{\tilde{A}} : X \to [0,1]$, describe una función de pertenencia asociada a un conjunto difuso $\tilde{A}$ que depende no sólo del concepto que representa, sino también del contexto en el que se usa. Sin embargo, existen un conjunto de familias de funciones, que son las que más comúnmente utilizadas. Para obtener una función de pertenencia concreta solo tenemos que fijar los parámetros

que definen la familia de una función. A continuación nombraremos las familias más utilizadas:

1. Funciones triangulares (ver figura A.2):

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & si\ x \leq a \\ \frac{x-a}{b-a} & si\ x \in [a,b] \\ \frac{c-x}{c-b} & si\ x \in [b,c] \\ 0 & si\ x \geq c \end{cases}$$

Figura A.2: Función triangular

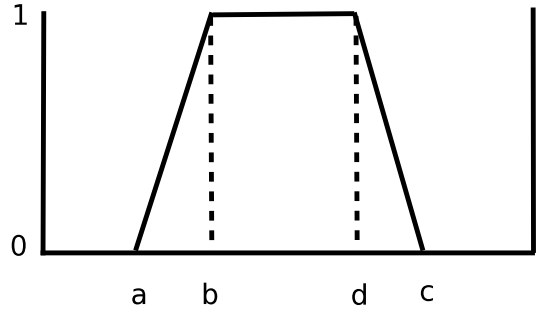


donde b es el punto modal de la función triangular y a y c los límites inferior y superior, respectivamente, para los valores no nulos de $\mu_{\tilde{A}}(x)$.

2. Funciones trapezoidales (ver figura A.3):

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & si\ x \leq a \\ \frac{x-a}{b-a} & si\ x \in [a,b] \\ 1 & si\ x \in [b,d] \\ \frac{c-x}{c-d} & si\ x \in [d,c] \\ 0 & si\ x \geq c \end{cases}$$

Figura A.3: Función trapezoidal

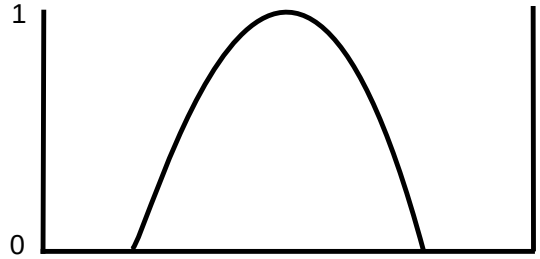


donde b y d indican el intervalo donde la función de pertenencia vale 1.

3. Funciones Gaussianas (ver figura A.4):

$$A(x) = e^{-k(x-m)^2}$$

Figura A.4: Función gaussiana



donde $k > 0$.

Principio de extensión

El principio de extensión es un concepto básico de la Teoría de Conjuntos Difusos. Se utiliza para generalizar conceptos matemáticos no difusos, tales como las operaciones aritméticas habituales sobre números reales, a conjuntos difusos

y ha sido formulado de diferentes formas a lo largo del tiempo [101, 108, 218]. A continuación presentaremos la definición más habitual del principio de extensión:

Definición A.6. *Sea X el producto cartesiano de los universos $X_1, ..., X_r$ y sean $\tilde{A}_1, ..., \tilde{A}_r$, r conjuntos difusos en $X_1, ..., X_r$ respectivamente. Sea f una función definida desde el universo X, ($X = X_1 \times ... \times X_r$), al universo Y, $y = f(x_1, ..., x_r)$. El principio de extensión nos permite definir un conjunto difuso $\tilde{B}$ en Y, a partir de los conjuntos difusos $\tilde{A}_1, ..., \tilde{A}_r$ representando su imagen a partir de la función f, de acuerdo a la siguiente expresión,*

$$\tilde{B} = \{(y, \mu_{\tilde{B}}(y)) / y = f(x_1, ..., x_r), (x_1, ..., x_r) \in X\}$$

donde

$$\mu_{\tilde{B}}(y) = \begin{cases} \sup_{(x_1,...,x_r)\in f^{-1}(y)} \min\left\{\mu_{\tilde{A}_1}(x_1), ..., \mu_{\tilde{A}_r}(x_r)\right\} & si\ f^{-1}(y) \neq \emptyset \\ 0, & en\ otro\ caso \end{cases}$$

Para $r = 1$, el principio de extensión se reduce a:

$$\tilde{B} = f(A) = \left\{y, \mu_{\tilde{B}(y)} / y = f(x), x \in X\right\}$$

donde

$$\mu_{\tilde{B}}(y) = \begin{cases} sup_{x\in f^{-1}(y)} \mu_{\tilde{A}}(x), & si\ f^{-1}(y) \neq \emptyset \\ 0, & en\ otro\ caso \end{cases}$$

A.3. Número difuso

Existe un subconjunto de los conjuntos difusos y que ha sido especialmente utilizado debido a las propiedades que cumple. Estos conjuntos difusos son conocidos como "números difusos" o "intervalos difusos" y se definen de la siguiente forma [218]:

Definición A.7. *Un número difuso $\tilde{A}$ es un subconjunto de $\mathbb{R}$ que verifica las siguientes propiedades:*

1. *La función de pertenencia es convexa,*

$$\forall x, y \in \mathbb{R}, \forall z \in [0,1], \mu_{\tilde{A}}(z) \geq \text{mín}\left\{\left\{\mu_{\tilde{A}}(z)\right\}, \left\{\mu_{\tilde{A}}(y)\right\}\right\}$$

2. *Para cualquier $\alpha \in (0,1]$, ${}^{\alpha}A$ debe ser un intervalo cerrado*

3. *El soporte de $\tilde{A}$ debe ser finito.*

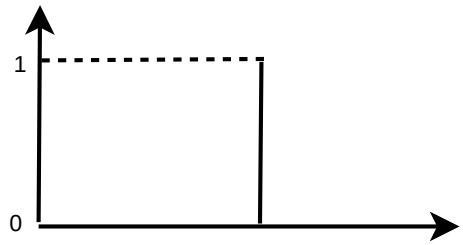
4. *$\tilde{A}$ está normalizado,*

$$\sup_x \mu_{\tilde{A}}(x) = 1$$

Casos particulares de números difusos [108]:

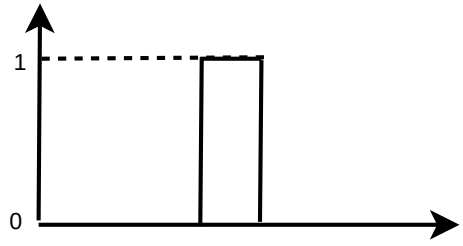
- Los números reales (ver figura A.5)

Figura A.5: Número real



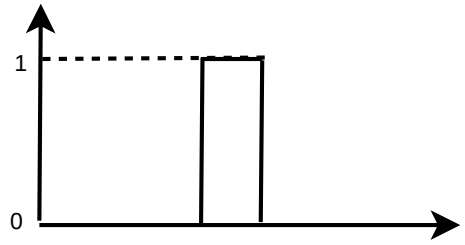
- Intervalos de números reales (ver figura A.6)

Figura A.6: Intervalo de números reales



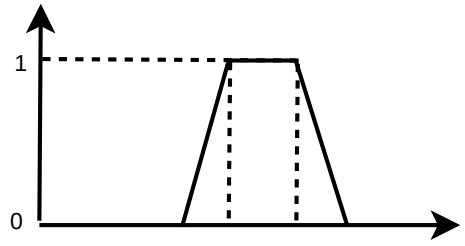
- Valores aproximados (ver figura A.7)

Figura A.7: Valores aproximados



- Intervalos aproximados o difusos (ver figura A.8)

Figura A.8: Intervalo aproximado



Apéndice B

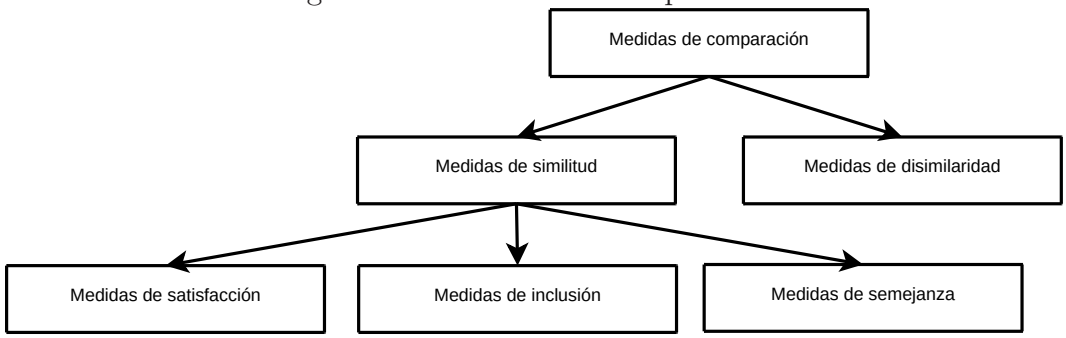
Medidas de comparación entre conjuntos difusos

La comparación de objetos ha sido una tarea muy habitual en muchos campos, tales como, la psicología, ciencias físicas, procesado de imágenes, clustering, y razonamiento deductivo. Generalmente, estas comparaciones están basadas en medidas de diferencia y de similitud entre dos objetos. Las medidas de comparación pueden tener ser expresadas mediante distintas funciones [51, 52, 225], dependiendo para qué, lo vayamos a utilizar. En la literatura podemos encontrar distintos tipos de medidas de comparación (ver figura B.1) [27, 169]:

1. *Medidas de satisfacción:* estas medidas se utilizan en situaciones donde se parte de un objeto referencia o clase y queremos saber si un nuevo objeto es compatible con él o lo satisface. Esta situación es típica en razonamiento basado en prototipos, donde las referencias son prototipos y un nuevo objeto debe ser asociado con una de ellas [26].

2. *Medidas de semejanza:* las medidas de semejanza son utilizadas para hacer comparaciones entre descripciones de objetos, al mismo nivel de generalidad, y para decidir si entre ellos hay suficientes características comunes.

Figura B.1: Medidas de comparación



Esta situación ocurre habitualmente en sistemas de razonamiento basados en casos. También es la base de la lógica de la similitud [54, 173].

3. *Medidas de inclusión:* en este tipo de medidas también se considera un objeto de referencia, tal y como, ocurre en las medidas de satisfacción. Si embargo, en este tipo de medida, se busca determinar la importancia de las características comunes en A y en B con respecto al objeto A. Estas medidas pueden ser usadas en sistemas de gestión de bases de datos para decidir si una clase está incluida en alguna otra [151, 171].

4. *Medidas de disimilaridad:* este tipo de medida no mide la similitud entre los objetos, sino la diferencias entre ellos. Esta medida esta basada en el concepto de distancia entre dos conjuntos difusos [109, 143, 160].

Las tres primeras medidas son conocidas también como medidas de similitud [27] (ver figura B.1). Por ejemplo, podríamos usar las medidas de similitud en razonamiento deductivo para evaluar si una observación satisface una regla dada o, en razonamiento basado en casos, para medir la semejanza entre las características de un caso conocido y uno nuevo.

En el modelo presentado en el capítulo 4 se utiliza una medidas de semejanza para comparar el perfil de usuario con la descripción de los objetos ambos

representables mediante conjuntos difusos. En [52] se propuso una medida de semejanza simple de utilizar y que es la que utilizaremos en el modelo propuesto en el capítulo 4 para el cálculo de la similitud entre dos conjuntos difusos:

$$D\left(A,B\right)=\sup_{x}\text{mín}\left(f_{A}\left(x\right),f_{B}\left(x\right)\right) \tag{B.1}$$

Esta medida aplicada a dos conjuntos difusos, A y B, obteniene una medida de cuanto se parece un conjunto a otro, cuanto mayor sea dicho valor, más parecidos serán.

Apéndice C

Rellenado relaciones de preferencia

En este apéndice, veremos las propuestas más significativas en la literatura para resolver el problema del rellenado relaciones de preferencia lingüísticas incompletas. Comentaremos, brevemente, su funcionamiento y las principales ventajas e inconvenientes. Finalmente, mostraremos la alternativa para rellenar relaciones de preferencias, que hemos encontrado más adecuada, para el tipo de problemas en el que desarrollamos nuestro trabajo. En donde debería tener más relevancia la información proporcionada por el usuario, que la estimada por el algoritmo.

Todos los algoritmos que explicaremos posteriormente partirán de una relación de preferencias P.

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix}$$

donde p_{ij} podrá ser un valor conocido o desconocido, y en donde p_{ii} tendrá el valor que representa la indiferencia.

C.1. Algoritmo de rellenado de relaciones de preferencia numéricas en el intervalo $[0,1]$

Este algoritmo fue presentado en [88], y usa la transitividad aditiva para rellenar la relación de preferencia de preferencia. Trabaja con matrices de preferencias numéricas en $[0,1]$. El método propone rellenar una relación de preferencia recíproca numérica definida en $[0,1]$ a partir de los $n-1$ valores conocidos $\{p_{12}, p_{23}, \ldots, p_{n-1n}\}$ de la relación de preferencia incompleta P siguiendo los siguientes pasos:

1. Calcula el conjunto de valores de preferencias B como
   $B = \{p_{ij}, i < j \wedge p_{ij} \notin \{p_{12}, p_{23}, \ldots, p_{n-1n}\}\}$,
   $p_{ij} = \frac{j-i+1}{2} - p_{ii+1} - p_{i+1i+2} \cdots - p_{j-1j}$

2. $a = |\text{mín}\{B \cup \{p_{12}, p_{23}, \ldots, p_{n-1n}\}\}|$

3. $P = \{p_{12}, p_{23}, \ldots, p_{n-1n}\} \cup B \cup \{1-p_{12}, 1-p_{23}, \ldots, 1-p_{n-1n}\} \cup \neg B$

4. La relación de preferencia consistente P' se obtiene como $P' = f(P)$ tal que
   $f : [-a, 1+a]\,[1,0]$,
   $f(x) = \frac{x+a}{1+2a}$

Este modelo fue extendido a modelos de decisión multiplicativos en [88].

Ejemplo

Si partimos de la siguiente relación de preferencia incompleta:

$$\begin{pmatrix} 0.5 & 0.55 & & \\ & 0.5 & 0.65 & \\ & & 0.5 & 0.75 \\ & & & 0.5 \end{pmatrix}$$

Se realizan los siguientes cálculos

$p_{31} = 1.5 - p_{12} - p_{23} = 1.5 - 0.55 - 0.65 = 0.3$

$p_{41} = 2 - p_{12} - p_{23} - p_{34} = 2 - 0.55 - 0.65 - 0.75 = 0.05$

$p_{42} = 1.5 - p_{23} - p_{34} = 1.5 - 0.65 - 0.75 = 0.1$

$p_{21} = 1 - p_{12} = 0.45$

$p_{13} = 1 - p_{31} = 0.7$

$p_{14} = 1 - p_{41} = 0.95$

$p_{32} = 1 - p_{23} = 0.35$

$p_{24} = 1 - p_{42} = 0.9$

$p_{43} = 1 - p_{34} = 0.25$

y se obtienen:

$$\begin{pmatrix} 0.5 & 0.55 & 0.7 & 0.95 \\ 0.45 & 0.5 & 0.65 & 0.9 \\ 0.3 & 0.35 & 0.5 & 0.75 \\ 0.05 & 0.1 & 0.25 & 0.5 \end{pmatrix}$$

Ventajas:

Las principales ventajas de este modelo es que obtenemos una relación de preferencia completa y consistente. Aunque en el modelo inicial, aquí revisado, se parte de $n-1$ valores $\{p_{12}, p_{23}, \ldots, p_{n-1n}\}$, en otros estudios posteriores y que presentaremos en las siguientes secciones se supera esta limitación y se permiten

utilizar otros $n - 1$ valores diferentes de la diagonal principal pero en el que al menos debe de compararse una de las alternativas una vez.

Inconvenientes:

La principal desventaja de este modelo es que puede requerir un paso, el número 4, para transformar los valores obtenidos por el algoritmo al intervalo $[0, 1]$, ya que, en ciertas condiciones, los valores calculados pueden superar dicho rango. Esto puede conllevar que se alteren los valores proporcionados por el usuario, $\{p_{12}, p_{23}, \ldots, p_{n-1n}\}$, los cuales suponemos como más importantes que los calculados por el algoritmo a la hora de representar las información de preferencia del usuario.

Ejemplo

Si partimos de la siguiente relación de preferencia incompleta:

$$\begin{pmatrix} 0.5 & 0.55 & & \\ & 0.5 & 0.65 & \\ & & 0.5 & 1 \\ & & & 0.5 \end{pmatrix}$$

Se realizan los siguientes cálculos

$p_{31} = 1.5 - p_{12} - p_{23} = 1.5 - 0.55 - 0.65 = 0.3$

$p_{41} = 2 - p_{12} - p_{23} - p_{34} = 2 - 0.55 - 0.65 - 1 = -0.2$

$p_{42} = 1.5 - p_{23} - p_{34} = 1.5 - 0.65 - 1 = -0.15$

$p_{21} = 1 - p_{12} = 0.45$

$p_{13} = 1 - p_{31} = 0.7$

$p_{14} = 1 - p_{41} = 1.2$

$p_{32} = 1 - p_{23} = 0.35$

$p_{24} = 1 - p_{42} = 1.15$

$p_{43} = 1 - p_{34} = 0$

y se obtienen:

$$\begin{pmatrix} 0.5 & 0.55 & 0.7 & 1.2 \\ 0.45 & 0.5 & 0.65 & 1.15 \\ 0.3 & 0.35 & 0.5 & 1 \\ -0.2 & -0.15 & 0 & 0.5 \end{pmatrix}$$

Como puede verse se han obtenido valores fuera del rango $[0, 1]$ y por lo tanto tenemos que hacer el paso 4 obteniendo:

$$\begin{pmatrix} 0.5 & 0.53 & 0.64 & 1 \\ 0.46 & 0.5 & 0.60 & 0.96 \\ 0.35 & 0.39 & 0.5 & 0.85 \\ 0 & 0.03 & 0.14 & 0.5 \end{pmatrix}$$

Puede comprobarse como no mantiene los valores originales dados por el usuario o experto.

C.2. Algoritmo de rellenado de preferencias lingüísticas

Este algoritmo fue presentado en [208]. Este modelo trabaja con una escala lingüística, $S = \{s_{-t}, \ldots, s_0, \ldots, s_t\}$, con cardinalidad impar en donde el término medio, s_0, representa la indiferencia. Este algoritmo de reconstrucción necesita expandir el dominio discreto de etiquetas a uno continuo $\overline{S} = \{s_\alpha \,|\alpha \in [-t, t]\}$ y definir una operación de agregación, $\oplus$:

Definición C.1. *Sea $s_\alpha, s_\beta \in S$, la operación $\oplus$ se define como:*

$$s_\alpha \oplus s_\beta = \text{máx}\left\{s_{-t}, \left\{s_{\alpha+\beta}, s_t\right\}\right\}$$

Zeshui Xu utiliza la transtividad aditiva para caracterizar la consistencia en una relación de preferencia y la define de la siguiente forma:

Definición C.2. [208] *Sea $P = (p_{ij})_{n\times n}$ una relación de preferencia lingüística y completa, P es una relación completa consistente si cumple:*

$$a_{ij} = a_{ik} \oplus a_{kj}\ \forall i, j, k$$

El algoritmo que emplea para la rellenado es el siguiente:

1. Sea $X = \{x_1, x_2, \ldots, x_n\}$ un conjunto discreto de alternativas. El experto debe proporciona $n-1$ comparaciones que no coincidan con la diagonal de la relación de preferencia ($i = j$). Además, deben de cumplir que al menos se debe de comparar cada una de las alternativas una vez.

2. Utilizar la transitividad aditiva definida como $a_{ij} = a_{ik} \oplus a_{kj}$ para, a partir de los elementos conocidos de la relación de preferencia, obtener los desconocidos

3. Fin

Ejemplo

Sea $S = \{s_{-2}, s_{-1}, s_0, s_1, s_2\}$ y la relación de preferencia incompleta la siguiente:

$$\begin{pmatrix} s_0 & s_1 & s_2 & s_2 \\ & s_0 & & \\ & & s_0 & \\ & & & s_0 \end{pmatrix}$$

Si empleamos el algoritmo anterior tendremos que realizar las siguientes operaciones:

$p_{2,1} = -p_{1,2}$

$p_{3,1} = -p_{1,3}$

$p_{4,1} = -p_{1,4}$

$p_{23} = p_{21} \oplus p_{13} = s_{-1} \oplus s_2 = s_1$

$p_{24} = p_{21} \oplus p_{14} = s_{-1} \oplus s_2 = s_1$

$p_{32} = p_{31} \oplus p_{12} = s_{-2} \oplus s_1 = s_{-1}$

$p_{34} = p_{31} \oplus p_{14} = s_{-2} \oplus s_2 = s_0$

$p_{42} = p_{41} \oplus p_{12} = s_{-2} \oplus s_1 = s_{-1}$

$p_{43} = p_{41} \oplus p_{13} = s_{-2} \oplus s_2 = s_0$

y obtendremos la siguiente relación de preferencia:

$$\begin{pmatrix} s_0 & s_1 & s_2 & s_2 \\ s_{-1} & s_0 & s_1 & s_1 \\ s_{-2} & s_{-1} & s_0 & s_0 \\ s_{-2} & s_{-1} & s_0 & s_0 \end{pmatrix}$$

Ventajas:

Este algoritmo presenta las siguientes ventajas:

- Mantiene las valoraciones proporcionadas por el usuario.
- Los cálculos son bastante sencillos de implementar y llevar a cabo.

Inconvenientes:

Este algoritmo presenta una serie de inconvenientes, tales como:

- La relación de preferencia resultante, no tiene porqué cumplir la transitividad aditiva, aunque en su reconstrucción nos hemos basado en ella.
- No se puede garantizar que se vaya a obtener una única relación de preferencia completa, ya que los valores de ésta dependen del orden en que se hayan rellenado los valores desconocidos.
- Aunque el modelo lingüístico usado para la reconstrucción tiene cierto parecido al modelo de 2-tupla [81], adolece en la interpretación de los valores. Así, mientras que en las 2-tuplas, el valor $(s_2, -0.3)$ tiene una interpretación lingüística asociada, el valor $s_{1.7}$ en esta representación, es considerada como una etiqueta virtual sin ningún significado o interpretación.

Los motivos de estos inconvenientes las enumeramos a continuación:

- Si uno de los valores desconocidos supera los rangos de definición del problema, éste es truncado por el límite superior o por el inferior. Como se puede ver este valor ya no cumplirá la transitividad aditiva.
- No se describe el algoritmo con una formulación formal, por lo que no refleja ni soluciona las situaciones en donde, debido a que no se cumple la transitividad aditiva, se pueden obtener distintos valores para un mismo valor desconocido.

Ejemplo

Sea $S = \{s_{-2}, s_{-1}, s_0, s_1, s_2\}$ y la relación de preferencia incompleta la siguiente:

$$\begin{pmatrix} s_0 & s_{-1} & s_1 & s_2 \\ & s_0 & & \\ & & s_0 & \\ & & & s_0 \end{pmatrix}$$

Si empleamos el algoritmo anterior:

$p_{21} = -p_{12} = s_1$

$p_{23} = p_{21} \oplus p_{13} = s_1 \oplus s_1 = s_2$

$p_{24} = p_{21} \oplus p_{14} = s_1 \oplus s_2 = s_3$

$p_{31} = -p_{13} = s_{-1}$

$p_{32} = p_{31} \oplus p_{12} = s_{-1} \oplus s_{-1} = s_{-2}$

Si calculamos el valor de p_{34} nos encontramos con que tenemos dos posibles alternativas para hacerlo, y que en ambas alternativas, se obtienen valores distintos:

$p_{34} = p_{31} + p_{14} = s_{-1} \oplus s_2 = s_1$

$p_{34} = p_{32} + p_{24} = s_{-2} \oplus s_2 = s_0$

C.3. Algoritmo de rellenado de relaciones de preferencias numéricas, intervalares o lingüísticas

Este algoritmo presentado en [4], permite rellenar relaciones de preferencia tanto numéricas como intervalares y/o lingüísticas. También utiliza la transitividad aditiva para dicho cometido. El objetivo de este algoritmo es proporcionar una solución única, con un alto grado de consistencia, y que no altere la infor-

mación proporcionada por el usuario. En el caso lingüístico, para representar la información se utiliza la 2-tupla lingüística y el algoritmo se podría escribir de la siguiente forma:

1. Inicialización:

   $P' = \Delta^{-1}(P)$

   $EMV_0 = \emptyset$

   $h = 1$

2. mientras $EMV_h \neq \emptyset\{$

3. para cada $(i,k) \in EMV_h\{$

4. $\mathcal{K} = \emptyset$

5. $H_{ik}^1 = \{j \neq i,k \,|(i,j),(j,k) \in KV_h\}; si\ (H_{ik}^1 \neq \emptyset)\ entonces\ \mathcal{K} = \mathcal{K} \cup \{1\}$

6. $H_{ik}^2 = \{j \neq i,k \,|(i,k),(j,i) \in KV_h\}; si\ (H_{ik}^2 \neq \emptyset)\ entonces\ \mathcal{K} = \mathcal{K} \cup \{2\}$

7. $H_{ik}^3 = \{j \neq i,k \,|(i,j),(k,j) \in KV_h\}; si\ (H_{ik}^3 \neq \emptyset)\ entonces\ \mathcal{K} = \mathcal{K} \cup \{3\}$

8. $Calcular\ cp_{ik}^{'} = \frac{1}{\#\mathcal{K}} \left(\sum_{l \in \mathcal{K}} \frac{\sum_{j \in H_{ik}^l}^{n} cp_{ik}^{jl}}{\#H_{ik}^l} \right)$

9. $si\ cp_{ik}^{'} < 0\ entonces\ p_{ik}^{'} = 0$

10. $si\ cp_{ik}^{'} > 1\ entonces\ p_{ik}^{'} = g$

11. $sino\ p_{ik}^{'} = cp_{ik}^{'}$

12. $h++$

13. $\}$

14. P''=$\Delta(p')$

Donde:

KV_h son los valores conocidos en la iteración h

UV_h son los valores desconocidos en la iteración h

EMV_h es el subconjunto de valores perdidos que pueden ser calculados en la iteración h

$EMV_h = \{(i,k) \in UV_h \,|\exists j \in H_{ik}^1 \cup H_{ik}^2 \cup H_{ik}^3\}$

$cp_{ik}^{j1} = p_{ij} + p_{jk} - \frac{g}{2}$

$cp_{ik}^{j2} = p_{jk} - p_{ji} + \frac{g}{2}$

$cp_{ik}^{j3} = p_{ij} - p_{kj} + \frac{g}{2}$

Ejemplo

Sea $S = \{s_0, s_1, s_2, s_3, s_4\}$ la escala lingüística que el experto puede utilizar, donde s_2 representa el valor indiferencia. Si el experto nos proporciona la siguiente relación de preferencia:

$$\begin{pmatrix} (s_2,0) & & (s_1,0) & \\ & (s_2,0) & & (s_2,0) \\ (s_3,0) & & (s_2,0) & (s_1,0) \\ & (s_3,0) & (s_3,0) & (s_2,0) \end{pmatrix}$$

Después de aplicar el algoritmo anteriormente descrito obtendremos la siguiente matriz de preferencia:

$$\begin{pmatrix} (s_2,0) & (s_1,-0.39) & (s_1,0) & (s_0,0) \\ (s_4,-0.39) & (s_2,0) & (s_3,-0.33) & (s_2,0) \\ (s_3,0) & (s_2,-0.33) & (s_2,0) & (s_1,0) \\ (s_4,0) & (s_3,0) & (s_3,0) & (s_2,0) \end{pmatrix}$$

Ventajas:

Las ventajas que proporciona son las siguientes:

- Mantienen las valoraciones proporcionadas por el experto.
- Presentan una versión del algoritmo para distintos dominios: numérico, lingüístico e intervalar.
- Proporciona una solución única.
- Los autores presentarón en [4] como calcular el grado de consistencia de la relación de preferencia rellenada.

Inconvenientes:

También presenta una serie de inconvenientes:

- Es más complejo que los anteriores, más difícil de utilizar y de implementar.
- Ignora valores obtenidos en la misma iteración para el cálculo del resto de valores de esa iteración. Este inconveniente no se puede evitar, ya que si se tuvieran en cuenta, se podrían obtener distintas relaciones de preferencia dependiendo del orden en el calcularamos los valores desconocidos, y por lo tanto, no podríamos garantizar que siempre obtuvieramos una relación única.

Ejemplo

Supongamos que partimos de la siguiente matriz de preferencia:

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ & p_{22} & & \\ & & p_{33} & \\ & & & p_{44} \end{pmatrix}$$

Siguiendo este algoritmo en el paso 1 será $EMV_1 = \{(2,3),(3,2),(2,4),(4,2),(3,4),(4,3)\}$ y para calcular estos valores se tendrá que hacer:

- p_{23} se calculará a partir de p_{13} y p_{12}
- p_{32} se calculará a partir de p_{12} y p_{13}
- p_{24} se calculará a partir de p_{14} y p_{12}
- p_{42} se calculará a partir de p_{12} y p_{14}
- p_{34} se calculará a partir de p_{14} y p_{13}. Sin embargo, en este cálculo se están omitiendo los valores ya calculados en esta misma iteración. Si se tuvieran en cuenta p_{34} se podría calcular también a partir de p_{32} y p_{24}, y también desde p_{24} y p_{23}.
- p_{43} se calculará a partir de p_{13} y p_{14}. Sin embargo, en este cálculo se están omitiendo los valores ya calculados en esta misma iteración. Si se tuvieran en cuenta p_{43} se podría calcular también a partir de p_{42} y p_{23}, y también desde p_{23} y p_{24}.

C.4. Propuesta de algoritmo de rellenado de relaciones de preferencia incompletas con tendencia al valor de indiferencia

En esta sección presentamos nuestra propuesta para completar relaciones de preferencia incompletas que utilizaremos en nuestro modelo de recomendación RRPI. Dicha propuesta trata de mejorar las desventajas que tienen los algoritmos alteriores.

Los algoritmos anteriores parten de la hipótesis de que los seres humanos son consistentes con respecto a la transitividad aditiva. Sin embargo, hay estudios que demuestran que la toma de decisiones está más relacionada con la parte emocional y por lo tanto no es totalmente racional [15, 96]. De aquí, se puede deducir que, la relación de preferencia calculada mediante estos algoritmos no tiene porque coincidir con la que el usuario daría si tuviera el tiempo necesario o el suficiente grado de conocimiento, ya que, estas preferencias no son totalmente racionales aunque si mantienen las tendencias de las preferencias.

Nuestra propuesta para completar una relación de preferencia incompleta parte de que no se puede predecir de forma exacta, cuál sería la preferencia de un usuario dada un par de alternativas, si él mismo puede no saberlo. Nuestro objetivo es presentar un algoritmo que tenga en cuenta que la información aportada por el usuario es más relevante que la inferida, es decir, el conocimiento dado por el usuario es más importante que la estimación inferida.

Para reflejar esto, presentamos un algoritmo basado en [4] que en lugar de obtener los distintos valores para una preferencia mediante una media, lo que hacemos es seleccionar aquel valor que sea el más cercano a la indiferencia. De esta forma damos menos importancia a los valores estimados que a los proporcionados por el usuario. En sistemas de recomendación esta alternativa podría proporcionar mejores resultados, a la hora de calcular el perfil de usuario.

El algoritmo, para el caso lingüístico, aquí comentado sería el siguiente:

Sea P la relación de preferencias lingüística incompleta que queremos rellenar,

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix}$$

donde p_{ij} podrá ser un valor conocido o desconocido, y en donde p_{ii} tendrá la

valoración lingüística que representa la indiferencia.

Los pasos a seguir para rellenar la relación de preferencia P son los siguientes:

1. Inicialización:

   $P' = \Delta^{-1}(P)$

   $EMV_0 = \emptyset$

   $h = 1$

2. mientras $EMV_h \neq \emptyset\{$

3. para cada $(i,k) \in EMV_h\{$

4. $\mathcal{K} = \emptyset$

5. $H_{ik}^1 = \{j \neq i,k \,|(i,j),(j,k) \in KV_h\}$; $si\ (H_{ik}^1 \neq \emptyset)\ entonces\ \mathcal{K} = \mathcal{K} \cup \{1\}$

6. $H_{ik}^2 = \{j \neq i,k \,|(i,k),(j,i) \in KV_h\}$; $si\ (H_{ik}^2 \neq \emptyset)\ entonces\ \mathcal{K} = \mathcal{K} \cup \{2\}$

7. $H_{ik}^3 = \{j \neq i,k \,|(i,j),(k,j) \in KV_h\}$; $si\ (H_{ik}^3 \neq \emptyset)\ entonces\ \mathcal{K} = \mathcal{K} \cup \{3\}$

8. $Calcular\ p'_{ik} = masCercanoIndiferencia(cp_{ik}^{jl}, \forall l \in K, \forall j \in H_{ik}^l)$

9. $h++$

10. $\}$

11. P''=$\Delta(P')$

Donde:

KV_h son los valores conocidos en la iteración h

UV_h son los valores desconocidos en la iteración h

EMV_h es el subconjunto de valores perdidos que pueden ser calculados en la iteración h

$EMV_h = \{(i,k) \in UV_h \,|\exists j \in H_{ik}^1 \cup H_{ik}^2 \cup H_{ik}^3\}$

$cp_{ik}^{j1} = p_{ij} + p_{jk} - \frac{g}{2}$

$cp_{ik}^{j2} = p_{jk} - p_{ji} + \frac{g}{2}$

$cp_{ik}^{j3} = p_{ij} - p_{kj} + \frac{g}{2}$

$masCercanoIndiferencia(cp_{ik}^{jl}, \forall l \in K, \forall j \in H_{ik}^l)$: esta función devolverá, de todos los valores posibles de cp_{ik}^{jl}, el más cercano al valor de indiferencia.

Ejemplo

Supongamos que partimos de la siguiente matriz de preferencia:

$$P = \begin{pmatrix} (s_2,0) & (s_0,0) & (s_1,0) & (s_3,0) \\ & (s_2,0) & & \\ & & (s_2,0) & \\ & & & (s_2,0) \end{pmatrix}$$

Si empleamos nuestro algoritmo tendremos que realizar las siguientes operaciones:

$cp'_{23} = p'_{13} - p'_{12} + 2 = 1 - 0 + 2 = 3;\ p'_{23} = 3$

$cp'_{32} = p'_{12} - p'_{13} + 2 = 1;\ p'_{32} = 1$

$cp'_{24} = p'_{14} - p'_{12} + 2 = 3 - 0 + 2 = 5;\ p'_{24} = 4$

$cp'_{42} = p'_{12} - p'_{14} + 2 = 0 - 3 + 2 = -1;\ p'_{42} = 0$

$cp'_{34} = p'_{14} + p'_{13} + 2 = 3 - 1 + 2 = 4;\ p'_{34} = 4$

$cp'_{43} = p'_{13} - p'_{14} + 2 = 1 - 3 + 2 = 0;\ p'_{43} = 0$

$$\begin{aligned} cp'_{21} &= masCercanoIndiferencia\,(p'_{23} - p'_{13} + 2, p'_{24} - p'_{14} + 2) = \\ &= masCercanoIndiferencia\,(3 - 1 + 2, 4 - 3 + 2) = \\ &= masCercanoIndiferencia\,(4,3) = 3; p'_{21} = 3 \end{aligned}$$

$$\begin{aligned} cp'_{31} &= masCercanoIndiferencia\,(p'_{32} - p'_{12} + 2, p'_{34} - p'_{12} + 2) = \\ &= masCercanoIndiferencia\,(1 - 0 + 2, 4 - 3 + 2) = \\ &= masCercanoIndiferencia\,(3, 3) = 3; p'_{31} = 3 \end{aligned}$$

$$\begin{aligned} cp'_{41} &= masCercanoIndiferencia\,(p'_{42} - p'_{12} + 2, p'_{43} - p'_{13} + 2) = \\ &= masCercanoIndiferencia\,(0 - 0 + 2, 0 - 1 + 2) = \\ &= masCercanoIndiferencia\,(2, 1) = 2; p'_{41} = 2 \end{aligned}$$

Y obtendremos la siguiente relación de preferencia:

$$P = \begin{pmatrix} (s_2, 0) & (s_0, 0) & (s_1, 0) & (s_3, 0) \\ (s_3, 0) & (s_2, 0) & (s_3, 0) & (s_4, 0) \\ (s_3, 0) & (s_1, 0) & (s_2, 0) & (s_4, 0) \\ (s_2, 0) & (s_0, 0) & (s_0, 0) & (s_2, 0) \end{pmatrix}$$

A continuación enumeraremos las ventajas del algoritmo que acabamos de proponer en esta sección:

- Mantiene las valoraciones proporcionadas por el experto.
- Proporciona una solución única.
- Tiene en cuenta que la información aportada por el usuario es más importante que la inferida.

Apéndice D

English summary

Here, we include a summary of this thesis, entitled *Recommender System Models with Lack of Information. Applications to the Tourism Sector*,as partial fulfilment for the *European Ph. D.* In this summary, we will present several recommender models whose aims are to make recommendations when the historical information is not available or useful.

First of all, our motivations and aims of this report are exposed. Secondly, each model is presented with a summary, in which we will introduce its main aims, features, advantages and disadvantages, and afterwards, it is attached the document where the model was presented. Thirdly, REJA, a recommender system software program is presented in section D.8. Finally, the conclusions, our future works and research are pointed out and commented.

D.1. Motivations

In the beginning, my research was focused on the topic of preference modelling and decision making. Specifically, I studied the use of incomplete preference relations, filling methods to complete them, properties of preference relations, etc. Meanwhile, I was also looking for what fields would be useful for these studies and we stared at Internet.

Internet has caused important changes in our society. It offers free on-line access to encyclopedia, any kind of dictionary, pieces of news from anywhere in the world, thousands of products or different services. There are an endless number of possibilities we can do or buy by using Internet. The most important limitation users have, when they surf internet, is the time they can spend searching for what they need or they like. This limitation has been an important drawback in some environments.

In Internet, we can find many services or business models that, in the beginning, they have been expected to be overwhelmingly successful, but, actually, they suffered from important economic losses, some of them went bankrupt, or fired the staff. One of the most affected fields was the e-commerce, which was expected to be very successful but, however, it just met the lowest success expectations.

The e-commerce companies offered, in their e-shops, a wide range of products with the aim of satisfying hundreds or even thousands of clients individually. At first, this was an important advantage for customers since, they have a more variety of products and more information without needing to go out and go to a retail shop. However, this wide range of products, instead of being an advantage, it was a disadvantage. When users visited a e-shop and committed their necessities trough a query, these e-shops gave back thousands of products related to that

query. Nevertheless, not all of them were useful for the customers, and among the products that could satisfy their necessities, only a few of them met customers' needs. Finding these products it was a time-consuming and boring task. Then, many users gave their searches up and preferred go shopping to retail shops where they were easy guided by shop assistants in order to find appropriate products, even though, the quality of the products could be worse or had a lesser variety of products.

To address these problems, some tools were developed in the e-commerce field. The most successful ones have been the recommender systems [30, 31, 41, 113, 166, 179, 192]. These systems assist customers in their search processes in the e-shops. There is not a unique recommender model, but a family of them that shares the same aim, to lead the users to interesting products, and the same working structure:

1. Recommender systems use background data to make the recommendations.
2. Customers provide information about their necessities and/or tastes.
3. An algorithm combines the background data and information provided by the users to order to generate recommendations.

However, the use of recommender systems presents some drawbacks:

1. *The use of precise scales:* many recommender systems force their customers to use precise scale, usually numbers. Although these scales make easier the management of the data, since it is represented in the same domain as the domain used internally by the recommender system, this domain is more difficult to be used by human beings, because the information is usually related to perceptions, tastes or opinions.

2. *An unique scale is used:* these systems use a unique scale without taking into account what they are evaluating, or the experts' degree of knowledge. Most of recommender systems internal operations are designed to deal with data expressed in an unique domain. Even though, the data has been provided by customers or experts and they might have different degree of knowledge about them.

3. *Historical information:* the recommendations are usually based on historical information about the user past actions, opinions, or purchases. If these systems deal with new users, they do not have any information about them, and they are then unable to generate any recommendation.

4. *Dependent on the database density:* the density of the database is hightly related to the quality of the recommendations. Many systems encourage their users to provide more information about their tastes, or necessities. However, in some cases, this is impossible or not advisable because of the nature of the problem, or because it is not expected that user interacts with the system again.

Our aim in this research report is to propose new recommendations models to address or smooth out the previous problems. We then expect these new models would be useful in a wide range of situations. For example, within tourist sector, recommending items such as restaurants, hotels, monuments, and scenic places can be a very difficult task for classic recommender models as:

1. There is no historical information.

2. Users cannot spend enough time to provide the information needed to make the recommendations since most of these models required much information.

Eventually, we will implement a hybrid recommender system, that includes some of the models presented in this summary, to make recommendations about restaurants in the province of Jaén.

D.2. Aims

The main aims of this report are the following ones:

- To study how to model the information in this kind of problems. We want to know which structures and domains can be used and which ones are the most suitable. Due to the fact that the information we are going to deal with, is subjective and uncertain since is related to users' opinions, tastes and perceptions.

- To improve the users' information gathering process in recommender system in order to define the users' profiles. We want users to be required the least quantity of information to let the system define their user profile. To do so, we will use incomplete preference relations.

- To design recommender system models that are able to make recommendations in situations where there is no historical information about their past opinions, rates, or purchases, or the historical information is scarce.

- To implement a prototype of a recommender system that can be used in the touristic sector. Our aim is to implement a software that uses both classic recommender system models and the models proposed in this report.

D.3. Structure of this report

This report is structure in the following sections:

- In section D.4, recommender systems are reviewed briefly, since in the scientific documents attached in this summary explain them in detail.

- In section D.5, the tools we have used to model and manage the information provided by the user are enumerated. These tools are explained in depth within the documents where the models were presented.

- In section D.6, a content based recommendation model adapted to situations where there is no historical information is presented. This model is defined in a multigranular linguistic context in order to model the information, both the user preferences and the descriptions of the items.

- In section D.7, two knowledge based recommendation models are presented. The first one is also defined in a multigranular linguistic context to model the information and improve the previous model as it can be applied to a greater number of products. The second one, to improve the user preference gathering process with the aim of building better user profiles. To do so, we propose the use of linguistic preference relations.

- In section D.8, an implementation of a recommender system applied to the touristic sector is presented. Specifically, this recommender system is designed to make recommendations of restaurants in the province of Jaén.

- Finally, we will point out the most important conclusions obtained in this report. Moreover, we will describe our future works and researchs, and

lastly, we have gathered the most outstanding bibliography related to the topic of this report.

D.4. Recommender Systems

This section is a brief summary of the full chapter presented in the chapter 2 of the Spanish version of this report. Throughout the chapter, recommender systems are defined, a classification is given and some commercial examples are explained.

Recommender systems are a kind of software whose aim is to assist customers with their *shopping searches* by leading them to interesting items by means of recommendations.

Before the development of recommender systems, users usually had to choose from the first subset of items that could partially fulfil their needs or give their searchers up, because, unfortunately, they are unable to explore the entire range of items in the e-shop. This problem caused important looses in the e-commerce companies what lead them to develope tools to assist users in their purchase processes. The most important key issues in the development and success of e-commerce[180], were the recommender systems.

Recommender system are classified in six categories:

- *Demographic Recommender Systems [148, 155]:* these systems categorize their users into demographic groups, and make recommendations to a specific user according to the information about the people who belong to the same demographic group.

- *Content-based Recommender Systems [7, 17, 116, 128, 146, 148, 126, 154, 183]:* a content-based recommender system learns a user profile based on the features of the items that the user has liked and, it uses this profile to find out similar items that the user could like.

- *Collaborative Filtering Recommender Systems [19, 28, 41, 59, 64, 67, 72, 92, 159, 165, 176, 185, 182]:* they use users' ratings to filter and recommend items to a specific user. In the simplest case, these systems predict the users' preferences as a weighted aggregation of the other users' preferences.

- *Knowledge Based Recommender Systems [30, 32, 201]:* these systems use the knowledge about users' necessities and how an item matches these necessities to infer recommendations about which items fulfil the user's expectations.

- *Utility Based Recommender Systems [31]:* they make recommendations by computing a utility value for each object.

- *Hybrid Recommender Systems [13, 30, 40, 155]:* these systems arose with the aim of addressing several drawbacks presented in the previous ones. To accomplish this aim, they combine different techniques to improve the accuracy of the recommendations.

Trought the chapter, the content-based, the knowledge based and the collaborative filtering recommender systems are explained in depth, since the models presented in the thesis used them. In this summary, the Content-based recommender system model is presented in section D.6, two Knowledge based recommender system models in section D.7, and, an implementation of an hybrid recommender system, composed of a collaborative filtering and knowledge based recommender system, is presented in section D.8 and a further detail about them are presented in such sections.

D.5. Information modeling in recommender system

Throughout the chapter 3 of the Spanish version of this report, it was reviewed the linguistic background and preference structures used in the models presented in the report.

As I have aforementioned, one of our aims is to improve the preference modeling of the information involved in the recommender systems. To do so, the chapter 3 reviewed the Fuzzy Linguistic Approach [218], since it is used to model the information provided by the users. This information is usually vague and imprecise as it is related to human perceptions, opinions, tastes, etc.

It was also reviewed the structures used to model the users' information. Although the most common structure to model the user information is a utility vector[49, 127, 189], in the chapter, the preference relations [56, 79, 84, 103, 104, 206] are explained in depth since one of our models uses them to make the recommendations.

We also focused on how to deal with multigranular linguistic contexts [76, 97] because, both the content based recommender model, presented in section D.6 , and the first knowledge based recommender model, presented in section D.7, are defined in such a type of context. Moreover, it is given a brief explanation of the 2-tuples linguistic representation model [81] since it is used in the multigranular context and in the last of our model, in the knowledge based recommender Systems that uses Incomplete Linguistic Preference Relations to make recommendations.

All these concepts, tools and models are explained within the attached documents in sections D.6 and D.7.

D.6. A Multi-granular Linguistic Content Based Recommender Model

Content-based recommender model bases their recommendation on historical information about the user past purchase, their rates, tastes and/or opinions. They define the user profile gathering the main features or the items the user has bought or liked, and look for other items with similar features (see figure D.1).

Although, these models has been used successfully in a wide range of problems, they present some drawbacks that made them unsuitable in some situations:

- They are unable to provide recommendations if there is no historical information about the user.
- All the recommendations are related to the users' past actions. This means that this kind of recommendation systems for examples does not adapt their recommendations if the users change their tastes quickly either if they want to receive recommendations for a particular reason not related to their past.

In order to smooth out these problems we developed a content-based recommendation model (see figure D.2) whose aims are the followings:

- To be able to make recommendation without needing historical information about the user.
- To offer the users a linguistic environment where they are able to express their necessities or taste easily but without lost of accuracy.

Therefore, we decided let users define their own user profile. When a user expresses their necessities, they usually provide the features they would like to find

Figure D.1: Content based recommender system

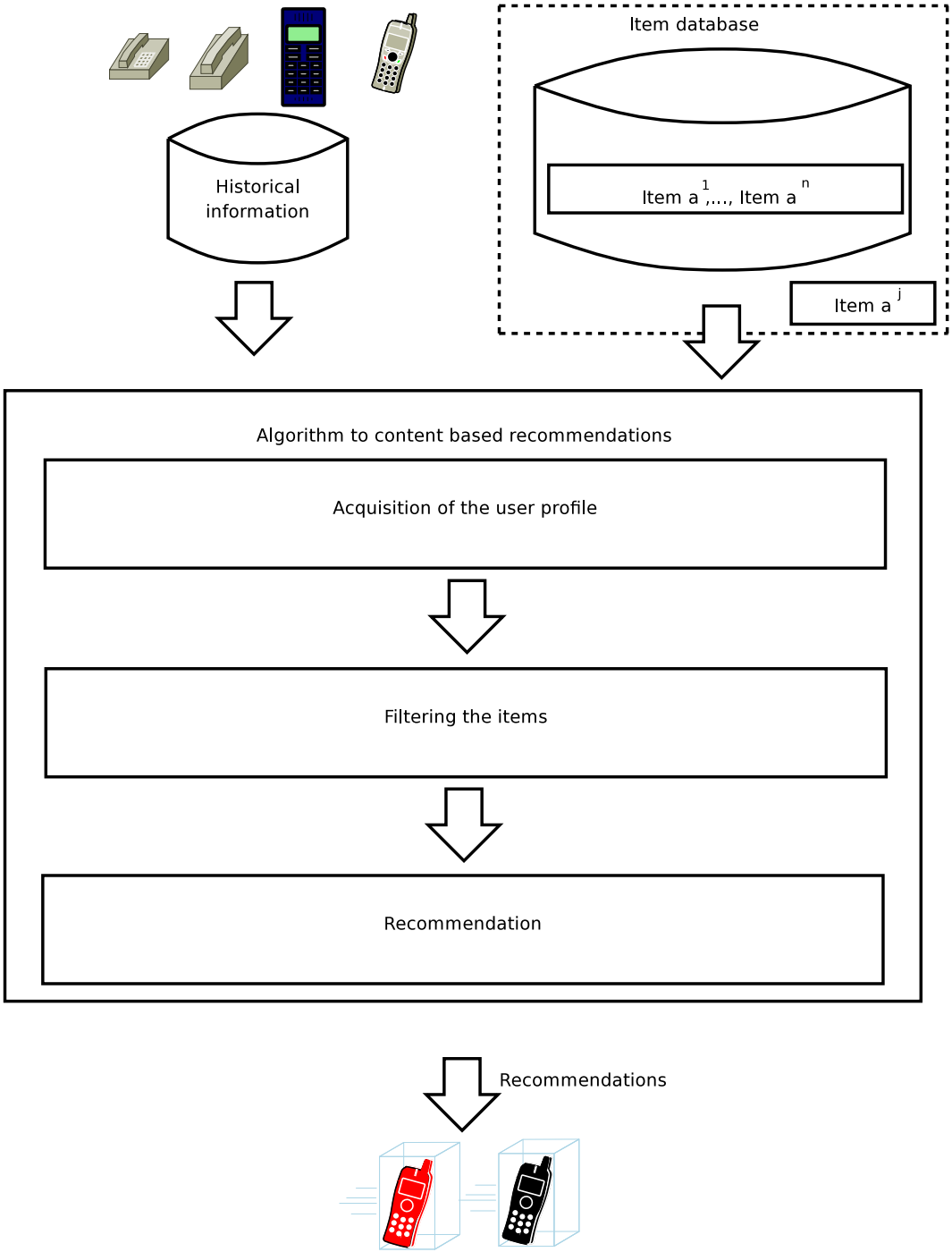
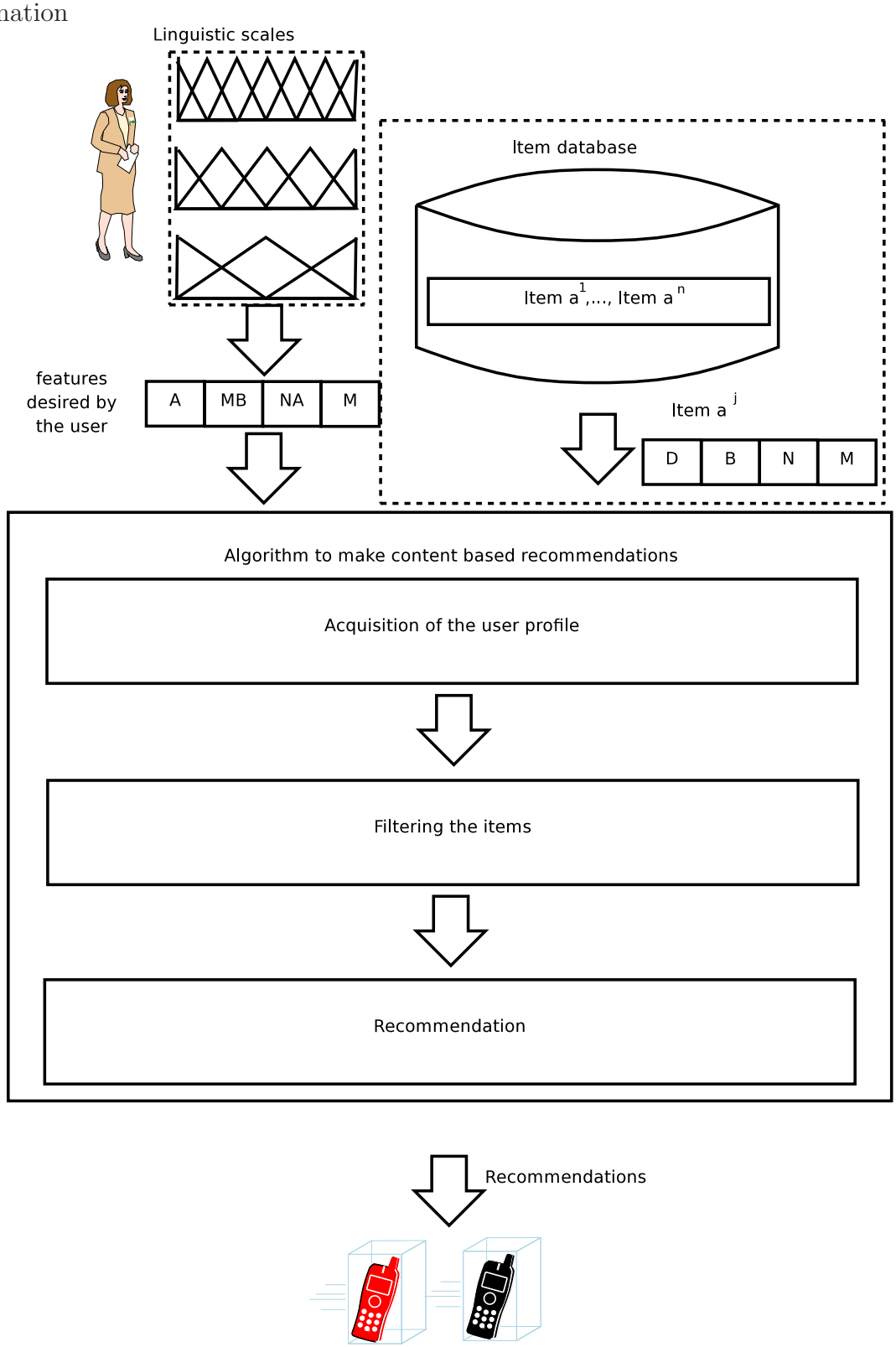


Figure D.2: Content based recommender system without using historical information



in the items they would like to buy. These features are usually related to tastes, opinions and/or gathered through perceptions. However, classical recommender systems force their users to provide this information by using precise information, i.e., numbers as the domain used internally by the recommender system. We think that this domain is not the most suitable in the real world since this information is usually expressed by words, and not with number. For this reason, we proposed that this model will deal with linguistic information instead of numerical one. However, the use of a multigranular context is even better because of the following reasons:

- The model is dealing with two kinds of users with different degree of knowledge: on the one hand, customers who wants to buy items, and on the other hands, experts who have described all the items.
- Each feature could have a different nature, for example, it could be gathered by the different senses, and, each sense has a different meaning.

A detailed description of this model is attached following, by means of the paper published in the International Journal of Intelligent System (22:5) that introduces the context and background necessary to understand the proposed model and finally such a model is described in detail.

A Multi-granular Linguistic Content-Based Recommendation Model

L. Martínez, L.G. Pérez, M. Barranco

Dept. Of Computer Sciences, University of Jaén,23071, Jaén, Spain

Abstract

The massive use of Internet has developed new types of services in different areas as business, commerce, education, administration, etc. A really important and promising or these topics is the e-commerce. In this paper, we focus on the area of the B2C in which the development of intelligence e-services plays a key role to be successful in this new market. One of the most studied services in this area are the Recommendation Systems, that try to help the users to find out the most suitable products they are looking for among a vast quantity of information that there exist in Internet or in a website. There are different types of recommendation systems: (i) Based on content, (ii) Collaborative and (iii) Hybrid. These systems deal with information provided by the customers regarding the features they wish, in order to reach the most suitable product for them. This information is related to the human perception because it expresses the necessities, preferences, taste, etc., of the customers, i.e., usually it is qualitative in nature, however current recommendation systems force the customers to provide their preferences using a predefined numerical scale, it seems not very suitable due to the uncertainty that involves this type of information. We propose in this paper a based-content recommendation model to deal with qualitative information by means of the fuzzy linguistic approach because it is more suitable to model qualitative aspects. This model will allow to the customers not only to provide their preferences in a linguistic way but also they could assess their preferences in different scales, i.e., different linguistic term sets. Because different customers can have different perceptions about their own preferences or taste. Hence, our proposal consists of offering a multi-granular linguistic context to the customers instead of forcing all of them to provide their preferences in the same scale. Once the customers have provided their profiles, these will be matched with the products features in which the customers are interested and according to the resemblance obtained, the recommendation model will choose the most suitable products for each customer.

Keywords: e-commerce, e-services, recommendation systems, fuzzy linguistic approach, fuzzy rankings, decision-making

1. Introduction

Almost all the areas related to the human beings has been interfered by Internet in the last years, this has implied the appearance of new market niches, services, and much information available for

the users. This explosive growth has produced that one of the main problems users face navigating in Internet is the vast quantity of information they find, being most of it useless for their aims. Due to this fact, different e-services have risen to help the internet users to reach easy and quickly their necessities, such services, can help them to obtain some information or to find out a product they are searching and so forth. In this paper, we focus on the recommendation systems that help the users to find out the most suitable products according to their preferences, necessities or taste, hiding or removing the useless information there are in the websites. Companies such as google, amazon or Los Angeles Times use recommendation systems to assist users in their searches. The recommendation systems are a class of software [Kau98, Res97] that has emerged in the last years as an e-service within the domain of the E-Commerce [Sch01].

The purpose of these systems is to recommend the most suitable items, from a set of them according to the user's desires. Traditionally, these systems have fallen in three main categories: (i) *Collaborative filtering systems* [Gol92, Kon97, Per99, Sha95]. (ii) *Content-based filtering systems* [Lie95, Paz96, Jen92] and *Hybrid content-based and collaborative recommendation systems* [Bas98, Pop01, Vie04]. These systems gather preference information from the customers, experts, etc., rank the items, and make a decision about what items are the most attractive to the users. From this viewpoint, the final recommendation could be seen as a decision making process, such that, it makes a recommendation about which item(s) are the most preferred for the user. This decision is made taking into account the preferences and opinions gathered by the Recommendation Systems from different types of information sources [Ans00,Bre98]. These information sources provide their preferences, *source profile*, to the Recommendation System as opinions about their necessities regarding the items they are searching, according to their own perceptions. This type of information is subjective because it is related to the sources own perceptions and usually involves uncertainty. So, the information provided by these sources is usually vague, incomplete and not precise. However, most of recommendation systems force the sources to express their preferences using just one numerical scale [Hay01]. This fact implies a lack of expressiveness for the sources and bounds to a lack of precision in the recommendations made by the systems.

Despite there are different types of recommendation systems as we have aforementioned, we focus on content-based recommendation models that filter and recommend items according to a matching process between the customers' profiles and the description of the items in its database in order to choose those one(s) pretty similar to the customer's preferences.

In this paper, we propose a new model to improve the effectiveness of the recommendations given by the content based Recommendation Models. It consists of offering a multi-granular linguistic context [Her00] to model the user preferences as the item features. Therefore, the users could express their preference information using linguistic assessments instead of numerical ones, due to the fact that linguistic information is more suitable than numerical to assess qualitative information [Zad75] (human perceptions, taste, necessities). In addition, each user can choose their own linguistic term set to provide their preference information according to their knowledge about the products. Also in the item database the item features provided by experts will be assessed by means of linguistic labels that could be assessed in different linguistic term sets. To deal with the multi-granular linguistic information in our recommendation model we shall use the fuzzy linguistic approach [Zad75] to model the input information and fuzzy tools, such as, fuzzy measures of comparison [Bou95, Bou96] to evaluate the resemblance of the products with the customers' profiles and rank them.

Our proposal for a multi-granular linguistic content-based recommendation model will act according to the followings steps (graphically, see Fig.1):

1. *Acquisition of the user profile and the item features*: the user profile is an information structure to gather the information provided by the user about his/her necessities, tastes, interesting areas, etc. and the item features are the characteristics of the items to be recommended stored in a database that are provided by experts. In this model so the customers as the experts will provide their information by means of linguistic information and depending on the aspect they are assessing they can use linguistic assessments belonging to different linguistic term sets.
2. *Matching items for the user*: to find out the most interesting items for the customers, the model compares each item in its item database with the customer necessities (profiles) by means of fuzzy measurements of comparison.
3. *Making a Recommendation*: the model will rank the items according to its similarity with the customer profile, such that, those items top ranked will be recommended to the customer.

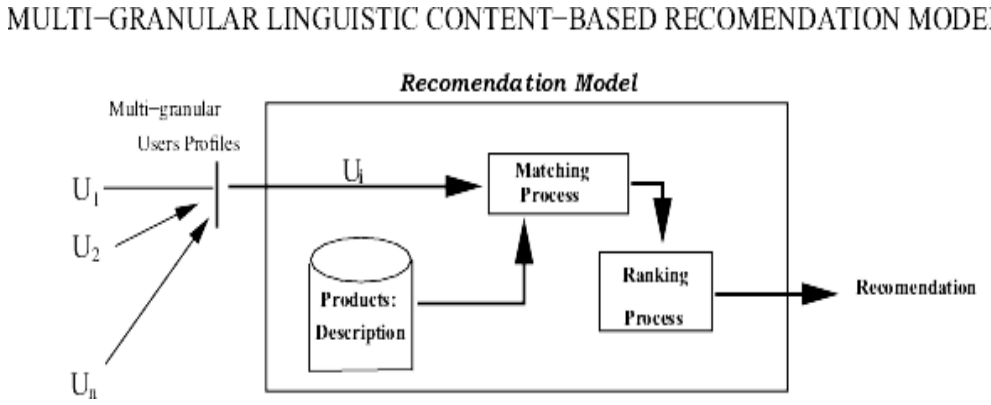


Fig. 1: A Multi-Granular Linguistic Content-Based Recommendation Model

This paper is structured as follows. In the Section 2 we shall make brief review of the fuzzy linguistic approach and different recommendation models. In the Section 3 we present our multi-granular based content recommendation model, and in the Section 4, we shall show the working of this model with a simple example. The paper is concluded in the Section 5.

2. Preliminaries

In this section we shall review some core concepts about the fuzzy linguistic approach and also review the different types of recommendation models we can find in the literature.

2.1 Fuzzy Linguistic Approach

Usually, we work in a quantitative setting, where the information is expressed by means of numerical values. However, many aspects of different activities in the real world cannot be assessed in a quantitative form, but rather in a qualitative one, i.e., with vague or imprecise knowledge. In such a case, a better approach may be to use linguistic assessments instead of numerical values. The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables [Zad75]. This approach is adequate in some situations, such as, when attempting to qualify phenomena related to human perception, we are often led to use words in natural language.

The fuzzy linguistic approach has been successfully applied to different problems [Bor93,Del92,Her95,Ya95].

We have to choose the appropriate linguistic descriptors for the term set and their semantics. In order to accomplish this objective, an important aspect to analyse is the "granularity of uncertainty", i.e., the level of discrimination among different counts of uncertainty. Therefore, according to the source of information knowledge it can choose different counts of uncertainty. The universe of the discourse over which the term set is defined can be arbitrary, linguistic term sets are usually defined in the interval [0,1]. In [Bon86] the use of term sets with an odd cardinal was studied, representing the mid term by an assessment of "approximately 0.5", with the rest of the terms being placed symmetrically around it and with typical values of cardinality, such as 7 or 9. In this paper, we shall deal with sources of information with different degrees of knowledge, so each one could use different linguistic term sets with different granularity. We call this type of context multi-granular linguistic contexts [Her00].

One possibility of generating the linguistic term set consists of directly supplying the term set by considering all terms distributed on a scale on which a total order is defined [Her95,Yag95]. For example, a set of seven terms S, could be given as follows:

$$S = \{s_0 = \text{None}; s_1 = \text{Very Low}; s_2 = \text{Low}; s_3 = \text{Medium}; s_4 = \text{High}; s_5 = \text{Very High}; s_6 = \text{Perfect}\}$$

In these cases, it is usually required that there exist:

- A negation operator $\text{Neg}(s_i) = s_j$ such that $j = g-i$ (g+1 is the cardinality).
- A minimization and a maximization operator in the linguistic term set: $s_i <= s_j \Leftrightarrow i <= j$.

The semantics of the terms are given by fuzzy numbers defined in the [0,1] interval, which are described by membership functions. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function [Bon86]. Since the linguistic assessments given by the users are just approximate ones, some authors consider that linear trapezoidal membership functions are good enough to capture the vagueness of those linguistic assessments, since it may be impossible and unnecessary to obtain more accurate values [Del92].

This parametric representation is achieved by the 4-tuple **(a, b, d, c)**, where **b** and **d** indicate the interval in which the membership value is 1, with **a** and **c** indicating the left and right limits of the definition domain of the trapezoidal membership function [Bon86]. A particular case of this type of representation are the linguistic assessments whose membership functions are triangular, i.e., b = d, so we represent this type of membership function by a 3-tuple (a; b; c). For example, we may assign the following semantics to the set of seven terms (see Fig. 2).

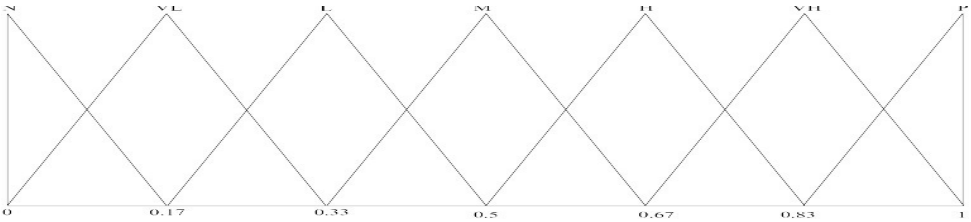


Fig. 2: A linguistic term set of seven terms and its semantics

Other authors use a non-trapezoidal representation, e.g., Gaussian functions [Bor93].

2.2 Recommendation Systems

The current recommendation systems can be classified attending to the process and the source of information that are used to achieve the recommendations.

In [Ans00] we can see that the information used by the recommendation systems may be provided from different sources and there exist, at least, these five types of information sources:

a) A person's expressed preferences or choices among alternative products.
b) Preferences for product attributes.
c) Other people's preferences or choices.
d) Expert judgments.
e) Individual characteristics that may predict preferences.

So depending on which sources and how the recommendation system deals with the information gathered in order to produce a recommendation could be distinguished three main classes of recommendation systems:

1. ***Collaborative filtering systems***: [Gol92,Kon97,Per99,Sha95] *use explicit and implicit preferences from many users to filter and recommend objects to a given user, ignoring the representation of the objects*. In the simplest case, these systems predict a person's preference as a weighted sum of other people's preferences, in which the weights are proportional to correlations over a common set of items evaluated by two people. Collaborative filtering algorithms were first introduced by Golberg and colleagues (Goldberg et al. 1992). They are used by Los Angeles Times, London Times, CRAYON, and Tango to customize online newspaper; by Bostondine to recommend restaurant in and around Boston; by Sepia Video Guide to make customized video recommendations; by Movie Critic, Moviefinder and Morse to recommend movies; and by barnesandnoble.com to recommend books.
2. ***Content-based filtering systems***: [Lie95,Paz96,Jen92] *filter and recommend the items by matching user query terms with the index term used in the representation of the items, ignoring data from other users*. There are some commercial systems has been offered by PersonalLogic, Frictionless Commerce, and Active Research that use self-explicated importance ratings and/or attribute trade-offs to make their recommendations.
3. ***Hybrid content-based and collaborative recommendation systems***: [Bas98,Pop01,Vie04] this new class has emerged between the content-based and collaborative recommendation systems and its aim is to smooth out the disadvantages of each one of them. *A usual way to hybrid both classes is to make a two level filter algorithm, where we use first one of the algorithm (the content-based filtering algorithm) to obtain the first set of items and afterwards, we use the second algorithm (the collaborative filtering algorithm) to filter and recommend items from this se*t. Applications of hybrid-based recommendation systems on the Web include search tools such as Google (www.google.com) and Inquirus 2 (inquirus.nj.nec.com/i2/inq2.pl) that combines results of both content searches and collaborative recommendations. However, these systems are more complex and have got new design problems to resolve in order to handle efficiently all the information available.

In this paper we focus on content based recommendation models.

3. A Multi-granular linguistic Content Based Recommendation model

Here, we present our proposal for a multi-granular linguistic content based recommendation model. Therefore, in order to make a recommendation for a user, it will just consider information

from the customer and the objects in its item database, ignoring information from other users:

- A person's expressed preferences or choices among alternative products: *Customer profile.*
- Preferences for product attributes: *Item features.*

In our proposal, both the *user profiles* and the *item features* can be assessed by means of multi-granular linguistic information, it means, different customers or experts can use different linguistic term sets to provide their assessments. The recommendation process will consist of a matching process between the customers profiles and the items features of each item in the database to obtain a measure of similarity, and afterwards the products will be ranked according to this measure in order to be recommended (graphically, see Fig. 3):

1. *Acquisition of the user profiles and item features*: in this stage the item features are added to the item database if it is necessary and the user preferences are gathered into a profile.
2. *Matching items for the user*: it computes a measure of similarity between the user profile and each item stored in the item database.
3. *Making a recommendation*: it chooses the most suitable item/s for a customer according to their similarity with his profile.

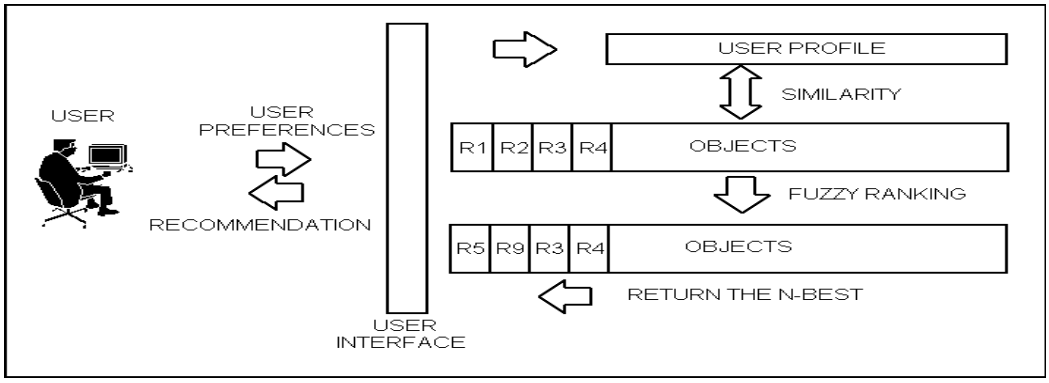


Fig. 3: Content-Based Recommendation model

In the following subsections we shall present in detail these stages and how this model works in order to recommend the most suitable products to the customers.

3.1 Acquisition of the user profiles and the item features

The aim of this stage is to gather the information with regards to products in which the customer u is interested. To do so, let $C=\{c_1,\ldots,c_k,\ldots,c_l\}$ be a set of criteria or attributes that the customer uses to describe his necessities, preferences and taste about the items he is interested.

The recommendation model will have a set of items or products (item database) that can be recommended, where each *item* is described by a set of values, *item features,* for each criterion, c_k, provided by some experts where these values will be linguistic labels that could assessed in different linguistic term sets.

To obtain an user profiel, each user u, that wants to obtain a recommendation about the items or products of the item database, must provide his/her profile according to his/her preferences.

Mathematically we can describe this stage in our recommendation model as a phase in which a customer u who wants to find out which is the most suitable product/s among a set of them:

$$A=\{a_1,\ldots,a_j,\ldots,a_n\},$$

so the user will use a recommendation model which describes each object, a_j, by means of a set of criteria:

$$C=\{c_1,\ldots,c_k,\ldots,c_l\},$$

such that, the recommendation model has a database in which each item is described by means of a vector of item features (Table 1):

$$F_j=\{v_1^j,\ldots,v_k^j,\ldots,v_l^j\},\ j=1,\ldots,n$$

being v_k^j a linguistic value provided by the user for the object, a_j, of the criterion, c_k.

Our model will recommend those products a_j more similar or suitable to the user preference. Therefore the user, u, will provide his preference profile:

$$P_u=\{p_1^u,\ldots,p_k^u,\ldots,p_l^u\},$$

by means of a utility vector that express his preferences, necessities and taste with regards to the products he is looking for. Where, $p_k^u \in S_{uk,}$ is the linguistic value that the user u assign to the criterion c_k, according to their knowledge, taste, preference and necessities and S_{uk} is the linguistic term set chose by the customer u to provide his preference about the criterion c_k.

Different customers can have different perceptions about their own preferences or taste or even the same customer can have different knowledge about his preference in different criteria. Hence, we offer the possibility that customers can assess their preferences in different linguistic term sets according to their knowledge. So, in our proposal we offer the customers a flexible multi-granular linguistic context instead of forcing all of them to provide their preferences in the same scale. Therefore, each user can choose his own linguistic terms set to provide his profile.

3.2 Matching Items for each user

Once we have got the user profiles the recommendation model will have*:*

a) A user profile $P_u=\{p_1^u,\ldots,p_l^u\}$ with the user preferences provided by the user u, which are described by means of linguistic labels assessed in $S_{uk.}$

b) A set of products $A=\{a_1,\ldots,a_n\}$ described by means of their item features $F_j=\{v_1^j,\ldots,v_l^j\}$ for each criterion/attribute $C=\{c_1,\ldots,c_l\}$ assessed linguistically. The item features will be in the item database as can be seen in the Table 1.

	c_1	,...,	c_k	,...,	c_l
a_1	v_1^1	,...,	v_k^1	,...,	v_l^1
...	...		...		...
a_j	v_1^j	,...,	v_k^j	,...,	v_l^j
...	...		...		...
a_n	v_1^n	,...,	v_k^n	,...,	v_l^n

Table 1: Item features in the database of items

In order to find out which are the most suitable product/s for a customer u, the recommendation model will compare the user profile P_u with the item features of all the items in the database by means of a matching process in order to obtain the closest products in the database according to user preferences or necessities.

In our case, the information that represents so the user profiles as the item features are linguistic labels whose semantics are given by fuzzy numbers, so to carry out this matching process we need measures of comparison between fuzzy numbers. In the following subsections we shall review in short this type of measures and afterwards we present the matching process used by our model to obtain the similarity between the items and the user profile.

3.2.1 Measures of Comparison

The comparison of objects is a usual task in many fields as psychology, analogy, physical sciences, image processing, clustering, deductive reasoning, etc. Generally, these comparisons are based on measures of the differences and similitudes between two objects. In the literature we can find different types of comparison measures [Bou96, Rif00]:

1. *Measures of satisfiability*: These measures correspond to a situation in which we consider a reference object or a class and we need to decide if a new object is compatible with it or satisfies it.
2. *Measures of resemblance*: A measure of resemblance is used for a comparison between the descriptions of two objects, of the same level of generality, to decide if they have *enough* common characteristics.
3. *Measures of inclusion*: Considering a reference object like in the measures of satisfiability. We obtain how important are the common characteristics of *A* and *B*, with regards to *A*.
4. *Measures of dissimilarity*: This is other kind of measure that not assesses the similitude but the differences. This measure is based on the concept of distance between two fuzzy sets.

In our proposal to compare the user profiles and the item features, we shall use measures of resemblance. Before showing the measures of resemblance we are going to use, it is necessary to revise some basic concepts.

For any set Ω of elements, let $F(\Omega)$ denote the set of fuzzy subsets of Ω, f_A the membership function of any set A in $F(\Omega)$ and $supp(A) = \{x \in \Omega / f_A(x) \neq 0\}$

To compare two fuzzy sets, it is important to consider the intersection of them. The membership function of the intersection set is given as:

$$f_{A \cap B}(x) = min(f_A(x), f_B(x))$$

Definition 1. [Bou96]: A fuzzy set measure M is a mapping: $F(\Omega) \rightarrow [0,1]$ such that, for every A and B in $F(\Omega)$:

P1: $M(0)=0$

P2: *if* $B \subseteq A$, *then* $M(B) \leq M(A)$

In [Dub83] was proposed a measure of resemblance which is easy to manage and compute. This one has been widely used in the literature to carry out this kind of processes.

M1: $D(A,B) = \sup_x \min\left(f_A(x), f_B(x)\right)$

After applying this measure to two fuzzy sets, we obtain some knowledge about their similarity: the greater value the more similarity.

3.2.2 Matching Process

In the second stage, the recommendation model will compute the resemblance between the user profile P_u and the item features F_j of each product in the item database, a_j, $j=1,\ldots,n$.

Therefore, let $P_u=\{p_1^u,\ldots,p_l^u\}$ *be* an user profile and $\{F_j, j=1..n\}$ a set of products features where $F_j=\{v_1^j,\ldots,v_l^j\}$. *O*ur proposal to obtain a similarity measure, r_k^j, between each correspondent customer preference and item feature assessment, (p_k^u, v_k^j), for all the products *j*, and for all attributes, *k*. It is to apply a matching process by means of the similarity measure M1:

$$r_k^j = D\left(p_k^u, v_k^j\right) = \sup_x \min\left(f_{p_k^u}(x),\ f_{v_k^j}(x)\right)$$

Hence, we shall obtain as similarity measure, between a user profile and each item, a fuzzy set, $R_j^u = \left(r_1^j,...,r_l^j\right)$, where each component, r_k^j , is computed by the above measure of resemblance between each user profile criterion c_k, i.e. p_k^u, and its correspondent item feature that describes the value of the criterion c_k for the product a_j, i.e. v_k^j.

$$R_j^u = Similarity\left(P_u, F_j\right) = \left(r_1^j,...,r_l^j\right)$$

For instance, let's consider the product criterion, c_k, and we want to match it with the user preference p_k^u . Both of them are assessed by means of linguistic terms in the linguistic term set of the *Fig.2*. In this case we have for a product a_j the values $p_k^u = L$ and $v_k^j = M$, then, the matching process will be (graphically, see Fig. 5):

$$r_k^j = \sup_x \min\left(L, M\right) = 0.5$$

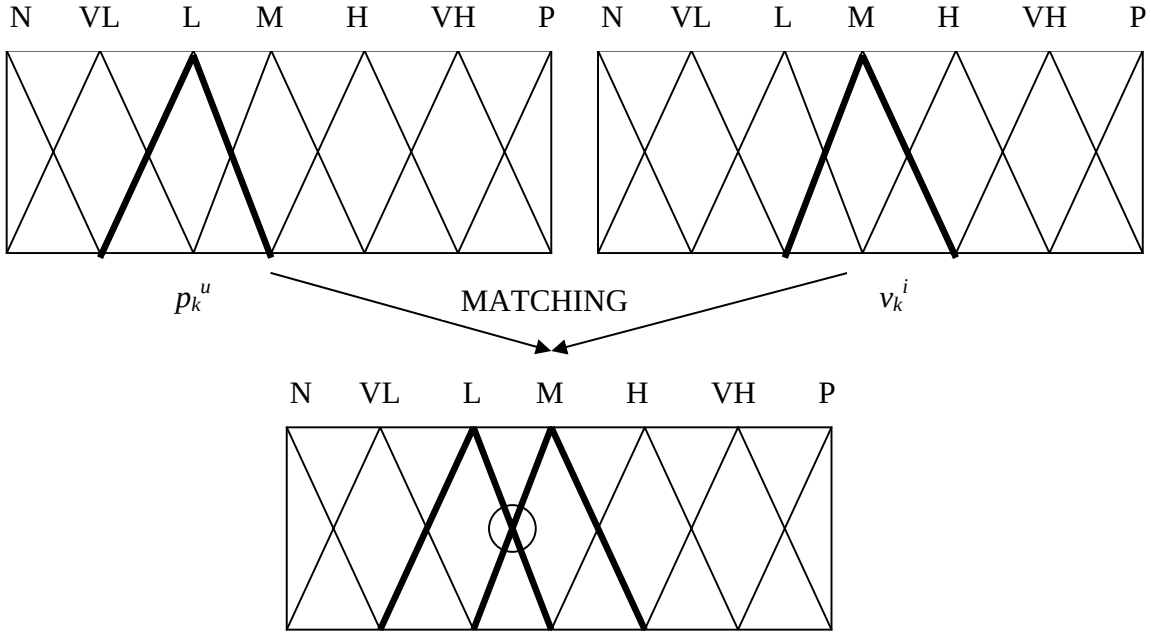


Fig.5: Matching Process

3.3 Making a Recommendation

The objective of a recommendation model is to find out which are the most suitable product/s for the customer. So far, we have computed the similarity, R_j^u , between each item, F_j, and the customer profile, P_u. We shall consider in this stage of the recommendation process that the similarity can be interpreted as a preference, due to the fact that the greater value the more suitable.

Therefore to achieve the objective, we have to rank the items according to their similarity with the user profile, but the similarity values computed are expressed by means of fuzzy sets. Therefore to rank them and recommend the most suitable product/s, we shall use the three-step ranking process presented in [Her00]:

1. To build a preference relation from the measures of similitude

2. To compute a Non Dominance Degree (NDD) for each item
3. To rank the items according to the NDD, and recommend the *n* top ranked.

Following we show in further detail each step of the recommendation stage.

3.3.1 Building a preference relation

Here we shall build a preference relation, $Q=[q_{ij}]$, from the similarity values, R_j^u , in order to rank the items according to their similarity with the user preferences.

To do so, we shall use an inclusion measure. In the literature we can find different types of inclusion measures [San79, Dub80]. But after a study about them and considering the facility of computing and its good performance in our model, we have chosen the next one:

M2: $S(A,B) = \inf_x \min\left(1 - f_A(x) + f_B(x), 1\right)$

Let *A* and *B* be two fuzzy sets, an inclusion measure, *S(A,B)*, computes how much *A* is included in *B*, however, we want to know how much *A* covers *B* to interpret this value as a preference one. Therefore, to obtain the preference degree of *A* over *B*, q_{AB} , we shall compute the inclusion of *B* in *A*:

$$q_{AB}=S(B,A)$$

Consequently to build the preference relation, *Q*, from the similarity measures:

$$R_j^u=\{r_1^j,...,r_k^j,\ldots,r_l^j\} \text{ , where } r_k^j = \sup_x \min\left(f_{p_k^u}(x), f_{v_k^j}(x)\right),$$

we shall use the inclusion measure M2 in order to measure how much covers R_i^u to R_j^u, for all *i* and *j*. This value will express the preference degree, q_{ij}, of R_i^u over R_j^u, and it is computed as:

$$q_{ij} = S(R_j^u, R_i^u) = \inf_x \min\left(1 - f_{R_j^u}(x) + f_{R_i^u}(x), 1\right)$$

Applying this process for all the possible pairs, we obtain the fuzzy preference relation $Q=[q_{ij}]$.

$$Q = \begin{pmatrix} q_{11} & \cdots & q_{1j} & \cdots & q_{1n} \\ \vdots & & \vdots & & \vdots \\ q_{i1} & \cdots & q_{ij} & \cdots & q_{in} \\ \vdots & & \vdots & & \vdots \\ q_{n1} & \cdots & q_{nj} & \cdots & q_{nn} \end{pmatrix}$$

Example:given two fuzzy sets R_i^u*={0, 0.5, 1, 0.5, 0}* and R_j^u*={0, 0.5, 0.5, 0.5, 0}* corresponding to the resemblance measures between the user profile P_u and the products features F_i and F_j respectively, we can obtain the next preferences degrees (the symbol ^ stands for the operator *min*):

$q_{ij} = S(R_j^u, R_i^u) = \inf_x \min\left(1 - f_{R_j^u}(x) + f_{R_i^u}(x), 1\right)=$

((1-0+0)^1)^((1-0.5+0.5)^1)^((1-0.5+1)^1)^((1-0.5+0.5)^1)^((1-0+0)^1)=1

$q_{ji} = S(R_i^u, R_j^u) = \inf_x \min\left(1 - f_{R_i^u}(x) + f_{R_j^u}(x), 1\right)=$

((1-0+0)^1)^((1-0.5+0.5)^1)^((1-1+0.5)^1)^((1-0.5+0.5)^1)^((1-0+0)^1)=0.5

So, the preference degree of R_i^u over R_j^u , $q_{ij} = 1$, whereas the degree of R_j^u over R_i^u, $q_{ji} = 0.5$.

3.3.2 Computing the Non Dominance Degree

To rank the items in order to be recommended we have built a preference relation, so we shall apply a choice degree to order the items according to its similarity with the customer profile. In

[Orl78] we can find different choice degrees, in our model we have chosen the Non Dominance Degree, that indicates which item is non dominated by the other ones.

Definition 2 [Orl78]**.** Let $Q=[q_{ij}]$ be a fuzzy preference relation defined over a set of alternatives X. For the alternative x_i its non-dominance degree, NDD_i, is obtained as

$$NDD_i = \min_{X_j} \left\{ 1 - q^s_{ji},\ j \neq i \right\}$$

where $q^s_{ji} = \max\left(q_{ji} - q_{ij}, 0\right)$ represents the degree to which x_i is strictly dominated by x_j.

Now, our aim is to obtain the non-dominance degree (*NDD*) of every alternative product, according to the *Definition 2*. For this purpose, it is necessary to build the *strict preference relation:*

$$Q^s = [q^s_{ij}] \text{ where } q^s_{ij} = max(q_{ij}-q_{ji},0)$$

Going on the previous example, the strict preference relation between the two fuzzy sets are:

$$q^s_{ij}=max(q_{ij}-q_{ji},0)=max(1-0.5,0)=0.5$$
$$q^s_{ji}=max(q_{ji}-q_{ij},0)=max(0.5-1,0)=0$$

Finally, we have to compute the *NDD* for each product a_i as:

$$NDD_i = \min_{i \neq j} \left\{ 1 - q^s_{ji} \right\}$$

3.3.3 Ranking the items in order to their recommendation

Eventually the best products or alternatives are those with a greater *NDD,* i.e., the alternatives less dominated by the other ones. We must take into account that there can be several alternatives with the same NDD. These ones will occupy the same order in the ranking.

Example:Let's suppose we have computed the non-dominance choice degree of each alternative:

$$\left\{ NDD_1 = 0.49, NDD_2 = 1, NDD_3 = 0.48, NDD_4 = 1 \right\}$$

So, the solution is a ranking where NDD_2 and NDD_4 are the best alternatives. After them we find NDD_1 and finally, the worst is NDD_3.

A general scheme of the recommendation process carried out by this recommendation model can be seen in the Fig. 4.

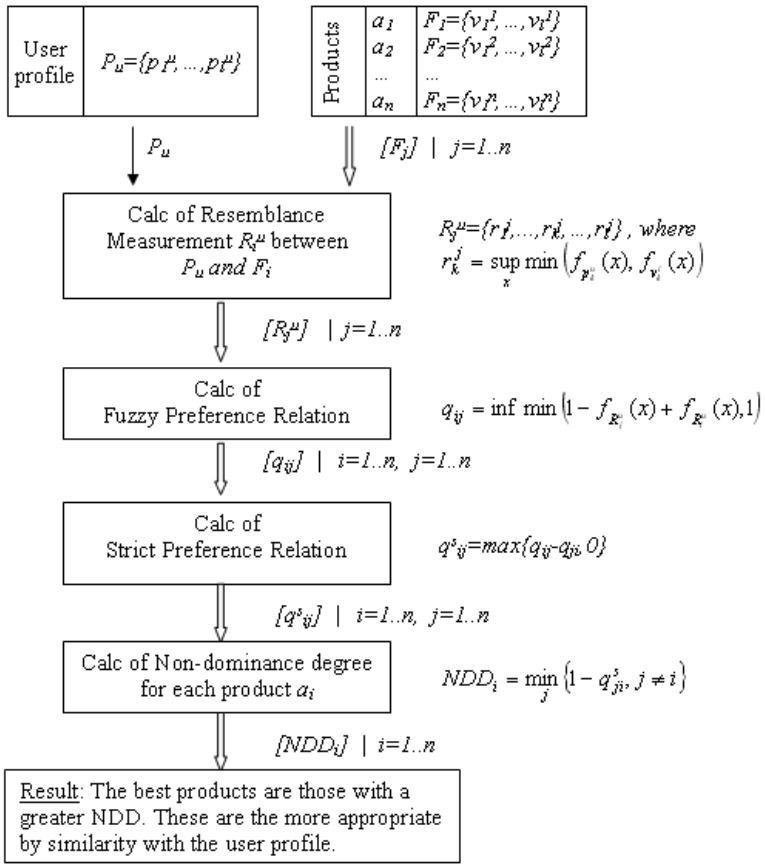


Fig. 4: Content-Based Recommendation model in great detail

4 Example of a recommendation process

To chose the finest toy/s for a child is a really hard task, because each child is different and needs different aspects to improve his verbal skills, reasoning, athletic ability,… In this section we are going to apply our based content representation model to the process of chosing a suitable toy for a child in a toy shop.

The recommendation model is guided by several criteria that describe the features of the toys, in this example each criterion will be assessed in a linguistic term set (see Table 2) according to the knowledge that the experts have about them:

- *Independent play*: this learning parameter promotes self-esteem and confidence in children by empowering them with choices and by organizing stimulating play activities. It will be assessed in the linguistic term set C.
- *Mathematical play*: it measures if the children is involves in problem solving activities, reasoning, and sequencing. This helps the child to acquire an understanding of basic math skills and develop fine motor skills. It will be assessed in the linguistic term set B.
- *Musical play*: the toy engages children in rhythmic musical activity. There are studies that prove that music enhances reading, math, and creative skills. It will be assessed in the linguistic term set B.
- *Linguistic play*: it encourages a child's verbal skill. It will be assessed in the linguistic term set B.

- *Motor skill*: it helps to promote and develop children's physical athletic ability, manual dexterity, and/or eye-hand coordination. It will be assessed in the linguistic term set C.
- *Cooperative play*: it improves cooperation and interaction with the objective of achieving common goals. It will be assessed in the linguistic term set B
- *Visual play*: it stimulates the child in visual evaluation and activities that enhance creativity. It will be assessed in the linguistic term set C.
- *Easy to learn how to play*: Some games need more time than others to learn how to play it or need the help of an adult to be played. It will be assessed in the linguistic term set A.

These parameters are also used to describe the user profile to simplify the example we use the same linguistic term sets for each one, but can be different ones.

The semantics of the linguistic term sets are showed in the Table 2 and in the Fig. 5:

Linguistic term set A		**Linguistic term set B**		**Linguistic term set C**	
Difficult (D)	(0,0,0.5)	Very basic (VB)	(0,0,0.25)	Nothing (N)	(0,0,0.16)
Suitable (S)	(0,0.5,1)	Basic (B)	(0,0.25,0.5)	A little (LT)	(0,0.16,0.33)
Easy (E)	(0.5,1,1)	Normal (N)	(0.25,0.5,0.75)	Less than average (LA)	(0.16,0.33,0.5)
		Advanced (A)	(0.5,0.75,1)	Average (AV)	(0.33,0.5,0.66)
		Very Advanced (VA)	(0.75,1,1)	More than Average (MA)	(0.5,0.66,0.83)
				A lot (AL)	(0.66,0.83,1)
				All (A)	(0.83,1,1)

Table **2**: Semantic of the linguistic term sets A, B and C

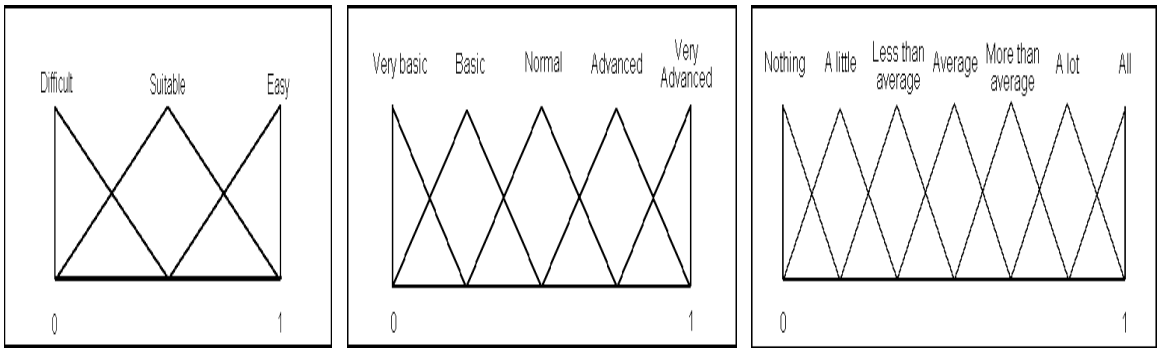


Figure **5**: The linguistic term sets A,B,C

In the Table 3, we can see the item database that we use in this example:

Toy	**Independent play**	**Mathematical play**	**Musical play**	**Linguistic play**	**Motor skill**	**Coopeerat. Play**	**Visual Play**	**Learning**
T_1	N	VB	VB	N	AV	VB	MA	D
T_2	LT	B	VA	VB	AV	B	AL	D
T_3	AV	B	B	A	LA	N	A	S
T_4	LA	N	N	N	A	A	AV	S
T_5	AV	A	VA	A	AL	VA	LA	D
T_6	AV	VB	N	N	MA	N	N	S
T_7	MA	N	N	VA	AV	A	AV	S
T_8	AL	VA	N	N	N	N	AV	S
T_9	N	N	B	VA	N	VA	N	D
T_{10}	LT	A	N	N	MA	N	AL	D

Table 3: Descriptions of toys of our Recommendation System

The process to recommend a toy for a customer follows the process presented in the before section.

1. Acquisition of the user profiles.

A user provides his profile in order to obtain a recommendation according to his necessities:

Independent play	Mathematical play	Musical play	Linguistic play	Motor skill	Cooperative play	Visual play	Learning
AV	B	B	A	AV	N	A	S

Table **4**: user profile

With this information our recommendation model will find those toys that are closer to the user necessities.

2. Matching Items for each user

The first step in our process is to find the similarity between the user profile and every toy, by means of matching process presented in 3.2.2 (see Table 5):

T_1	T_2	T_3
$R^u_{T_1} = (0,0.5,0.5,0.5,1,0,0,0.5)$	$R^u_{T_2} = (0,1,0,0,1,0.5,0.5,0.5)$	$R^u_{T_3} = (1,1,1,1,0.5,1,1,1)$

T_4	T_5	T_6
$R^u_{T_4} = (0.5,0.5,0.5,0.5,0,0.5,0,1)$	$R^u_{T_5} = (1,0,0,1,0,0,0,0.5)$	$R^u_{T_6} = (1,0.5,0.5,0.5,0.5,1,0,1)$

T_7	T_8	T_9
$R^u_{T_7} = (0.5,0.5,0.5,0.5,1,0.5,0,1)$	$R^u_{T_8} = (0,0,0.5,0.5,0,1,0,1)$	$R^u_{T_9} = (0,0.5,1,0.5,0,0,0,0.5)$

T_{10}
$R^u_{T_{10}} = (0,0,0.5,0.5,0.5,1,0.5,0.5)$

Table **5**: Similarity degree between user profile and every toy

Where, for example, T_1, is calculated using the similarity function, $D(A,B) = \sup_x \min(f_A(x), f_B(x))$, between T_0 and the user profile U:

$$Similarity(T_1, U) = \{D_{Independent\ Play}(N, AV), \ldots, D_{learning}(D, S)\} =$$
$$= S_{T_1} = (0,0.5,0.5,0.5,1,0,0,0.5)$$

3. Making a recommendation

The last stage is to make a recommendation. To do so, first the model computes a fuzzy preference relation, *Q*, such as it was shown in the section 3.3.1.(see Fig. 6).

$$Q=\begin{pmatrix} - & 0.5 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0.5 & - & 0 & 0.5 & 0 & 0 & 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & - & 1 & 1 & 1 & 0.5 & 1 & 1 & 1 \\ 0 & 0 & 0 & - & 0.5 & 0.5 & 0 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & - & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 1 & 0.5 & - & 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 1 & 0.5 & 0.5 & - & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & - & 0.5 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & - & 0 \\ 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0.5 & 0.5 & - \end{pmatrix} \quad Q^S=\begin{pmatrix} - & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & - & 0 & 0.5 & 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & - & 1 & 1 & 1 & 0.5 & 1 & 1 & 1 \\ 0 & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0.5 & 0.5 & - & 0 & 1 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 & 0.5 & 0 & - & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & - \end{pmatrix}$$

Figure 6: Fuzzy preference relation Q and the strict preference relation Q^s.

Where q_{12} represents the preference degree of the toy T_1 over T_2 (or how much similarity degree of T_1 covers T_2).

For example, we show how the preference degrees q_{12} and q_{13} are computed:

$$q_{12}=\inf_x \min\left(1-f_{T_2}(x)+f_{T_1}(x),1\right)=\left((1-0+0)\wedge 1\right)\wedge\left((1-1+0.5)\wedge 1\right)\wedge\left((1-0+0.5)\wedge 1\right)\wedge$$
$$\wedge\left((1-0+0.5)\wedge 1\right)\wedge\left((1-1+1)\wedge 1\right)\wedge\left((1-0.5+0)\wedge 1\right)\wedge\left((1-0.5+0)\wedge 1\right)\wedge\left((1-0.5+0.5)\wedge 1\right)=0.5$$

$$q_{13}=\inf_x \min\left(1-f_{T_3}(x)+f_{T_1}(x),1\right)=\left((1-1+0)\wedge 1\right)\wedge\left((1-1+0.5)\wedge 1\right)\wedge\left((1-1+0.5)\wedge 1\right)\wedge$$
$$\wedge\left((1-1+0.5)\wedge 1\right)\wedge\left((1-0.5+1)\wedge 1\right)\wedge\left((1-1+0)\wedge 1\right)\wedge\left((1-1+0)\wedge 1\right)\wedge\left((1-1+0.5)\wedge 1\right)=0$$

Where $\wedge$ stands for "min".

Now, for each toy T_i, the model calculates its non-dominance degree NDD_i. First, the strict preference relation Q^s is computed (Fig. 6).

Then, we compute the non-dominance choice degree of each toy:

$\{NDD_1=0.5, NDD_2=0.5, NDD_3=1, NDD_4=0, NDD_5=0, NDD_6=0, NDD_7=0.5, NDD_8=0$
$\text{NDD}_9=0, NDD_{10}=0\}$,

where,

$$NDD_1=\min\{(1-0),(1-0.5),(1-0),(1-0),(1-0.5),(1-0),(1-0),(1-0),(1-0.5)\}=0.5$$

Finally, the model ranks the toys using the non-dominance choice degree (Table 6):

First Level	Second Level	Third Level
T_3	T_1	T_4
	T_2	T_5
	T_7	T_6
		T_8
		T_9
		T_{10}

Table 6: Ranking of the toys.

The most suitable toy to recommend according to the user profile is T_3, after T_3 we could recommend T_1, T_2 and T_7, and the least suitable toys to recommend are T_4, T_5, T_6, T_8, T_9 , T_{10}.

5 Concluding Remarks

In this paper, we have presented a recommendation model that can be applied in decision making problem when we have at one's disposal some linguistic information descriptive of every alternative. In addition, the user supplies his preferences over each attribute that describe the alternatives given a user profile. The model matches the user preferences with the description of every product obtaining a resemblance index.

In further works we are going to consider not only the matching between user profile and product features but the distance between these sets. Moreover, we will research the performance of other indices to evaluate the preference between two fuzzy sets, and other indices for establishing choice degrees in a fuzzy preference relation.

Acknowledgments

This work has been partially supported by the Research Project TIC 2002-03348.

References

[**Ans00**] A. Ansari, S. Essegaier, and R. Kohli. *Internet Recommendation Systems*. Journal of Marketing Research XXXVII (2000) 363- 375.

[**Bas98**] C. Basu, H. Hirsh, and W. Cohen. *Recommendation as CIassification: Using Social and Content-Based Information in Recommendation*. Proceedings of the Fifteenth National Conference on Artificial Intelligence, (1998) 714-720.

[**Bon86**] P.P. Bonissone and K.S. Decker. *Selecting uncertainty calculi and granularity: an experiment in trading-off precision and complexity*. In: L.H. Kanal, J.F. Lemmer, (Eds.), Uncertainty in
Artificial Intelligence, North-Holland, Amsterdam, (1986) 217-247.

[**Bor93**] G. Bordogna and G. Passi. *A fuzzy linguistic approach generalizing boolean information retrieval: a model and its evaluation*. J. Amer. Soc. Inform. Sci. 44 (1993) 70-82.

[**Bre98**] J. Breese, D. Heckerman, and C. Kadie. *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. Technical report MSR-TR-98-12, Microsoft Research. (1998).

[**Bou95**] B. Bouchon-Meunier and M. Rifqi. *Resemblance in database utilization.* 6th IFSA World Congress, 1995.

[**Bou96**] B. Bouchon-Meunier and M. Rifqi, and S. Bothorel. *Towards general measures of comparison of objects*. Fuzzy Sets and Systems, (84):143-153, 1996.

[**Del92**] M. Delgado, J.L. Verdegay and M.A. Vila. *Linguistic Decision Making Models*. Int. J. of Intelligent Systems 7 (1992) 479-492.

[**Dub80**] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. (Academic Press, New York, 1980)

[**Dub83**] D.Dubois and H.Prade. *Ranking fuzzy numbers in the setting of possibility theory*. Information Sciences 30 (1983) 183-224.

[**Gol92**] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. 1992. *Using Collaborative Filtering to Weave*. Communications of the ACM, 35(12) (1992) 61-70.

[**Hay01**] C. Hayes and P. Cunningham. 2001. *Smart Radio - Community Based Music Radio*. Knowledge-Based Systems 14 (3-4): 197-201.

[**Her95**] F. Herrera, E. Herrera-Viedma and J.L. Verdegay. *A sequential selection process in group decision making with linguistic assessment*. Inform. Sci., 85 (1995) 223-239.
[Her00] F. Herrera, E. Herrera-Viedma, L. Martínez. *A fusion approach for managing multi-granularity linguistic term sets in decision making*. Fuzzy Sets and Systems 114 43-58 (2000).
[**Vie04**] Herrera-Viedma, E., F. Herrera, L. Martinez, J.C. Herrera, and A.G. Lopez. 2004. *Incorporating Filtering Techniques in a Fuzzy Linguistic Multi-Agent Model for Gathering of Information on the Web*. Fuzzy Sets and Systems 148 (1) (2004) 61-83.
[**Jen92**] A. Jennings, and H. Higuchi. *A Personal News Service Based on a User Model Neural Network*. IEICE Transactions on Information and Systems E75-D (2) (1992) 198-209.
[**Kau98**] H. Kautz. *Recommender Systems*. AAAI Press, Menlo Park, CA. 1998.
[**Kon97**] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. *Grouplens: Applying Collaborative Filtering to Usenet News*. Communications of the ACM 40 (3) (1997) 77-87.
[**Lie95**] H. Lieberman. *Letizia: An Agent That Assists Web Browsing*. In Proceedings of the 1995 International Joint Conference on Artificial Intelligence. Montreal, Canada (1995).
[**Orl78**] S.A. Orlovski, *Decision Making with a Fuzzy Preference Relation*, Fuzzy Sets and Systems 1 (1978) 155-167
[**Paz96**] M. Pazzani, J. Muramatsu, and D. Billsus. *Syskill & Webert: Identifying Interesting Web Sites*. In Proceedings of the Thirteenth National Conference on Artificial Intelligence AA.AI, no. 96 (1996) 54-61.
[**Per99**] P. Perny, and J. D. Zucker. *Collaborative Filtering Methods Based on Fuzzy Preference Relations*. EUROFUSE-SIC 99 (1999). Budapest.
[**Pop01**] A. Popescul, L. II. Ungar, D. M. Pennock, and S. Lawrence. *Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments*. In Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI) (2001), San Francisco, 437-444.
[**Res97**] P. Resnick and H.R. Varian. *Recommender Systems*. Communications of the ACM 40 (1997) 56-58.
[**Rif00**] M.Rifqi, V.Berger and B.Bouchon-Meunier. *Discrimination power of measures of comparison*. Fuzzy Sets and Systems 110 (2000) 189-196.
[**San79**] E. Sanchez. *Inverses of fuzzy relations, applications to possibility distributions and medical diagnosis*. Fuzzy Sets and Systems 2 (1979) 75-86
[**Sch01**] J.B. Schafer, J.A. Konstan and J. Riedl. *E-Commerce Recommendation Applications*. Data Mining and Knowledge Discovery, no. 5 (2001) 115-153.
[**Sha95**] U. Shardanand. *Social Information Filtering: Algorithms for Automating "Word of Mouth*. Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems (1995), Denver, 210-217.
[**Yag95**] R. R. Yager, *An approach to ordinal decision making*, Int. J. Approximate Reasoning, no. 12, (1995) 233–261.
[**Zad75**] L.A. Zadeh. *The concept of a linguistic variable and its application to approximate reasoning*. Information Science (8 and 9) (1975): (Part I and II) 8, pp 199-249 and pp. 301-357, (Part III) 9, pp, 43-80.

D.7. Knowledge Based Recommender Systems Models

In our research, we realized that although the previous model was correct and useful for its aim, other models were developed in the e-commerce field to make recommendations when there is no historical information or it is irrelevant, the knowledge based recommender systems.

In this section, we will present two knowledge based recommender systems models. The first one, called RML, is a classic knowledge based recommender system model (see figure D.3). The aims of this model are:

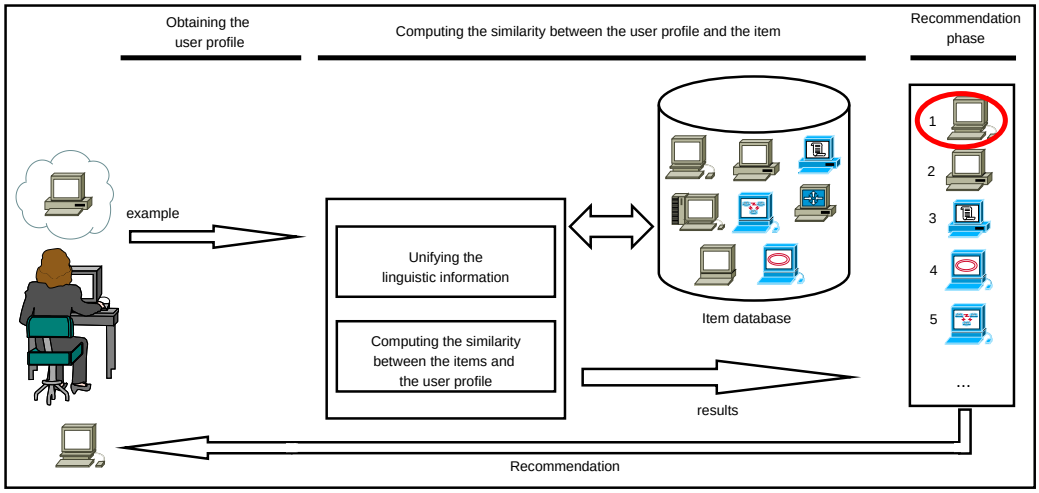
- To make easier the user preference gathering process. This user profile is obtained through an example that the user provides. If the user profile does not represent faithfully the user's necessities, it can be changed throughout a refinement phase. This user profile is more complete and it is obtained faster than in the previous model.

- To compute in a more accurate way how close are two linguistic assessments. To find suitable items for user, it is needed to compare the user profile with the description of the items. Therefore, the more accurate comparison computations are made, the better results the system will obtain.

- To decrease the computation load. The computations of this model to make recommendations are simpler.

A further description of this model is attached following in the paper *A Knowledge Based Recommender System with Multigranular Linguistic Information* published in the International Journal of Computational Intelligence Systems. In this

Figure D.3: Knowledge based recommender model with multigranular information



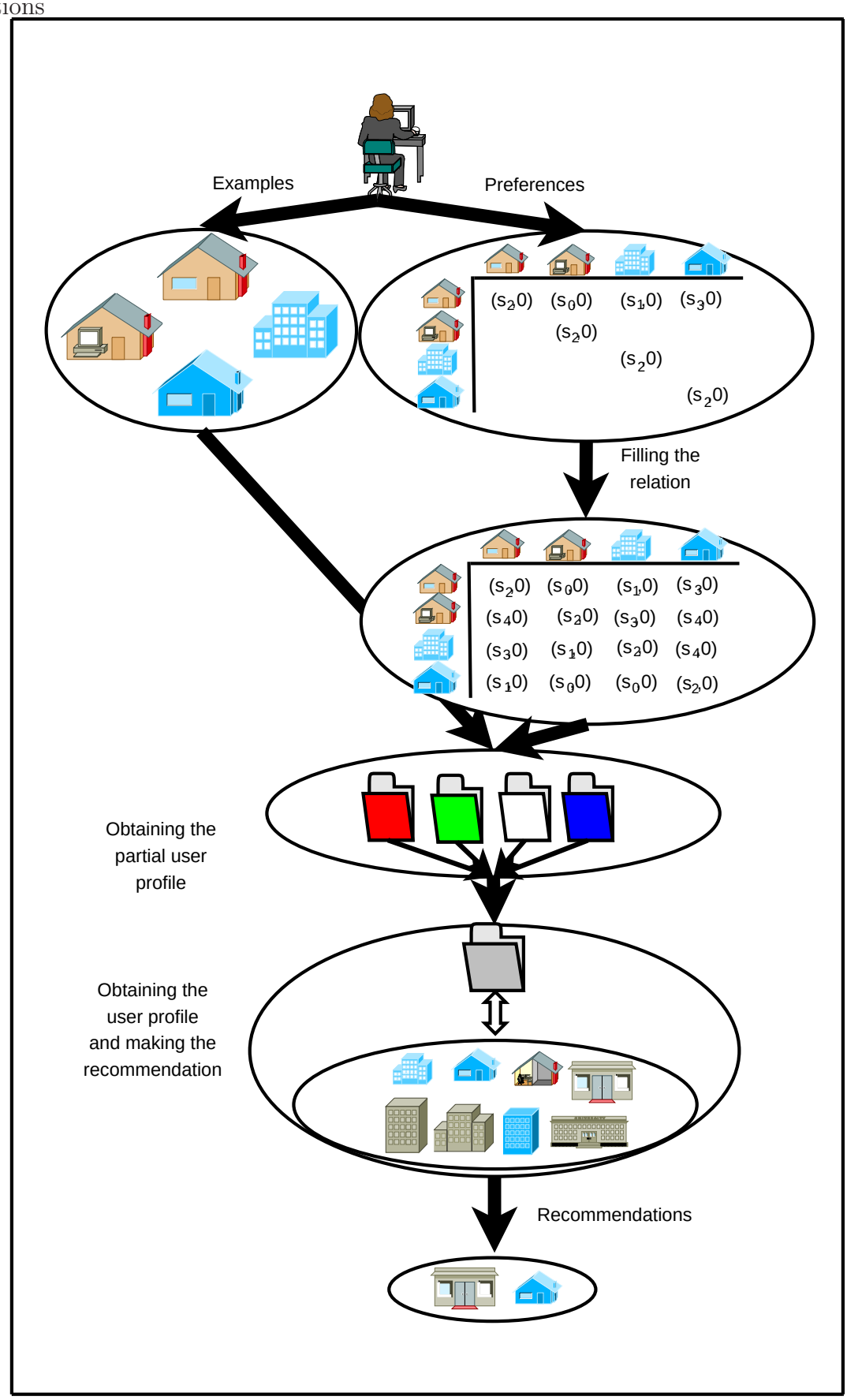
document, it is introduced the context and background necessary to understand the proposed model and finally such a model is described in depth.

The second model (see figure D.4), known as RRPI, is an improvement of the previous one, whose aim is to enhance the mechanisms used to build the user profile in order to obtain a closer profile to the user necessities without requiring the user to provide much information. This model expects users to provide examples of their necessities and a preference relation over them. The main advantages of this model are:

- The refinement phase is avoided.

- We have reduced the dependence between the quality of the recommendations, and the selection of the example. In the previous knowledge based recommender system model if the example was not well-chosen and the user did not use correctly the refinement phase, it would not make accurate recommendations. In this model, this dependence is lower, and therefore, it is likely that it will make better recommendations.

Figure D.4: Knowledge based recommender system model with preference relations



A description in depth of this model is attached following in the paper *Improving the Effectiveness of Knowledge Based Recommender Systems Using Incomplete Linguistic Preference Relations* accepted in the International Journal of Uncertainity, Fuzziness and Knowledge-based Systems,that explains it in detail, the necessary background and how this model makes the recommendations.

International Journal of Computational Intelligence Systems, Vol. 0, No. 0 (April, 2000) 00–00

A KNOWLEDGE BASED RECOMMENDER SYSTEM WITH MULTIGRANULAR LINGUISTIC INFORMATION

Luis Martínez, Manuel J. Barranco, Luis G. Pérez, Macarena Espinilla
Dpt. of Computer Science, University of Jaen
23071 - Jaen, Spain
E-mail: martin,barrancolgonzaga,mestevez@ujaen.es

Received (to be inserted
Revised by Publisher)

Recommender systems are applications that have emerged in the e-commerce area in order to assist users in their searches in electronic shops. These shops usually offer a wide range of items that cover the necessities of a great variety of users. Nevertheless, searching in such a wide range of items could be a very difficult and time-consuming task. Recommender systems assist users to find out suitable items by means of recommendations based on information provided by different sources such as: other users, experts, item features, etc. Most of the recommender systems force users to provide their preferences or necessities using an unique numerical scale of information fixed in advance. In spite of this information is usually related to opinions, tastes and perceptions, therefore, it seems that is usually better expressed in a qualitative way, with linguistic terms, than in a quantitative way, with precise numbers. We propose a Knowledge Based Recommender System that uses the fuzzy linguistic approach to define a flexible framework to capture the uncertainty of the user's preferences. Thus, this framework will allow users to express their necessities in scales closer to their own knowledge, and different from the scale utilized to describe the items.

1. Introduction

One of the main problems users face when they are surfing in Internet is the vast quantity of information, being most of it useless. For instance, most of the e-shops offer thousand of products that conform a search space that the user can not evaluate carefully in order to find out the most suitable products according to his/her necessities. In such cases, users can feel disappointed because they do not find what they really want among so huge amount of alternatives despite wasting much time. Different e-services have risen to help them to reach easy by quickly the products that meet their necessities. In this paper, we focus in recommender systems, a class of software [5,21,22] that has emerged in the last years within E-Commerce area [23]. Its aim is to assist users in their searchs in order to find out the most suitable item/s according to their preferences, necessities or tastes. To do so, the systems provide recommendations and/or hide or remove the useless information.

Essentially, all the recommender systems follow the same steps to make recommendations: first the systems gather preference information from users, experts, etc., related to their preferences, tastes, and opinions such that using this information they rank the items and make recommendations about which items are more attractive for them. Depending on the information gathered by the system, and the technique that ranks the items to suggest recommendations, the recommender systems can be classified in different types:

- *Demographic recommender systems* [16]*:* They classify users into demographic groups, using personal attributes. A user will receive recom-

mendations according to the group in which was classified.

- *Content-based recommender systems* [20]*:* This type of recommender systems compute recommendations according to the past user behaviour as well as to the features of items that the user liked before.

- *Collaborative filtering recommender systems* [8]*:* They gather ratings about products instead of item features and group the users according to their similarity. The recommendations are computed by means of a prediction about how much does a user like an item taking into account the other group members ratings.

- *Knowledge based recommender systems* [4]*:* These systems compute their recommendations using case based reasoning processes, i.e., the users provide an example similar to his/her aims and the systems infers a profile in order to find the better match product in the search space.

- *Utility based recommender systems* [9]*:* They compute recommendations based on the calculation of the utility of each item according with the user interests.

- *Hybrid recommender systems* [1,5]*:* This kind of recommender systems arise with the aim of solving drawbacks that the other systems present in some situations. In order to do that, these systems combine different techniques of the before recommender systems.

All these types of recommender systems have been developed and applied to different situations, being the most used the content-based and collaborative ones. However these systems are not always successful because they need big amount of users and items information to obtain good results and it is not always available. Therefore, different solutions have been proposed to avoid unsuccessful recommendations when there is not information available such as hybridization that can be useful in several cases but not always, or the use of knowledge based recommender systems, when there is not user information available. In this paper we focus in the last type of recommender systems.

The information gathered by recommender systems is usually vague and incomplete because it is related to users' own perceptions. In spite of this fact, most of Recommender systems force their users to provide the information in a numerical scale fixed a priori [10] . This obligation implies a lack of expressiveness and hence a lack of precision in the suggested recommendations. To overcome this drawback, our proposal for a Knowledge Based Recommender System will offer the users the possibility of expressing their preference information using linguistic assessments instead of numerical ones, since the linguistic information is usually more suitable to assess qualitative information (human perceptions, taste, necessities) [17,25]. In addition, our system allows the users to use their own linguistic term set to express their preferences according to their knowledge about the items. Thus, the context on which the recommendations are computed is a multi-granular linguistic context [11,13,14]

In this paper we present a Knowledge based recommender system that will filter and recommend the closest items to the user's necessities computing the similarity among the descriptions of the items and the user profile inferred from the examples provided by the user according to their necessities. The system accomplishes the following steps to make the recommendations:

1 *Profiling process:* the user profile is an information structure that express the user preferences (necessities, tastes, etc). In this phase, the system infers the user profile from an example provided by the user.
2 *Recommendation process:* To find out the most suitable items for the users, the system will measure the similarity between the user profile and the items of the item database. The system ranks these items according to the similarity measure and recommends the most similar ones.

This paper is structured as follow. In section 2 we review some preliminaries about recommender systems and linguistic information that are useful to understand our proposal. In section 3 we present our linguistic multigranular knowledge based recommender system. In section 4 we show, by means of an example, how this system works. Finally, in section 5 some conclusions are pointed out.

2. Preliminaries

Before introducing our proposal, we shall review, firstly, the drawbacks that classical recommender systems present. Secondly we shall make a brief review of knowledge base recommender systems. And finally we review the Fuzzy Linguistic Approach, which is needed in our proposal in order to model the vague and imprecise information provided by humans.

2.1. *Drawbacks of Classical Recommender Systems*

Recommender systems use [5]:

(i) Background data: the information that the system has before the recommendation process begins,
(ii) Input data: the information that user must communicate to the system in order to generate a recommendation, and
(iii) An algorithm that combines background and input data to achieve the suggestions.

Classical recommender systems, both the Collaborative and Content-based ones, need a great amount of information about the users to exploit the background and the input data. Sometimes, this is an important drawback, mainly when a new user accesses to the system ('ramp-up' problem). When collaborative filtering recommender systems have not background data about the target user, it is not possible to compare him/her with other users in order to make recommendations. In a similar manner, content-based ones do not work properly in this situation because they are not able to build the user profile when background information about the user does not exist.

Also, we can see the 'ramp-up' problem when a new item is added in collaborative filtering system. In this case, the new item could be interesting to be recommended to some user, but the system does not use it because it has not any rating about the item yet. Therefore, this item will not be taken into account in the recommendation process until it has received enough votes. Due to the fact that it is not being recommended, it is not likely that the item receives enough rating to be recommended.

Other problem we observed in collaborative recommenders is the 'gray sheep' problem. It occurs when a user falls on a border between existing groups of users. The target user is equally similar to two or more groups of users and, hence, the recommendation he/she will receive may be inaccurate.

The need of a large historical data set to ensure successful results is also an important requirement for both collaborative filtering and content-based recommender systems. These systems do not work properly when the historical data set is too small or sparse because, in this case, the probability of matching between the target user and other users will be low.

In order to overcome these drawbacks, some proposals have been given, such as hybrid recommender systems and Knowledge Based Recommender Systems.

In this paper we present a Knowledge Based Recommender Systems that deals with linguistic information. In the next subsection we shall explain the classical knowledge based recommender system in further details.

2.2. *Knowledge Based Recommender Systems*

Our interest in this paper is the Knowledge Based Recommender Systems [4]. These systems use case based reasoning [15] to make recommendations, i.e., they work starting with an example that the user points out, according with his/her tastes or necessities, from which the system infers a user profile utilized to find the best match products in the search space. For reaching its purpose, this system matches the user profile and the possible recommendations. So, when the user provides some knowledge about his/her needs or preferences, the recommender system is able to make recommendations using such matching process. This "user knowledge" can be any knowledge structure that allows to build a user profile. The simplest case could be that the user chooses, among all the available products, one of them that acts as an example of his/her necessities or tastes. These systems manage three types of knowledge:

- Catalog knowledge: It is the knowledge that the Recommender System has about the products and their features. For example, a recommender system for new cars needs to know the characteristics of a car X: safety,

comfort, price, velocity, fuel-consumption, etc.

- Functional knowledge: The system needs some knowledge to relate products to the the user's necessities. For example, it is important to know that a need for a travelling salesman may be a car with low fuel-consumption and very high safety.

- User's knowledge: To provide good recommendations, the system needs to gather information about the user profile. For example, the user chooses a car which satisfies, more or less, his/her expectations. Then, the system will use this information to build an initial user profile consisted of the features of the chosen car.

A good example of this kind or recommender systems may be "The restaurant recommender Entree"[*6]. This recommender system makes its recommendations by finding restaurants in a new city similar to restaurants the user knows and likes. The system allows users to navigate by stating their preferences with respect to a given restaurant, thereby refining their search criteria.

Knowledge based recommender systems are specially suitable for casual searching, when the information about the user does not exist or is scarce. Other systems have a period of start-up until the system gathers historical information about the user. The quality of the recommendations during this period is low. Knowledge based ones do not suffer this drawback because they do not need such kind of historical information. They work very well with only a small amount of knowledge about the user. In a typical situation, the user knowledge is stated by the user by means of an example of his/her necessities that he/she must point out. Starting with this example, the system builds a user profile and searches the products that best fit with this profile. So, we can enumerate the following advantages of this kind of systems:

(i) They do not suffer ramp-up problems (both "new user" and "new item" problems).
(ii) The "gray sheep" problem does not appear in these systems.
(iii) They do not depend on large historical data set

On the other hand, these systems present two disadvantages related to the gathering of user knowledge:

(i) When the amount of products is very large, the process of providing an example to express the user necessities may be a hard task. It may be a reason for the user to give up his/her search because he/she can not easily find a suitable example.
(ii) It is possible that the user does not find an example that fits exactly his/her necessities. So, the system recommends him/her products that perhaps do not satisfy the user. So, it may be another reason for leaving the search too.

In section 3, we present a proposal of a Knowledge Based Recommender System to overcome these drawbacks.

2.3. *Fuzzy Linguistic Approach*

Usually, we work in a quantitative setting, in which the information is expressed by means of numerical values. However, many aspects of different activities in the real world cannot be assessed in a quantitative form, but rather in a qualitative one, i.e., with vague or imprecise knowledge. In that case, a better approach may be to use linguistic assessments instead of numerical values. The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables [24]. This approach is adequate in some situations in which the information may be unquantifiable due to its nature, and thus, it may be stated only in linguistic terms.

In our proposal, the system deals with information that is related to user's tastes, preferences or qualitative features of the items such as *Safety* or *Comfort*. Although, this information could have been modelled by means of numerical values, we have considered using linguistic assessment since this information is vague, imprecise and will be better expressed with the Fuzzy Linguistic Approach.

[*] http://archive.ics.uci.edu/beta/datasets/Entree+Chicago+Recommendation+Data

One possibility of generating the linguistic term set consists of directly supplying the term set by considering all terms distributed on a scale on which a total order is defined. For example, a set of seven terms S, could be given as follows:

$$\{s_0 : N, s_1 : VL, S_2 : L, S_3 : M, S_4 : H, \\ S_5 : VH, S_6 : P\}$$

In these cases, it is required that there exists:

- A negation operator $Neg(s_i) = s_j$ such that $j = g - i$ (g+1 is the cardinality).
- A min and a max operator in the linguistic term set: $s_i \leq s_j \Leftrightarrow i \leq j$.

The semantics of the terms are given by fuzzy numbers defined in the $[0,1]$ interval. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function [2]. The linguistic assessments given by the users are just approximate ones, some authors consider that the linear trapezoidal membership functions are good enough to capture the vagueness of those linguistic assessments.

This parametric representation is achieved by the 4-tuple (a,b,d,c), in which b and d indicate the interval in which the membership value is 1, with a and c indicating the left and right limits of the definition domain of the trapezoidal membership function [2]. A particular case of this type of representation are the linguistic assessments whose membership functions are triangular, i.e., $b = d$, so we represent this type of membership function by a 3-tuple (a,b,c). An example may be the figure 1:

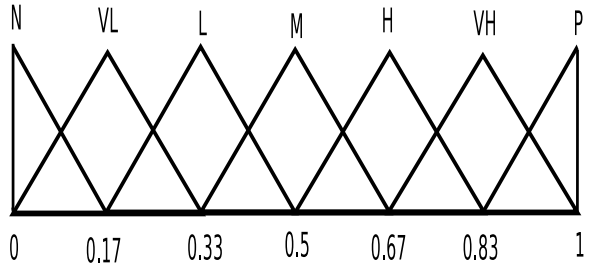


Fig. 1. A linguistic term set and its semantic

Other authors use a non-trapezoidal representation, e.g., Gaussian functions [3].

2.4. *Multigranular Linguistic Information*

An important aspect of fuzzy linguistic approach is the granularity of the uncertainty, i.e., the level of discrimination among different counts of uncertainty. Therefore, according to the source of information knowledge it can be chosen different counts of uncertainty. The granularity should be small enough so as not to impose useless precision levels on the users but big enough to allow a discrimination of the assessments in a limited number of degrees [13].

There are two reasons to utilize multigranularity in our proposal:

1 Users with different degrees of knowledge about the products. User with more knowledge can discriminate among more counts of uncertainty than other ones with less knowledge.
2 Different attributes can need different accuracy degrees. When the products are described in order to fit the product database, each feature can be valuated using a different linguistic term set with different granularity.

Typical values of cardinality used in the linguistic models are the odd ones, such as 7 or 9 [19], where the mid term represents an assessment of approximately 0.5, and with the rest of the terms being placed symmetrically around it. In this paper, we shall deal with sources of information with different degrees of knowledge, so each one could use different linguistic term sets with different granularity. We call this context as multi-granular linguistic context [11].

Figures 2 and 3 show two examples of linguistic term sets with different granularity.

$$s_0^1 = Extremely\,low = (0,0,.125)$$
$$s_1^1 = Very\,low = (0,.125,.25)$$
$$s_2^1 = Low = (.125,.25,.375)$$
$$s_3^1 = A\,bit\,low = (.25,.375,.5)$$
$$s_4^1 = Average = (.375,.5,.625)$$
$$s_5^1 = A\,bit\,high = (.5,.625,.75)$$
$$s_6^1 = High = (.625,.75,.875)$$
$$s_7^1 = Very\,high = (.75,.875,1)$$
$$s_8^1 = Extremely\,high = (0.875,1,1)$$

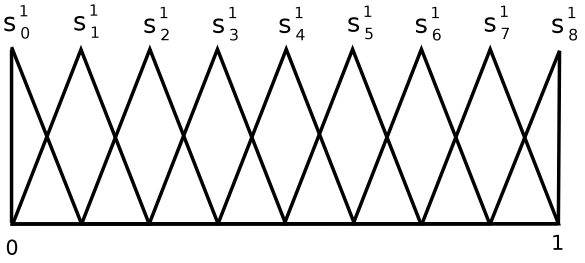


Fig. 2. The linguistic term set S_1

$$s_0^2 = Extremely\,low = (0,0,.16)$$
$$s_1^2 = Very\,low = (0,.16,.33)$$
$$s_2^2 = Low = (.16,.33,.5)$$
$$s_3^2 = Average = (.33,.5,.66)$$
$$s_4^2 = High = (.5,.66,.83)$$
$$s_5^2 = Very\,high = (.66,.83,1)$$
$$s_6^2 = Extremely\,high = (.83,1,1)$$

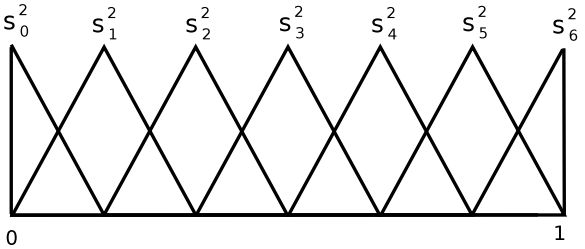


Fig. 3. The linguistic term set S_2

3. A Knowledge Based Recommender System Model with multigranular linguistic information

In this section, we present our proposal for a Knowledge Based Recommender System. This model expects users to provide an example of their preferences (for example, Bella Italia restaurant) in order to generate an initial user profile. So, the initial profile match exactly with the description of the selected item. This profile consists of a vector of features in which each feature is described by a *linguistic label*. Each feature describes a different aspect of the user profile, and therefore, it could be assessed with a different linguistic term set according to the available knowledge about this feature.

Sometimes, the given example does not fix exactly what the user wants since one or several features that have been assessed with linguistic terms that do not fulfil user's expectations or needs. So, the user needs to refine his/her profile by changing some of its linguistic assessments. Suppose he/she agrees with all the features of the given example except one. For example, considering the price as the feature in disagree, next, the user can change the value "low" with the value "very low". In such cases, sometimes it could be suitable to offer the user another linguistic term set closer to him/her knowledge than the one used in the descriptions of the items. With the changes provided by the user, the system will generate the final profile that will be used in the recommendation process.

Therefore, our proposal develops its activity according to the schema (see figure 4).

1 *Profiling process:* The system builds the user profile which contains information concerning the necessities of the user. This phase has two steps:
   (a) *Gathering the preferred example from the user:* The user chooses an item as an example of his/her necessities. The description of this item will define the initial user profile.
   (b) *Casual modification of preferences:* Usually the user does not search an item exactly equal to the given example, but a similar one, with some differences in its attributes. So, in such cases, the user must refine his/her profile by using a set of linguistic terms adequate to his/her knowledge level.

2 *Recommendation process:* The system calculates the similarity between the user profile and the items, and recommends the most suitable ones. This process is composed of the following steps:
   (a) *Unification of the linguistic information:* Due to the fact we are dealing with multigranular linguistic information, it is necessary to unify it in an unique domain called Basic Linguistic Term Set (BLTS).
   (b) *Calculation of the similarity between the user profile and the items:* In order to recommend an item to the user, we need to know how close the items are to the user profile.
   (c) *Providing the recommendation:* This is the final step in which the closest items

to the user necessities will be recommended.

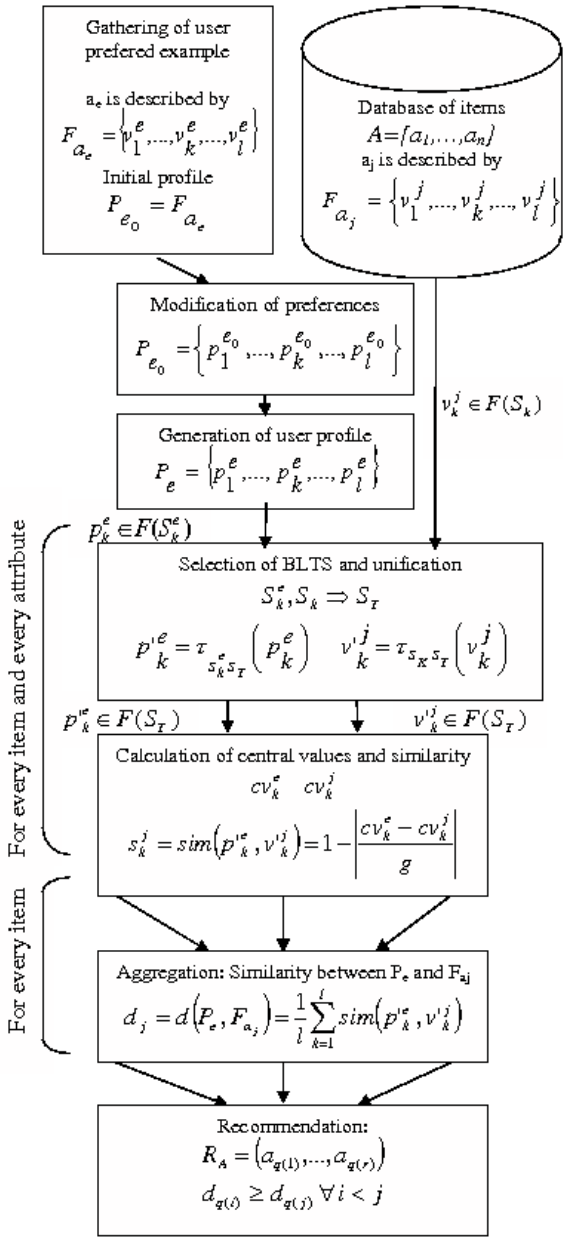


Fig. 4. Recommender System Model

In the following sections we will explain these steps in further detail.

3.1. *Profiling process*

In this phase, the system gathers the user's necessities or preferences in order to know what kind of item is required by the user u_e. The Recommender System has a database $A = \{a_1, \ldots, a_n\}$, with n items, all of them described by means of a set of attributes $\{c_1, \ldots, c_l\}$. Therefore, every item a_j is described by an utility vector $F_{a_j} = \left\{v_1^j, \ldots, v_l^j\right\}$, in which v_k^j is the value of the attribute c_k for the item a_j assessed in the linguistic term set S_k. These descriptions are obtained either directly from experts or from surveys realized about the items. In any case, the information about the attributes can be assessed in a multigranular context that allows to use different linguistic term sets. So, every attribute can be assessed in a different label set according to the existing degree of the knowledge about them. Therefore, the description of the items stored in the database conforms a multigranular linguistic space. Once we know how the items are described in the Recommender System, we can study how to build the user profile.

3.1.1. *Gathering the preferred example from the user*

In our Recommender System, the starting point to define the user necessities is the selection of an example. Let a_e be the item given as an example by the user u_e. This item is described in the database by means of an utility vector $F_e = \left\{v_1^e, \ldots, v_l^e\right\}$, in which $v_k^e \in S_k$ is an assessment for attribute c_k expressed in terms of S_k. This selected example defines a initial user profile that we denote as $P_{e_0} = \left\{p_1^{e_0}, \ldots, p_l^{e_0}\right\}$, where $p_k^{e_0} = v_k^e$. In this initial user profile, the linguistic terms sets are the same than the ones used in the system database.

3.1.2. *Casual modification of preferences*

Once the initial user profile is defined, the system offers the user the possibility of changing one or more values of his/her profile in order to refine the recommendation process. Probably, the knowledge the user has about a given attribute c_k is different to the knowledge that the experts (database's builders) has about the same attribute. So, the linguistic term set used by the experts may not be appropriate for the user. Therefore, the system allows the user to utilize other linguistic term set more suitable to his/her knowledge about the attribute. In this case, for an

attribute c_k, the user can assign a new value, $p_k^{e_1}$, expressed in other linguistic term set, S'_k according to his/her knowledge. Then, we have a final user profile $P_e = \{p_1^e, \ldots, p_l^e\}$ where $p_k^e \in S_k^e$ are obtained in the following way:

(a) $p_k^e = p_k^{e_0}, p_k^{e_0} \in S_k^e = S_k$ if the attribute c_k has not been modified
(b) $p_k^e = p_k^{e_1}, p_k^{e_1} \in S_k^e = S'_k$ otherwise.

In order to make easy this task, the system will provide an easy-use interface that allows to select a suitable the linguistic term set for the attribute to change.

3.2. *Recommendation process*

In this phase, the system computes how close the items are to the user profile by means of measure of resemblance or similarity. To accomplish this phase the system will evaluate the similarity between all the items of the database $A = \{a_1, \ldots, a_n\}$ and the user profile following these steps:

1 *Unify the linguistic information:* because there is no way to deal directly with information that has been assessed in different linguistic term sets, we need to unify the information in a unique domain.
2 *Calculate the similarity between every item and the user profile:* they system computes the similarity degree between the user profile and database items.
3 *Providing the recommendations:* finally, the system suggests to the user the most suitable items, i.e., the closest ones to the user profile.

3.2.1. *Unification of the linguistic information*

In order to manage multigranular information, we must unify it using a unique expression domain [12,13,18]. In this case, we choose as unification domain a Basic Linguistic Term Set (BLTS) that we note as S_T. The information will be unified by means of fuzzy sets defined in the BLTS, $F(S_T)$. The selection of the BLTS is explained in [12].

Following, the system unifies the multi-granular linguistic terms by means of fuzzy sets defined in the BLTS, $F(S_T)$, using the following transformation function:

Definition 1 [12] *Let $A = \{l_0, \ldots, l_p\}$ and $S_T = \{s_0, \ldots, s_g\}$ be two sets of linguistic terms such that $g \geq p$. Then, a function of multigranular transformation τ_{AS_T}, is defined as:*

$$\tau_{AS_T} : A \rightarrow F(S_T)$$
$$\tau_{AS_T}(l_i) = \left\{ \left(s_k, \alpha_k^i\right) | k \in \{0, \ldots, g\} \right\}, \forall l_i \in A$$
$$\alpha_k^i = max_y \, min\{\mu_{l_i}(y), \mu_{s_k}(y)\}$$

where $F(S_T)$ is the set of all the fuzzy sets defined on S_T, and $\mu_{l_i}(y)$ and $\mu_{s_k}(y)$ are membership functions of the fuzzy sets associated to the terms l_i and s_k respectively.

To unify the multigranular linguistic information the system will use the transformation functions $\tau_{S_k S_T}$ in order to express the user profile and the descriptions of the items by means of fuzzy sets defined into fuzzy sets in the BLTS. For instance, an assessment of the user profile, $p_k^e \in S_k^e$, is transformed into a fuzzy set, $p_k'^e$, in which this fuzzy set is described by a tuple of membership degrees $\left(\alpha_{k0}^e, \ldots, \alpha_{kg}^e\right)$.

In the same manner, the descriptions of the items are also transformed into the BLTS. An assessment, $v_k^j \in S_k$, of the item, a_j, is transformed into a fuzzy set, v'^j_k, and it is also represented in the same way $\left(\alpha_{k0}^j, \ldots, \alpha_{kg}^j\right)$.

Once all the information is expressed in the same expression domain, we can proceed to calculate the similarity between the user profile and item database.

3.2.2. *Calculation of the similarity between the user profile and the items*

Once the information has been unified, the system will look for which items are closer to the user's necessities. To accomplish this step, we need to calculate the similarity between the user profile, P_e, and the item, a_j, of the database using the following function:

$$d_j = d(P_e, a_j) = \frac{1}{l} \sum_{k=1}^{l} w_i sim\left(p'^e_k, v'^j_k\right)$$

where w_i represents the importance of each attribute and $\sum w_i = 1$, being *sim* a function that computes the similarity between the values P_e and a_j

Although initially, we have considered this function, *sim*, to be accomplished by using the Euclidean distance. However, it was discarded because its results are not good (see [14]).

Then, we propose to compute the similarity using a measure based on the use of central values [14] that obtain suitable results according to our aim in this phase of the recommendation process.

Definition 2 [14] *Giving a fuzzy set* $b' = (\alpha_1, \ldots, \alpha_g)$ *defined on* $S = \{s_h\}$ *for* $h = 0, \ldots, g$, *we obtain its central value cv in the following way:*

$$cv = \frac{\sum_{h=0}^{g} idx(s_h)\,\alpha_h}{\sum_{h=0}^{g} \alpha_h}, \text{ where } idx(s_h) = h$$

This value represents the central position or centre of gravity of the information contained in the fuzzy set b'. *The range of this central value is the closed interval* $[0, g]$

Therefore, from this definition, it is defined the following similarity function *sim*:

Definition 3 [14] *Let* b'_1 *and* b'_2 *be two fuzzy sets defined on the BLTS,* $S_T = \{s_0, \ldots, s_g\}$, *and let* cv_1 *and* cv_2 *be the central values of* b'_1 *and* b'_2 *respectively, then the similarity between them is calculated as:*

$$sim\left(b'_1, b'_2\right) = 1 - \left|\frac{cv_1 - cv_2}{g}\right|$$

The final result of this step is a similarity vector $D = (d_1, \ldots, d_n)$ in which the system will keep the similarity between the user profile P_e and all the items in the database.

3.2.3. *Recommendation*

Here, the system will rank the items according to the similarity values of the vector $D = (d_1, \ldots, d_n)$. The best ones will be those that are closer the user profile, i.e., those with the greatest score in the similarity. The system will recommend the top-N items that reach a given threshold [7], i.e., if one of the top-N items is too far from the user profile (its similarity degree is less than the threshold) then this item will not be included in the recommendation.

Let the item set $A = \{a_1, \ldots, a_n\}$, r the maximum number of items to be recommended and h the threshold to be reached. Then, the recommendation to the user is given by the recommendation vector, R_A, where the first element is the top one recommended item, the second is the second closest to the user profile and so on:

$$R_A = \left(a_{q(1)}, \ldots, a_{q(r_1)}\right) \text{ where } r_1 \leq r$$

where the function q is defined in the following way:

$$\begin{array}{l} q : \{1, 2, \ldots, r_1\} \rightarrow \{1, 2, \ldots, n\} \\ q(i) \neq q(j)\ \forall i \neq j \\ q(i) \neq e\ \forall i = 1, \ldots, r_1 \\ \text{being } a_e \text{ the example given by the user} \\ d_{q(i)} \geq d_{q(j)}\ \ \forall i < j \\ d_{q(i)} \geq h\ \ \forall i = 1, \ldots, r_1 \end{array}$$

4. Example

We shall show an easy example in order to clarify our proposal. Let us suppose we are looking for a new car. The system has a database with the cars that could be recommended. They are described using a set of features. To simplify, in this example we have six items, $A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$ where each item corresponds to a car which is described by a set of four features $C = \{c_1, c_2, c_3, c_4\}$, where $c_1 = Price$, $c_2 = Velocity$, $c_3 = Safety$ and $c_4 = Comfort$. In real systems we could find thousand or millions items stored in the database and dozens of attributes are used to describe them.

To describe the attributes we could use different domains, i.e. different linguistic term sets. Here we have used the linguistic term set $S_{1,2}$ (see figure 5) for the attributes c_1 and c_2, and the linguistic term set $S_{3,4}$ (see figure 6) for c_3 and c_4 . These sets are defined by the following membership functions:

$$\begin{array}{l} s_0^{1,2} = Extremely\,low = (0, 0, .125) \\ s_1^{1,2} = Very\,low = (0, .125, .25) \\ s_2^{1,2} = Low = (.125, .25, .375) \\ s_3^{1,2} = A\,bit\,low = (.25, .375, .5) \\ s_4^{1,2} = Average = (.375, .5, .625) \\ s_5^{1,2} = A\,bit\,high = (.5, .625, .75) \\ s_6^{1,2} = High = (.625, .75, .875) \\ s_7^{1,2} = Very\,high = (.75, .875, 1) \\ s_8^{1,2} = Extremely\,high = (0.875, 1, 1) \end{array}$$

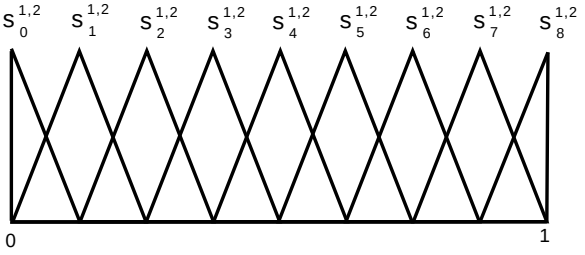


Fig. 5. The linguistic term set $S_{1,2}$

$$
\begin{aligned}
s_0^{3,4} &= Negligible = (0,0,.16)\\
s_1^{3,4} &= Very\,inferior = (0,.16,.33)\\
s_2^{3,4} &= Inferior = (.16,.33,.5)\\
s_3^{3,4} &= Average = (.33,.5,.66)\\
s_4^{3,4} &= Superior = (.5,.66,.83)\\
s_5^{3,4} &= Very\,superior = (.66,.83,1)\\
s_6^{3,4} &= Outstanding = (.83,1,1)
\end{aligned}
$$

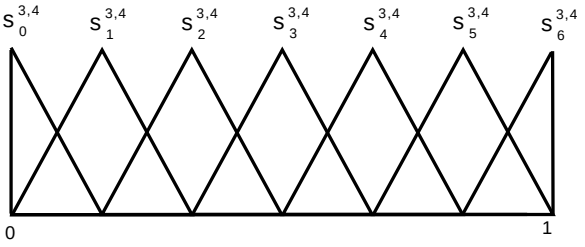


Fig. 6. The linguistic term set $S_{3,4}$

The descriptions of the items, using these linguistic term sets, can be seen in the table 1.

Table 1. Item database

	c_1	c_2	c_3	c_4
a_1	$s_0^{1,2}$	$s_3^{1,2}$	$s_3^{3,4}$	$s_2^{3,4}$
a_2	$s_5^{1,2}$	$s_2^{1,2}$	$s_1^{3,4}$	$s_4^{3,4}$
a_3	$s_7^{1,2}$	$s_4^{1,2}$	$s_0^{3,4}$	$s_5^{3,4}$
a_4	$s_5^{1,2}$	$s_6^{1,2}$	$s_2^{3,4}$	$s_6^{3,4}$
a_5	$s_8^{1,2}$	$s_0^{1,2}$	$s_4^{3,4}$	$s_6^{3,4}$
a_6	$s_1^{1,2}$	$s_8^{1,2}$	$s_3^{3,4}$	$s_1^{3,4}$

A user, u_e, wants to receive a recommendation from our system. Firstly, the system must build the user profile:

1 *Gathering the preferred example from the user:* the user states an item a_1, close to what he/she needs. So, this item is the example selected by the user and the initial user profile will be:

$$P_{e_0} = \left\{s_0^{1,2}, s_3^{1,2}, s_3^{3,4}, s_2^{3,4}\right\}$$

2 *Casual modification of preferences:* Nevertheless, the user realizes that the attribute c_1 (cars price) of the selected item does not represent what he/she wants. Due to this fact, he/she wants to provide a new value, however, the linguistic term set used to describe c_1 is not suitable for him/her knowledge. He/she doesn't need to discriminate between nine linguistic terms to express his/her necessity about this feature. So he/she would rather use a smaller set. He/she decides to use a linguistic term set with only five elements, S_1^{1e} that is defined below (see figure 7:)

$$
\begin{aligned}
s_0^{1e} &= Very\,low = (0,0,.25,)\\
s_1^{1e} &= Low = (0,.25,.5)\\
s_2^{1e} &= Average = (.25,.5,.75)\\
s_3^{1e} &= High = (0.5,0.75,1)\\
s_4^{1e} &= Very\,high = (.75,1,1)
\end{aligned}
$$

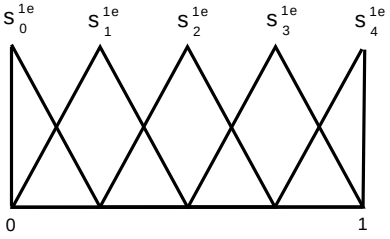


Fig. 7. The linguistic term set S_{1e}

The user assesses this attribute with the value s_1^{1e} and so, now, the user profile is:

$$P_e = \left\{s_1^{1e}, s_3^{1,2}, s_3^{3,4}, s_2^{3,4}\right\}$$

Once the the user profile is available, the system accomplishes the recommendation process:

1 *Unification of the linguistic information:* According to [11] the system chooses $S_{1,2}$ as BLTS (S_T), being its higher granularity the main criterion to make such decision, and transforms the user profile and item database into S_T. So, the item database (see Table 2) and the user

Table 2: Item database expressed into the BLTS

	c_1	c_2	c_3	c_4
a_1	$(1,0,0,0,0,0,0,0,0)$	$(0,0,0,1,0,0,0,0,0)$	$(0,0,.14,.57,1,.57,.14,0,0)$	$(.28,.71,.85,.42,0,0,0,0)$
a_2	$(0,0,0,0,0,1,0,0,0)$	$(0,0,1,0,0,0,0,0,0)$	$(.42,.85,.71,.28,0,0,0,0,0)$	$(0,0,0,0,.42,.85,.71,.28,0)$
a_3	$(0,0,0,0,0,0,0,1,0)$	$(0,0,0,0,1,0,0,0,0)$	$(1,.57,.14,0,0,0,0,0,0)$	$(0,0,0,0,0,.28,.71,.85,.42)$
a_4	$(0,0,0,0,0,1,0,0,0)$	$(0,0,0,0,0,0,1,0,0)$	$(0,.28,.71,.85,.42,0,0,0,0)$	$(0,0,0,0,0,0,.14,.57,1)$
a_5	$(0,0,0,0,0,0,0,0,1)$	$(1,0,0,0,0,0,0,0,0)$	$(0,0,0,0,.42,.85,.71,.28,0)$	$(0,0,0,0,0,0,.14,.57,1)$
a_6	$(0,1,0,0,0,0,0,0,0)$	$(0,0,0,0,0,0,0,0,1)$	$(0,0,.14,.57,.1,.57,.14,0,0)$	$(.42,.85,.71,.28,0,0,0,0,0)$

profile will be expressed in terms that can be compared one each other.

$$\begin{aligned} P_e = \quad & \{(0,0,.33,.66,1,.66,.33,0,0), \\ & (0,0,0,1,0,0,0,0,0) \\ & (0,0,.14,.57,1,.57,.14,0,0) \\ & (0,.28,.71,.85,.42,0,0,0,0)\} \end{aligned}$$

2 *Calculation of the similarity between the user profile and the items:* In order to obtain this measurement the system calculates the central values of the fuzzy sets of every item in the database (see table 3) and the fuzzy sets of the user profile:

$$P_e^{CV} = \{4,3,4,2.62\}$$

Table 3. Central values of the item database

	c_1	c_2	c_3	c_4
a_1	0	3	4	2.62
a_2	5	2	1.37	5.37
a_3	7	4	1.19	7.05
a_4	5	6	2.62	7.5
a_5	8	0	5.37	7.5
a_2	1	8	4	1.37

Finally, the system computes the similarity between the user profile and each item of the the database using the similarity function presented in 3 (see table 4) considering that all the features have the same importance.

Table 4. Similarity between the user profile and the items

a_1	a_2	a_3	a_4	a_5	a_6
0.87	0.77	0.64	0.67	0.58	0.71

3 *Providing the recommendations:* This is the final step of the recommendation process. The system will sort out the items according to the similarity to the user profile and obtains

$$R_A = (a_1, a_2, a_6, a_4, a_3, a_5)$$

The first item, a_1, cannot be recommended since it was chosen as an example of the user's necessities. Let's suppose that the system recommends the two items closest to the user profile, therefore the final recommendations will be:

$$\{a_2, a_6\}$$

5. Conclusions

Recommender Systems support users to find the most suitable products in e-shops within a huge amount of products according to their necessities and preferences. There exists different types of Recommendations Systems, as the Content-based or the the Collaborative Recommendation Systems, that provide good recommendations but they present some problems, overall, related to the amount of information necessary to make recommendations. Hybrid and Knowledge-based recommender systems face these problems from different points of view.

In this contribution we have presented a Knowledge Based Recommender System that deals with multigranular linguistic information instead of numerical values. The advantage of this representation is that we are able to gather the user's information, that is usually related to perceptions or tastes, without loosing expressiveness or accuracy. Moreover, we have defined a flexible model to deal with the information in which each attribute can be assessed with the most suitable linguistic term set and the users can use linguistic term sets according to their knowledge or preferences.

Besides, we have proposed the use of a similarity measure based on central values of fuzzy sets which is able to compute the similarity between items. In this manner, we haven't just taken into account if the assessments are the same, but also we have computed how different are they from each other.

6. Acknowledgements

This work is partially supported by the Research Projects TIN-2006-02121, 2007/BA020 and FEDER funds

References

1. C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and contentbased information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720, 1998.
2. P.P. Bonissone and K.S. Decker. *Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-Off Precision and Complexity*, chapter Uncertainty in Artificial Intelligence, pages 217–247. North-Holland, 1986.
3. G. Bordogna and G. Passi. A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation. *Journal of the American Society for Information Science*, (44):70–82, 1993.
4. R. Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69(32), 2000.
5. R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
6. R. D. Burke, K. J. Hammond, and B. C. Young. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997.
7. Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.
8. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61 – 70, 1992.
9. Robert H. Guttman. Merchant differentation through integrative negotiation in agent-mediated electronic comerce. *Master's Thesis, School of Architecture and Planning, Program in Media Arts and Sciences, Massachusetts Institute of Technology*, 1998.
10. C. Hayes and P. Cunningham. Smart radio - community based music radio. *Knowledge-Based Systems*, 14(3-4):197–201, 2001.
11. F. Herrera and E. Herrera-Viedma. Linguistic decision analisys: Steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, (115):67–82, 2000.
12. F. Herrera, E. Herrera-Viedma, and L. Martínez. A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems*, (114):43–58, 2000.
13. E. Herrera-Viedma, L. Martínez, F. Mata, and F Chiclana. A consensus support system model for group decision-making problems with multi-granular linguistic preference relations. *IEEE Transactions on Fuzzy Systems*, 13(5):644–658, 2005.
14. E. Herrera-Viedma, F. Mata, L. Martínez, F Chiclana, and L.G. Pérez. Measurements of consensus in multigranular linguistic group decision making. *Modeling Decisions for Artificial Intelligence, Proceedings Lecture Notes in Computer Science*, (3131):194–204, 2004.
15. J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, 1993.
16. B. Krulwich. Lifestyle finder: intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.
17. E. Levrat, A. Voisin, S. Bombardier, and J. Bremont. Subjective evaluation of car seat comfort with fuzzy set techniques. *International Journal of Intelligent Systems*, (12):891–913, 1997.
18. L. Martínez, J. Liu, J.B. Yang, and F. Herrera. A multigranular hierarchical linguistic model for design evaluation based on safety and cost analysis. *International Journal of Intelligent Systems*, 20(12):1161–1194, 2005.
19. G. Miller. The magical number sever or minus two: Some limits on our capacity of processing information. *Psychological Review*, (63):81–97, 1956.
20. M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61, 1996.
21. Saverio Perugini, Marcos Andre Goncalves, and Edward A. Fox. Recommender system research: A connection-centric survey. *Journal of Intelligent Information Systems*, 23(2):107–143, 2004.
22. P. Resnick and H.R. Varian. Recommender systems. *Association for Computing Machinery. Communications of the ACM.*, 40(3):56, Mar 1997.
23. J.B. Schafer, J.A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, (5):115–153, 2001.
24. L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-i, ii, iii. *Information Sciences*, 8-9:199–249, 301–357, 43–80, 1975.
25. L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.

International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems

IMPROVING THE EFFECTIVENESS OF KNOWLEDGE BASED RECOMMENDER SYSTEMS USING INCOMPLETE LINGUISTIC PREFERENCE RELATIONS*

Luis Martínez, Luis G. Pérez, Manuel Barranco, Macarena Espinilla

Department of Computer Sciences, University of Jaén, Campus Las Lagunillas s/n
Jaén, 23071, Spain
martin,lgonzaga,barranco,mestevez@ujaen.es

Received (received date)
Revised (revised date)

In the e-commerce arena new methods and tools have been recently developed to improve and customize the e-commerce web sites, according to users' necessities and preferences, that are usually vague and uncertain. The most successful tool in this field has been the Recommender Systems. Their aim is to assist e-shops customers to find out the most suitable products by using recommendations. Sometimes, these systems face situations where there is a lack of information or the information is vague or imprecise that yield unsuccessful results. Although several solutions have been proposed, they still present some limitations. In this paper, we present a Knowledge-Based Recommender System that manages and models the uncertainty related to users' preferences by using linguistic information. This system will overcome the problem of lack of information by computing recommendations through completing incomplete linguistic preference relations provided by the users.

Keywords: Recommender Systems, Fuzzy Linguistic Approach, incomplete preference relations, linguistic 2-tuples, e-commerce, e-services

1. Introduction

Recommender Systems[2,5,6,17,26,31,32,33] have been one of the key issues in the development and success of e-commerce[34]. Customers usually face websites with a huge range of items that can potentially satisfy their requirements. However, only a small set of these will fulfil their preferences and/or necessities. Unfortunately, these are often difficult to establish. It may seem a typical decision making problem, where the users must choose the best alternative(s) amongst the offered ones, but it is not. This is due to the fact that users are unable to explore the entire range of alternatives in the e-shop . For this reason, they usually choose from the first subset of items that could partially fulfil their needs. Notwithstanding the fact that they

*This work is partially supported by the Research Projects TIN-2006-02121, JA031/06 and FEDER funds

are aware these items are probably not the optimal ones. Recommender Systems can assist customers with their *shopping searches* by leading them to interesting items by means of recommendations.

All the Recommender Systems have essentially the same aim, to lead users through recommendations to those items that are the most suitable for them. However, the techniques used to achieve this aim are different from each other, both in the process of gathering information and in the process of computing the recommendations. According to these techniques, Recommender Systems are classified as:

- *Demographic Recommender Systems*[26]*:* these systems categorize their users into demographic groups, and make recommendations to a specific user according to, the information about the people that belong to the same demographic group.
- *Content-based Recommender Systems*[30,31,33]*:* A Content-Based Recommender System learns a user profile based on the features of the items that the user has liked and, it uses this profile to find out similar items that the user could like.
- *Collaborative Filtering Recommender Systems* [17]*:* they use users' ratings to filter and recommend items to a specific user. In the simplest case, these systems predict the users' preferences as a weighted aggregation of the other users' preferences.
- *Knowledge Based Recommender Systems*[5,32]*:* these systems use the knowledge about users' necessities and how an item matches these necessities to infer recommendations about which items fulfil the user's expectations.
- *Utility Based Recommender Systems*[18]*:* they make recommendations by computing a utility value for each object.
- *Hybrid Recommender Systems*[2,6]*:* these systems arose with the aim of addressing several drawbacks presented in the previous ones. To accomplish this aim, they combine different techniques to improve the accuracy of the recommendations.

To choose the most suitable items for a user, all Recommender Systems gather information about the items, the users and their necessities. The information related to users' necessities is usually vague and uncertain. However, most of Recommender Systems use numerical and precise assessments in the gathering process. The use of Linguistic Information[46] has provided successful results modelling uncertain information in different areas such as Information Retrieval[4], Clinical Diagnosis[12], Marketing[41], Risk in Software Development[29], Technology Transfer Strategy Selection[9], Education[27,28], Decision Making[10,13,21,23,38,40], Consensus[3,25], Recommender Sytems[30,44], etc. For this reason, the use of the Fuzzy Linguistic Approach to model uncertainty in Recommender Systems should be further considered.

On the other hand, another important problem of the Recommender Systems is that information related to the users and items might be scarce and insufficient. In such cases, classical Recommender Systems (the collaborative and content-based ones) are unable to make good recommendations. To overcome this drawback, hybrid and knowledge based systems have been used. In this paper we will focus our

interest in Knowledge Based Recommender Systems[5], that usually require the users to provide one example of their preferences. A user profile is then inferred from that example and utilized to find out the most similar items that are returned as recommendations. Sometimes, the information of the user profile inferred from the example might not match the user's preferences. Therefore, the users could change some features of the given example. This task may be tedious and time-consuming, and not all the users might be willing or trained to accomplish this refinement.

The aim of this paper is to provide a linguistic Knowledge Based Recommender System that improves the managing of the two problems previously mentioned: (i) the modelling of vague information and (ii) the lack of information. In order to do this, we propose a Recommender System that deals with a linguistic framework to model the user's preferences. Regarding the latter problem, our main concern would be to increase the knowledge about the user by obtaining an accurate and useful user profile, whilst decreasing the time he/she spends to provide it. Hence, to accomplish this second goal, the system will require the user a preference relation in order to gather more information. Initially the use of a preference relation could result in more time being spent and further inconsistencies. To avoid such problems and achieve our goal we propose that the system, will require the user to provide an incomplete linguistic preference relation that will be completed by using an algorithm based on the consistency property. In such a way, the information gathering process will increase the knowledge about the users and also decrease its time cost.

This contribution is structured as follows, in section 2 we shall review some necessary preliminaries to understand our proposal. In section 3 we shall present our Linguistic Knowledge Based Recommender System while in section 4 we shall demonstrate the application of our model. Finally in section 5 some conclusions are pointed out.

2. Preliminaries

This section analyzes the problems that motivates our proposal, namely the lack of information and the modelling of uncertainty in Recommender Systems. First, we review how does lack of information affect Recommender Systems? Afterwards some concepts about linguistic preference modelling, such as, the Fuzzy Linguistic Approach, the 2-tuple Linguistic Representation and linguistic preference relations are also reviewed. Finally, it is revised the use of the consistency property to complete an incomplete preference relation.

2.1. *Lack of Information in Recommender Systems*

The Collaborative Filtering[17] and the Content-based Recommender Systems[33] are the most used and well-known types of Recommender Systems, but they are not always suitable. Sometimes, they present important drawbacks. For example, the Collaborative Filtering Recommender Systems need a huge database of users' ratings to filter and recommend items to a specific one. In the simplest case, these

systems predict the users' preferences as a weighted aggregation of the other users' preferences, in which the weights are proportional to the similarity among users based on their ratings. On the other hand, Content-based Recommender Systems look for new items that are similar to those ones that the user has bought in the past. Therefore, both systems require that the user has assessed a minimum number of items to suggest good recommendations.

However, in the real world there are situations where the previous models are not suitable because of lack of information. Several common problems in these models, related to the lack of information were presented in Ref. 6:

- *The new user ramp-up problem:* it means inaccurate recommendations for users with few ratings. This problem is common in both Content-based Recommender Systems and Collaborative Filtering.
- *New item ramp-up problem:* in Collaborative Filtering Recommender Systems an item with few ratings, it is not easy to be recommended.
- *Grey sheep problem:* in Collaborative Filtering Recommender Systems could exist users whose ratings are not consistently similar with any group of users. Then they will not receive good recommendations.
- *Quality dependent on large historical data set.*

To address these problems some alternatives have been presented, such as the Hybrid Recommender Systems[2,6] and the Knowledge Based Recommender Systems[5]. The former combines different techniques, usually the collaborative and content-based algorithms to smooth out the disadvantages of both types of Recommender Systems. For instance, they do not suffer from new item ramp-up problem but they still suffer the new user ramp-up problem.

The latter does not suffer from the previous problems but they need a knowledge acquisition process[6], in which the users provide one example that reflects their preferences. The users do not want an item exactly equal to the example. The user then should modify several features of an initial profile inferred directly from the example. Finally, a user profile is obtained and utilized to find out the user's preferred items from a database. There are several methods to exploit this knowledge [7,8], for example Entree uses case based reasoning[19] to make recommendations. It's easy to observe that new users can obtain recommendations with these systems. But if the information gathering process is time-consuming, the users could not be willing to provide their preferences.

2.2. *Fuzzy Linguistic Approach*

It is remarkable that most of the Recommender Systems manage the subjective information related to the users' preferences or necessities by using crisp values [2,5,17,26]. In this paper, we shall propose to model this subjective information by means of linguistic information in order to offer a more flexible and suitable recommendation framework to the users.

Usually, we work in a quantitative setting, where the information is expressed by means of numerical values. However, many aspects of different activities in the real world involve vague or imprecise knowledge and cannot be easily assessed in a quantitative form, but rather in a qualitative one. In such a case a better approach may be to use linguistic assessments instead of numerical values. The fuzzy linguistic approach represents qualitative aspects as linguistic values by means of linguistic variables[46].

To deal with the linguistic information we have to choose the appropriate linguistic descriptors for the term set and their semantics. In the literature, several possibilities can be found (see Ref. 20 for a wide description). One possibility of generating the linguistic term set consists of directly supplying the term set by considering all terms distributed on a scale on which a total order is defined [42]. For example, a set of seven terms S, could be given as follows:

$$S = \{s_0 : N, s_1 : VL, s_2 : L, s_3 : M, s_4 : H, s_5 : VH, s_6 : P\}$$

Usually, in these cases, it is required that in the linguistic term set there exist:

1) A negation operator: $\text{Neg}(s_i) = s_j$ such that $j = g$-i (g+1 is the cardinality).

2) An order: $s_i \leq s_j \Longleftrightarrow i \leq j$. Therefore, there exists a min and a max operator.

The semantics of the terms are given by fuzzy numbers defined in the [0,1] interval, which are usually described by membership functions[14,46].

The use of linguistic information implies processes of computing with words (CW), in the literature can be found different computational models:

- *The approximate computational model based on the Extension Principle*[12]. This model uses fuzzy arithmetic based on the Extension Principle[15] to make computations over the linguistic variables.
- *The symbolic linguistic computational model*[14]. This symbolic model makes direct computations on labels, using the ordinal structure of the linguistic term sets.
- *The 2-tuple linguistic computation model*[22]. It uses the 2-tuple fuzzy linguistic representation model.

2.3. *The 2-Tuple Linguistic Representation Model*

Here we review the linguistic 2-tuple representation model[22] that we shall use in our proposal to carry out processes of CW in a precise way. This model is based on the concept of symbolic translation.

The 2-tuple fuzzy linguistic representation model represents the linguistic information by means of a 2-tuple, (s, α), where s is a linguistic label and α is a numerical value that represents the value of the symbolic translation.

Definition 1. The Symbolic Translation of a linguistic term $s_i \in S = \{s_0, ..., s_g\}$ is a numerical value assessed in $[-.5, .5)$ that supports the "difference of information" between an amount of information $\beta \in [0, g]$ and the closest value in $\{0, ..., g\}$ that

indicates the index of the closest linguistic term $s_i \in S$. Being [0,g] the interval of granularity of S.

This linguistic representation model defines a set of functions to make transformations between linguistic 2-tuples and numerical values:

Definition 2. Let $S = \{s_0, ..., s_g\}$ be a linguistic term set and $\beta \in [0, g]$ a value that represents the result of a symbolic aggregation operation. The 2-tuple that expresses the equivalent information to β is obtained as:

$$\Delta : [0, g] \longrightarrow S \times [-0.5, 0.5)$$
$$\Delta(\beta) = (s_i, \alpha), \ with \ \begin{cases} s_i & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-.5, .5) \end{cases}$$

where $round(\cdot)$ is the usual *round* operation, s_i has the closest index label to "β" and "α" is the value of the symbolic translation.

Proposition 1. *Let $S = \{s_0, ..., s_g\}$ be a linguistic term set and (s_i, α) be a linguistic 2-tuple. There is always a Δ^{-1} function, such that, from a 2-tuple it returns its equivalent numerical value $\beta \in [0, g]$ in the interval of granularity of S.*

Proof. It is trivial, we consider the function:

$$\Delta^{-1} : S \times [-.5, .5) \longrightarrow [0, g]$$
$$\Delta^{-1}(s_i, \alpha) = i + \alpha = \beta$$

□

Remark 1. From definitions 1, 2 and proposition 1, it is obvious that the conversion of a linguistic term into a linguistic 2-tuple is:

$$s_i \in S \Longrightarrow (s_i, 0)$$

This linguistic representation model has associated a computational model defined in Ref. 22 that defines the following basic operations:

(1) *Comparison of 2-tuples:* it is accomplished according to an ordinary lexicographic order. Let (s_i, α_i) and (s_j, α_j) be two 2-tuples:

   (a) if $i < j$ then (s_i, α_i) is smaller than (s_j, α_j)
   (b) if $i = j$ then
      i. if $\alpha_i = \alpha_j$ then (s_i, α_i) and (s_j, α_j) represents the same information
      ii. if $\alpha_i < \alpha_j$ then (s_i, α_i) is smaller than (s_j, α_j)
      iii. if $\alpha_i > \alpha_j$ then (s_i, α_i) is bigger than (s_j, α_j)

(2) *Aggregation of 2-tuples:* the aggregation of linguistic 2-tuples consists of obtaining a value that summarizes a set of values. In Ref. 22 we can find several 2-tuples aggregation operators.

(3) *Negation operator of a 2-tuple:* this operator is defined as:

$$Neg\,(s_i, \alpha) = \Delta\left(g - \Delta^{-1}\,(s_i, \alpha)\right)$$

where $g + 1$ is the cardinality of S.

2.4. *Preference relations*

Once we had decided the modelling of the subjective information in our model, we had to decide which structure of representation would be the more adequate to our aims. There exist different structures to represent preferences about a set of items $X = \{x_1, ..., x_n\}$. The most common ones are:

(1) *A preference ordering of items*[11]*:* in which a user provides his/her preferences about X as an individual preference ordering $O^k = \left\{o^k(1), ..., o^k(n)\right\}$, where $o^k(\cdot)$ is a permutation function over the index set $\{1, ..., n\}$ in which the items are ordered from best to worst.
(2) *A utility vector*[37]*:* the preferences are provided by using a set of n utility values $U^k = \left\{u_i^k, i = 1, ..., n\right\}$, in which, u_i^k, represents the utility assessment for the item x_i. The bigger the evaluation the more the preference.
(3) *A preference relation*[35]*:* in this representation the information is described by a preference matrix $P \subseteq X \times X$, $P = (p_{ij})$, where p_{ij} indicates the preference intensity for the item x_i regarding the item x_j.

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \dots & \ddots & \dots \\ p_{n1} & \cdots & p_{nn} \end{pmatrix}$$

In our proposal we shall deal with preference relations because the user can provide more detailed information about the his/her preferences. Initially this representation could be time-consuming and lead to inconsistences. But to overcome these drawbacks we will take advantage that in the literature has been studied the use of preference relations where the users are under time pressure, either there is a lack of information or some alternatives could be unknown [1,24,39]. In these cases some preferences, p_{ij}, could be missed. Such relations are called *incomplete preference relations*, that we shall use in our proposal in order to improve the time cost and consistency of the information gathering process.

Here, we review some definitions about preference relations that are used in our proposal of Knowledge Based Recommender System dealing with incomplete preference relations.

Definition 3. A reciprocal numerical relation[11,16,36] about a set of alternatives $X = \{x_1, ..., x_n\}$ is a function regarding the alternatives set $X \times X$:

$$\mu_p : X \times X \longrightarrow [0, 1],$$

where every value meets the conditions that $p_{ij} = 1 - p_{ji}\ \forall i, j \in \{1, ..., n\}$ and represents the preference degree or intensity of alternative, x_i regarding, x_j:

- $p_{ij} = 1/2$ indicates indifference between x_i and x_j. ($x_i \sim x_j$)
- $p_{ij} = 1$ indicates that x_i is absolutely preferred to x_j
- $p_{ij} > 1/2$ indicates that x_i is preferred to x_j ($x_i \succ x_j$)

Definition 4. A reciprocal numerical relation[11,16,36] about a set of alternatives $X = \{x_1, ..., x_n\}$ is additive consistent if and only if:

$$p_{ij} + p_{jk} + p_{ki} = \frac{3}{2} \forall i \leq j \leq k$$

Definition 5. [1] A function $f : X \rightarrow Y$ is partial if every element in the set X not necessarily maps onto an element in the set Y. However, if every element from the set X maps onto one element of the set Y then f is a total function.

Definition 6. [1] A preference relation P about a set of alternatives X with a partial function is an incomplete preference relation.

We have pointed out that our proposal will model the information provided by the users by means of linguistic information by using the linguistic 2-tuple representation model. We shall then deal with linguistic preference relations as:

Definition 7. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of alternatives and S a linguistic term set, the preference attitude about X can be defined as a linguistic preference relation, $P = (p_{ij}), i, j = 1, \ldots, n$, based on the 2-tuple linguistic model as:

$$\mu_P : X \times X \longrightarrow S \times [-0.5, 0.5),$$

where $\mu_P(x_i, x_j) = p_{ij} \in S \times [-0.5, 0.5)$ is a 2-tuple which denotes the preference degree of alternative x_i regarding x_j.

If there is not a 2-tuple for every pair of alternatives, we have an incomplete linguistic preference relation.

2.5. *Dealing with incomplete linguistic preference relations*

We mentioned that the use of preference relations can be time consuming and contain inconsistencies from the user. We then propose the use of incomplete preference relations to improve the efficiency of the information gathering process as well as facilitates the avoidance of inconsistences. To do so, we shall complete such incomplete preference relations by means of a completion algorithm based on the consistency property. There exist different algorithms [1,24,39] to complete an incomplete preference relation. The method suggested by Alonso et al. in Ref. 1 provides several interesting mathematical and consistency properties in order to complete an incomplete linguistic preference relation. Due to this fact, it will be used in our proposal and reviewed in further detail.

Let P an incomplete linguistic preference relation, $P = (p_{ij})$, whose assessments belong to a linguistic term set $S = \{s_0, ..., s_g\}$. The preference relation must fulfil the condition "$\forall i \; \exists j \; i \neq j \mid p_{ij}$ is known or p_{ji} is known". P, can be then completed with the algorithm proposed in Ref. 1 as follows:

1. *Initializations*
   $P' = \Delta^{-1}(P)$
   $EMV_0 = \emptyset$
   $h = 1$
2.1. *while* $EMV_h \neq \emptyset$ {
2.2. *for every* $(i,k) \in EMV_h$ {
2.3. $\mathcal{K} = 0$
2.4. $H_{ik}^1 = \{j \neq i,k | (i,j),(j,k) \in KV_h\}$; *if* $\left(H_{ik}^1 \neq \emptyset\right)$ *then* $\mathcal{K} = \mathcal{K} \cup \{1\}$
2.5. $H_{ik}^2 = \{j \neq i,k | (j,k),(j,i) \in KV_h\}$; *if* $\left(H_{ik}^2 \neq \emptyset\right)$ *then* $\mathcal{K} = \mathcal{K} \cup \{2\}$
2.6. $H_{ik}^3 = \{j \neq i,k | (i,j),(k,j) \in KV_h\}$; *if* $\left(H_{ik}^3 \neq \emptyset\right)$ *then* $\mathcal{K} = \mathcal{K} \cup \{3\}$
2.7. $cp_{ik}^{'} = \frac{1}{\#\mathcal{K}} \left(\sum_{l \in \mathcal{K}} \frac{\sum_{j \in H_{ik}^l} cp_{ik}^{jl}}{\#H_{ik}^l} \right)$
2.8. *if* $cp_{ik}^{'} < 0$ *then* $cp_{ik}^{'} = 0$
elseif $cp_{ik}^{'} > g$ *then* $cp_{ik}^{'} = g$
2.9. $p'_{ik} = cp'_{ik}$
2.10. }
2.11. $h++$
}
3. $P'' = \Delta(P')$

where:

KV_h means *known values in iteration h*

UV_h means *unknown values in iteration h*

EMV_h is the subset of the missing values which can be calculated in step h

$EMV_h = \left\{(i,k) \in UV_h | \exists j \in H_{ik}^1 \cup H_{ik}^2 \cup H_{ik}^3\right\}$

$cp_{ik}^{j1} = p'_{ij} + p'_{jk} - \frac{g}{2}$; $cp_{ik}^{j2} = p'_{jk} - p'_{ji} + \frac{g}{2}$; $cp_{ik}^{j3} = p'_{ij} - p'_{kj} + \frac{g}{2}$

An example about how this algorithm can complete an incomplete preference relation is showed.

Example. Let $S = \{s_0 = VL, s_1 = L, s_2 = I, s_3 = H, s_4 = VH\}$ and $X = \{x_1, x_2, x_3, x_4\}$ be a linguistic term set and a set of alternatives respectively.

Step 1

Let's suppose a user provides the following incomplete linguistic preference relation, P:

$$P = \begin{pmatrix} - & (VL,0) & (L,0) & (H,0) \\ x & - & x & x \\ x & x & - & x \\ x & x & x & - \end{pmatrix}$$

The preference relation P is transformed into, P', by means of the function Δ^{-1}:

$$P' = \begin{pmatrix} - & 0 & 1 & 3 \\ x & - & x & x \\ x & x & - & x \\ x & x & x & - \end{pmatrix}$$

Step 2

Then the algorithm described by the steps 2.1 to 2.11 is applied to complete the preference relation P':

$$P' = \begin{pmatrix} - & 0.00 & 1.00 & 3.00 \\ 3.50 & - & 3.00 & 4.00 \\ 2.53 & 1.00 & - & 3.17 \\ 1.59 & 0.00 & 0.83 & - \end{pmatrix}$$

These values have been computed with the following calculations:

$cp'_{23} = p'_{13} - p'_{12} + 2 = 1 - 0 + 2 = 3.00$; $p'_{23} = 3.00$

$cp'_{32} = p'_{12} - p'_{13} + 2 = 0 - 1 + 2 = 1.00$; $p'_{32} = 1.00$

$cp'_{24} = p'_{14} - p'_{12} + 2 = 3 - 0 + 2 = 5.00$; $p'_{24} = 4.00$

$cp'_{42} = p'_{12} - p'_{14} + 2 = 0 - 3 + 2 = -1.00$; $p'_{42} = 0.00$

$cp'_{34} = \frac{1}{3}\left((p'_{32} + p'_{24} - 2) + \frac{1}{2}\left((p'_{14} - p'_{13} + 2) + (p'_{24} - p'_{23} + 2)\right) + (p'_{32} - p'_{42} + 2)\right) = \frac{1}{3}\left((1 + 4 - 2) + \frac{1}{2}\left((3 - 1 + 2) + (4 - 3 + 2)\right) + (1 - 0 + 2)\right) = 3.17$; $p'_{34} = 3.17$

$cp'_{43} = \frac{1}{3}\left((p'_{42} + p'_{23} - 2) + \frac{1}{2}\left((p'_{13} - p'_{14} + 2) + (p'_{23} - p'_{24} + 2)\right) + (p'_{42} - p'_{32} + 2)\right) = \frac{1}{3}\left((0 + 3 - 2) + \frac{1}{2}\left((1 - 3 + 2) + (3 - 4 + 2)\right) + (0 - 1 + 2)\right) = 0.83$; $p'_{43} = 0.83$

$cp'_{21} = \frac{1}{2}\left((p'_{23} - p'_{13} + 2) + (p'_{24} - p'_{14} + 2)\right) = \frac{1}{2}\left((3 - 1 + 2) + (4 - 3 + 2)\right) = 3.5$; $p'_{21} = 3.5$

$cp'_{31} = \frac{1}{3}\left((p'_{32} + p'_{21} - 2) + (p'_{21} - p'_{23} + 2) + \frac{1}{2}\left((p'_{32} - p'_{12} + 2) + (p'_{34} - p'_{14} + 2)\right)\right) = \frac{1}{3}\left((1 + 3.5 - 2) + (3.5 - 3 + 2) + \frac{1}{2}\left((1 - 0 + 2) + (3.17 - 3 + 2)\right)\right) = 2.53$; $p'_{31} = 2.53$

$cp'_{41} = \frac{1}{3}\left(\frac{1}{2}\left((p'_{42} + p'_{21} - 2) + (p'_{43} + p'_{31} - 2)\right) + \frac{1}{2}\left((p'_{21} - p'_{24} + 2) + (p'_{31} - p'_{34} + 2)\right) + \frac{1}{2}\left((p'_{42} - p'_{12} + 2) + (p'_{43} - p'_{13} + 2)\right)\right) = \frac{1}{3}\left(\frac{1}{2}\left((0 + 3.5 - 2) + (0.83 + 2.53 - 2)\right) + \frac{1}{2}\left((3.5 - 4 + 2) + (2.53 - 3.17 + 2)\right) + \frac{1}{2}\left((0 - 0 + 2) + (0.83 - 1 + 2)\right)\right) = 1.59$; $p'_{41} = 1.59$

Step 3

Finally, we transform the preference relation, P', into a 2-tuple linguistic preference relation, P'', by means of the function Δ:

$$P'' = \begin{pmatrix} - & (VL, 0) & (L, 0) & (H, 0) \\ (H, 0.5) & - & (H, 0) & (VH, 0) \\ (H, -0.47) & (L, 0) & - & (H, 0.17) \\ (I, -0.41) & (VL, 0) & (L, -0.17) & - \end{pmatrix}$$

3. A Knowledge Based Recommender System Based on Incomplete Preference Relations

In this section, we present our model for a Knowledge-Based Recommender System that deals with incomplete linguistic preference relations. The information required by this system to accomplish its processes is:

- *Users' necessities.* An incomplete linguistic preference relation is required on-line when the user demands a recommendation.
- *Information about all the items that can be recommended.* Each item is described by a vector of features where each feature will be assessed with a number that belongs to the interval $[0, 1]$. These descriptions are stored in a database.

This model will have a key advantage regarding the content-based[30,33] and the collaborative ones[17]. Due to the fact that these systems require historical information about the preferences of the user in the past and complex processes for acquiring user's preferences. However, our Knowledge Based Recommender System only needs a selection of four or five preferred items that the user must choose, and then provide an incomplete preference relation about them. With this information the system builds a complete user profile that will use to find out the most promising items for the user from a database that contains all the items that can be purchased.

Therefore in our proposal the system will deal with a database noted as, X. Let $X = \{x_1, x_2, ..., x_m\}$ be the set of items to be recommended and each one is described by a vector of features $x_i = \{c_i^1, \ldots, c_i^t\}$. Although the process of building this type of database could be automatic, it is usually required the assistance of an expert[6] to define, (i) which features are important, (ii)which values could be used, (iii) if there is any kind of relationship between different values, etc.

Once we have got the database, the proposed system will deal with the following three phases (see figure 1):

(i) *Acquiring the user preference information:* The aim of this phase is to gather the user's preference information. This phase is a two-step process:
   (a) *Setting the favourite examples:* The user will choose his/her favourite examples. He/she will then provide an incomplete linguistic preference relation providing just one row of the relation.
   (b) *Completing the preference relation*: By using the algorithm proposed in section 2.5 the incomplete linguistic preference relation is completed.

(ii) *Building the user profile:* The system infers a user profile by using the complete preference relation and, the descriptions of the items contained in the database. This phase has two steps:
   (a) *Building partial user profiles:* The system exploits the preference relation to obtain partial user profiles.

(b) *Obtaining the user profile:* from the previous profiles is computed the final one.

(iii) *Recommendation:* The system makes use of the user's profile to find out the items that best satisfy his/her necessities or preferences.

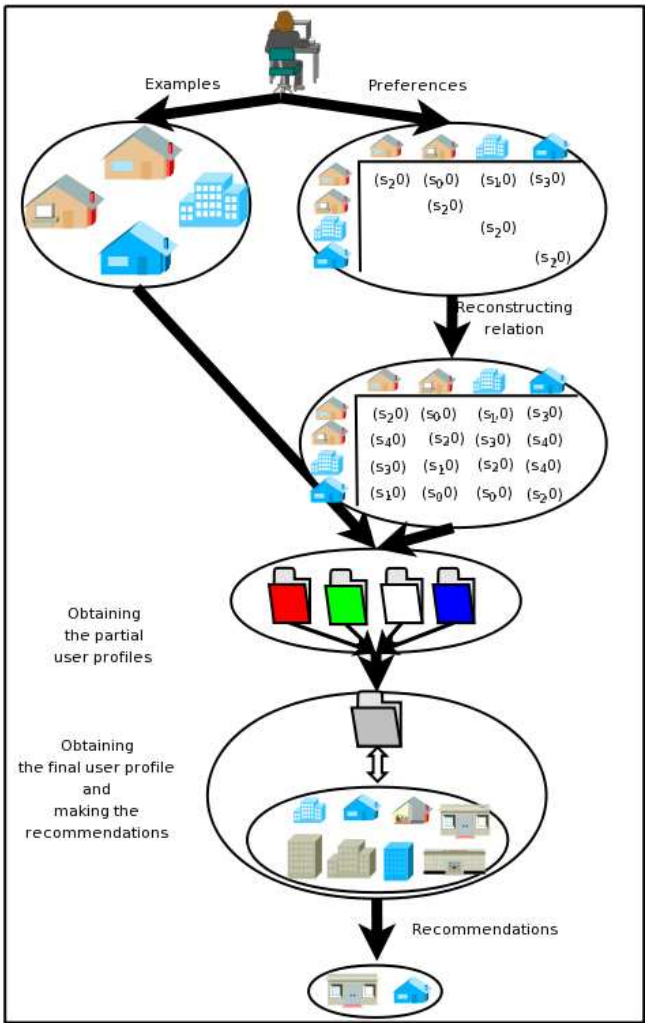


Fig. 1. Recommendation Model

We present in further detail these phases in the following subsections.

3.1. *Acquiring the user preference information*

The goal of this phase is to obtain information about the user's preferences. First, the user must choose some items (four or five) as examples of his/her preferences, tastes or necessities. Due to the fact that the database, X, can be huge to make easier this choice task, the system offers a subset $X_r = \{x_1^r, x_2^r, ..., x_{m'}^r\}$ $(m' \leq m)$. This subset should be big enough to have items that represent any kind of user's necessities, but not too big since users could find the task of choosing the example of his/her necessities too tedious and time-consuming. Moreover, these items ought to be "well-known" for almost everyone. There are several ways to obtain this subset, the easiest way is to use lists of items created by experts or the bestsellers, for instance, *CDNOW* (www.cdnow.com) offers their users lists of CDs by genre created by music experts or let them consult the bestsellers at the moment. It would be easy to define, X_r, from these lists.

Remark 2. There is not any correlation between "well-known" and "preferred", i.e., in this set, users could find interesting items as well as items that the user dislikes. For instance, in recommendation of hotels the *Hilton Hotels* are well known but it might be that the user does not like.

3.1.1. *Setting the favourite examples*

The set, X_r, is shown to the user, so he/she can choose a subset $X_u = \{x_1^u, ..., x_n^u\}$, with four or five elements, according to his/her needs.

Afterwards, the system inquires the users to express their preferences amongst the elements of X_u, by means of an incomplete linguistic preference relation where their preferences are assessed in a linguistic term set $S = \{s_0, ..., s_g\}$.

Due to the fact that one main goal of our proposal is decrease the time of the information gathering process. The system will require the user only one a row of the preference relation. It will be then completed by using the algorithm presented in section 2.5. So, the system provides three main benefits in order to gather the user's information:

(i) The task is easier and quicker for the user: he/she provides the minimum information necessary.
(ii) The proposed algorithm completes an incomplete preference relation with an only row of known values and avoids inconsistencies.
(iii) Since the system uses several examples, the recommendations are less dependent on the adequacy of the examples than in classical Knowledge Based Recommender System. On the one hand, in classical Knowledge Based Recommender Systems the recommendations are led by one example. If such an example is not adequate, the recommendations will be likely inaccurate. When recommendations are led by several examples, it will be more likely to obtain better recommendations whenever some of the examples are adequate.

3.1.2. *Completing the preference relation*

Once the user has provided only one row of the preference relation. The system then completes this preference relation by applying the algorithm showed in 2.5. The result is the following preference relation:

$$P'' = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p^*_{21} & p_{22} & \cdots & p^*_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p^*_{n1} & p^*_{n2} & \cdots & p_{nn} \end{pmatrix}$$

where p_{1j} is a known value provided by the user about the preference of x^u_1 regarding x^u_j, expressed in the linguistic term set S, and p^*_{ij} is a value estimated about the preference of x^u_i regarding x^u_j. By definition, p_{ii} is the indifference term of the term set S

3.2. *Building the user profile*

Now the system has a complete linguistic preference relation, the following phase will be to compute a user profile in order to compare the user's necessities with the items stored in the database. The system computes the user's profile by using the complete preference relation, P'', and the descriptions of the items in X_u, considered in such a preference relation. The user profile is computed in two steps:

3.2.1. *Building partial user profiles*

For each column of the preference relation, $(p_{1j}, p_{2j}, \ldots, p_{nj})$, the system obtains a partial user profile that represents the user's preferences related to the item $x^u_j \in X_u$ by aggregating the vectors of characteristics of the items , $x^u_i = \{c^{u1}_i, ..., c^{ut}_i\}, i \in \{1, ..., n\}$ *and* $i \neq j$. The aggregation operator used in our proposal is the IOWA (Induced OWA) operator proposed in Ref. 43.

The IOWA operator aggregates pairs of values, (v_i, a_i). Within these pairs, v_i is called the order inducing value and, a_i the argument value. The IOWA aggregation operator acts as:

$$F_W\left(\langle v_1, a_1\rangle, \ldots, \langle v_l, a_l\rangle\right) = W^T B_v$$

where $B_v = (b_1, \ldots, b_l)$ is the result of ordering the vector $A = (a_1, \ldots, a_l)$ according to the value of the order inducing variables, v_i, and W^T is the column vector of weights which satisfies the following conditions:

$$W = (w_1, \ldots, w_l)$$
$$w_i \in [0, 1] \quad \forall i \sum_{i=1}^{l} w_i = 1$$

Here is computed a partial profiles, $\left\{pp_j = \left(c^1_{pp_j}, \ldots, c^t_{pp_j}\right), j = \{1, ..., n\}\right\}$,for each item, x^u_j. The partial profile, pp_j, is a vector whose values indicate the user's preferences according to his/her necessities with regards to the item x^u_j. The partial profile, pp_j is obtained by aggregating the vectors $\left\{\left(c^1_i, ..., c^t_i\right), \forall i \neq j\right\}$ that describes the items $\{x^u_i, \forall i \neq j\}$. Each element, $c^k_{pp_j}$, is obtained by aggregating the $n-1$ elements $\left\{c^k_i, \forall i \neq j\right\}$ by means of the IOWA operator. Therefore, for the element $c^k_{pp_j}$, the order inducing variables are the values of the preference relation, p_{ij}, such that, $\{p_{ij}, \forall i \neq j\}$. Then:

$$c^k_{pp_j} = F_W\left(\left\langle p_{1j}, c^k_1\right\rangle, \ldots, \left\langle p_{nj}, c^k_n\right\rangle\right) = W^T B_v,$$

where the vector $B_v = (b_1, \ldots, b_{n-1})$ is given by a decreasing order of the elements of the set $\left\{c^k_i, \forall i \neq j\right\}$ according to the order induced by the values, $(p_{1j}, \ldots, p_{nj})$.

There are different methods to compute the weighting vectors $W = (w_1, \ldots, w_{n-1})$. Yager suggested an interesting way to compute the weighting vector for OWA operators using non-decreasing linguistic quantifiers[43]. A non-decreasing linguistic quantifier must satisfy three properties:

(i) $Q(1) = 1$
(ii) $Q(0) = 0$
(iii) $Q(r_1) \geq Q(r_2)\ if\ r1 \geq r2$

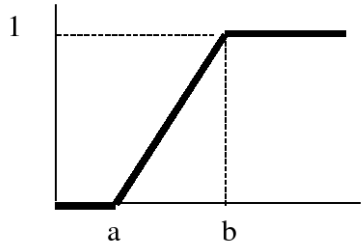


Fig. 2. Non-decreasing proportional quantifier

In the case of a non-decreasing proportional quantifier Q (see fig. 2), the weights for the element i, for $i = 1, \ldots, n-1$, are computed by the following expression:

$$w_i = Q\left(\frac{i}{n-1}\right) - Q\left(\frac{i-1}{n-1}\right) \tag{1}$$

where the membership function of a non-decreasing proportional quantifier Q is as follows:

$$Q(x) = \begin{cases} 0 & if\ x < a \\ \frac{x-a}{b-a} & if\ a \leq x \leq b \\ 1 & if\ x > b \end{cases}$$

3.2.2. *Obtaining the final user profile*

The system needs a user profile, $FP_u = (c^i_{fp}), i = 1 \ldots, t$, in order to compute the recommendations. To obtain such a profile, it combines the partial ones (see figure 3), $pp_1, \ldots, pp_n$, by using again the IOWA operator. The aggregation of the partial profiles, $pp_j = \left(c^1_{pp_j}, \ldots, c^t_{pp_j}\right)$, obtained for every item x^r_j is computed as:

$$c^k_{fp} = F'_W\left(\left\langle p_1, c^k_{pp_1}\right\rangle, \ldots, \left\langle p_n, c^k_{pp_n}\right\rangle\right) = W'^T B'_v,$$

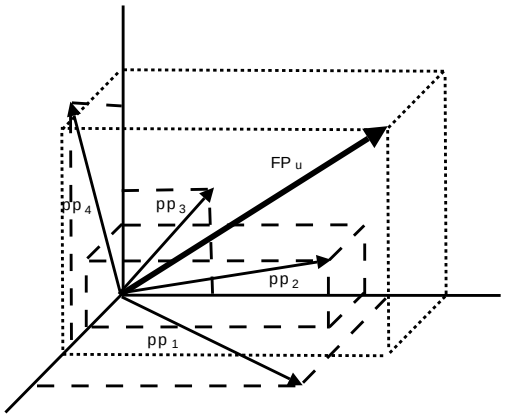


Fig. 3. Building the user profile

where the vector $B'_v = (b'_1, \ldots, b'_n)$ is given by a decreasing order of the values $\left\{c^k_{pp_i}, i = 1..., n\right\}$ according to the order induced by the variables, $(p_1, \ldots, p_n)$. The weighting vector $W' = (w'_1, \ldots, w'_n)$ is obtained by using a linguistic quantifier.

The order inducing variables $(p_1, \ldots, p_n)$ represent the importance of each partial profile. The more important the partial profile the better the user's necessities are represented. To compute these values the system uses the following function:

$$p_i = \frac{1}{n-1} \sum_{j=0|j\neq i}^{n} \beta_{ji}$$

Where $\beta_{ji} = \Delta\left(p_{ji}\right)$ and p_i represents a mean of the preferences provided by the user about the item x_i. These preferences are obtained from the preference relation, P''.

Finally, the system obtains the user profile, FP_u, that will be used in the recommendation phase.

$$FP_u = (c^1_{fp}, \ldots, c^t_{fp})$$

3.3. *Recommendation*

This is the most important phase of the Recommender System. Once the user profile $FP_u = (c^1_{fp}, \ldots, c^t_{fp})$, has been computed the system should recommend the closest items to the user's necessities.

The item database $X = \{x_1, x_2, ..., x_m\}$, contains all the items that could be recommended. Where each item x_i, is described by a set of features $x_i = \left\{c^1_i, \ldots, c^t_i\right\}$.

The process of computing the most similar items to the user profile consists of comparing each item description, x_i, with the user profile, FP_u. To do so, the system will compute the similarity between the item, x_i, and the user profile. We propose a measurement of similarity based on the cosine of two vectors [45] defines as (see figure 4):

Definition 8. *The similarity between the user profile, FP_u and the item x_i is obtained as*

$$Similarity\,(FP_u, x_i) = cos\left(\overrightarrow{FP_u}, \overrightarrow{x_i}\right) = \frac{\overrightarrow{FP_u} \cdot \overrightarrow{x_i}}{||FP_u|| \cdot ||x_i||}$$

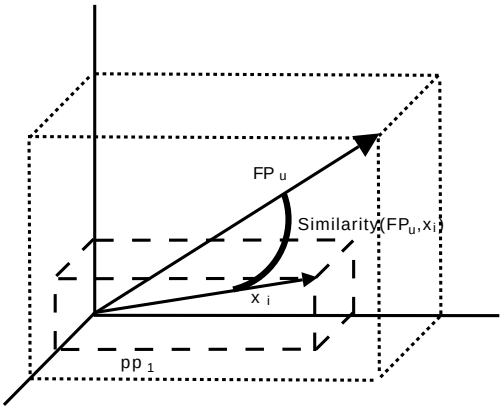


Fig. 4. Similarity between the final user profile and an item

Finally, the system will recommend to the user a set of k items, $X^B = \{x_1, ..., x_k\}$, that verifies:

(1) If $x_i \in X^B$ then $x_i \notin X_u$.
(2) $Similarity\,(FP_u, x_i)\,, x_i \in X^B$ are the top-k greatest values of similarity of the $x_i \in X - X_u$

4. Application of the Knowledge Based Recommender System based on Incomplete Linguistic Preference Relations

Here we present an application of the proposed model. Let $X = \{x_1, x_2, ..., x_m\}$ be a database of music CDs, in which every item is described by a set of features assessed in $[0, 1]$ (see Table 1). The system will use these features to compute the recommendations, but the user will receive a brief description easy to understand about the items (Singer, duration, music style,...). Following, it is showed in detail the recommendation process of our system.

Table 1. Products database

	Description									
Item	c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
item x_1	1.0	0.2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
item x_2	1.0	0.3	1.0	1.0	0	0	1.0	0	1.0	1.0
item x_3	0.5	0.1	0.4	0.8	1.0	1.0	1.0	0.4	0.9	0.9
item x_4	0.1	0.3	0.3	0.3	1.0	0	0.78	0	0.85	0.95
item x_5	0.74	0.37	0.26	0.41	0.39	0.86	0.22	0.05	0.62	0.62
item x_6	0.36	0.52	0.74	0.28	0.42	0.14	0.76	0.12	0.36	0.59
...	...	...	...	...	...	...	...	...	...	...
item x_8	0.55	0.012	0.81	0.88	0.45	0.97	0.13	0.60	0.88	0.49
...	...	...	...	...	...	...	...	...	...	...
item x_{11}	0.20	0.18	0.61	0.93	0.28	0.49	0.78	0.88	0.49	0.67
...	...	...	...	...	...	...	...	...	...	...
item x_{21}	0.82	0.30	0.89	0.46	0.38	0.12	0.26	0.27	0.57	0.49
item x_m	...	...	...	...	...	...	...	...	...	...

(i) Acquiring the user preference information.

The system will show the set $X_r = \{x_1^r, x_2^r, ..., x_{m'}^r\}$ $(m' \leq m)$ of the most illustrative examples of the system, and the user will select the four closest examples of his/her necessities (see Table 2). Let's suppose the user has chosen as examples of his/her necessities the CDs *Spirit*, *0304*, *Cry*, *No Angel*, being $X_u = \{x_{11}, x_{12}, x_{32}, x_{41}\}$.

Table 2. Given examples

		Hidden description									
CD	Item	c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
...	...	...	...	...	...	...	...	...	...	...	...
Spirit	x_{11}	1.0	0.2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
0304	x_{12}	1.0	0.3	1.0	1.0	0	0	1.0	0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...
Cry	x_{32}	0.5	0.1	0.4	0.8	1.0	1.0	1.0	0.4	0.9	0.9
...	...	...	...	...	...	...	...	...	...	...	...
No Angel	x_{41}	0.1	0.3	0.3	0.9	1.0	0	0.78	0	0.85	0.95
...	...	...	...	...	...	...	...	...	...	...	...
Item	$x^r_{m'}$	...	...	...	...	...	...	...	...	...	...

(a) Setting the favourite examples.

The user assesses his/her preferences in the linguistic term set,S:

$$S = \{s_0 = VL, s_1 = L, s_2 = I, s_3 = H, s_4 = VH\},$$

where VL, L, I, H, VH stand for "*Very Low*","*Low*", "*Indifference*", "*High*" and "*Very High*" respectively.

The user provides an incomplete preference relation:

$$P = \begin{pmatrix} (s_2,0) & (s_0,0) & (s_1,0) & (s_3,0) \\ & (s_2,0) & & \\ & & (s_2,0) & \\ & & & (s_2,0) \end{pmatrix}$$

(b) Completing the preference relation.

The system completes the user's preference relation by using the algorithm presented in section 2.5 obtaining a complete preference relation, P'':

$$P'' = \begin{pmatrix} - & (s_0,0) & (s_1,0) & (s_3,0) \\ (s_3,0.5) & - & (s_3,0) & (s_4,0) \\ (s_3,-0.47) & (s_1,0) & - & (s_3,0.17) \\ (s_2,-0.41) & (s_0,0) & (s_1,-0.17) & - \end{pmatrix}$$

(ii) Building the user profile.

First the system builds the partial profiles, and then, obtains the user profile from the partial profiles. To do so, it uses the IOWA operator. The weighting vectors used by this operator in both processes are, W and W', respectively. In this example, the weights are computed by using a linguistic quantifier represented by the values $a = 0$ and $b = 0.5$ in the equation (1). The vectors obtained are $W = (0.67, 0.33, 0)$ and $W' = (0.5, 0.5, 0, 0)$.

(a) Building the partial user profiles.

Following the process presented in 3.2.1 the partial profiles obtained can be seen in the Table 3 and its computation is as follows:

$$c^1_{pp_1} = F_W\left(\langle(s_3, 0.5), 1\rangle, \langle(s_3, -0.47), 0.5\rangle, \langle(s_2, -0.41), 0.1\rangle\right) =$$

$$= (0.67, 0.33, 0)\begin{pmatrix} 1 \\ 0.5 \\ 0.1 \end{pmatrix} = 0.67 + 0.33 \cdot 0.5 = 0.83$$

Table 3. Partial profiles

	Description									
Partial profile	c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
pp_1	0.83	0.23	0.8	0.93	0.33	0.33	1	0.13	0.97	0.97
pp_2	0.67	0.13	0.6	0.87	1	1	1	0.6	0.93	0.93
pp_3	1	0.27	1	1	0.33	0.33	1	0.33	1	1
pp_4	0.83	0.23	0.8	0.93	0.33	0.33	1	0.13	0.97	0.97

(b) Obtaining the user profile.

The user profile is computed by aggregating the partial profiles with the weighting vector W' and the order inducing variables, (p_1, p_2, p_3, p_4), computed as:

$$p_1 = \frac{1}{4-1}\sum_{j=0|j\neq 1}^{n}\beta_{j1} = \frac{1}{3}(3.50 + 2.53 + 1.59) = 2.54$$

$$p_2 = \frac{1}{4-1}\sum_{j=0|j\neq 2}^{n}\beta_{j2} = \frac{1}{3}(0 + 1 + 0) = 0.33$$

$$p_3 = \frac{1}{4-1} \sum_{j=0|j\neq 3}^{n} \beta_{j3} = \frac{1}{3}(1+3+0.83) = 1.61$$

$$p_4 = \frac{1}{4-1} \sum_{j=0|j\neq 4}^{n} \beta_{j4} = \frac{1}{3}(3+4+3.17) = 3.39$$

Then the final user profile obtained is showed in the Table 4 and its computation is carried out as:

$$c_{fp}^1 = F'_W(\langle 2.54, 0.83\rangle, \langle 0.33, 0.67\rangle, \langle 1.61, 1\rangle,$$

$$\langle 3.39, 0.83\rangle) = (0.5, 0.5, 0, 0) \begin{pmatrix} 0.83 \\ 0.83 \\ 1 \\ 0.67 \end{pmatrix} = 0.83 \cdot 0.5 + 0.83 \cdot 0.5 = 0.83$$

Table 4. Profile FP_u

FP_u									
c^1	c^2	c^3	c^4	c^5	c^6	c^7	c^8	c^9	c^{10}
0.83	0.23	0.8	0.93	0.33	0.33	1	0.13	0.97	0.97

(iii) Recommendation.

The system will recommend the k-closest items in the database to the user profile according to its rules introduced in section 3.3, (see Table 5).

Table 5. Recommendations

Product ID	Similarity
item x_2	0.97
item x_{11}	0.93
item x_1	0.90
item x_{21}	0.89
...	...

Let's suppose in this case that $k = 3$. Then the recommended items will be $X^B = \{x_2, x_1, x_{21}\}$, because $x_{11} \in X_u$.

5. Conclusions

From a limited knowledge of users' needs and preferences Recommender Systems will help them to find the most suitable items from a huge range of alternatives contained in e-shops. Different types of Recommender Systems exist, such as the Content-based and the Collaborative Filtering Recommender Systems. Generally, both of these systems provide good recommendations. However, they present some drawbacks related to the management of uncertainty and to the lack of information that can yield unsuccessful results.

In order to reduce the impact of those drawbacks, we have proposed a Knowledge Based Recommender Systems that gathers an incomplete linguistic preference relation which only requires a single row to be provided. From this row the preference relation can be completed by using the consistency property in order to exploit it and obtain better recommendations. Such a system is also less time-consuming in the information gathering process for the user.

References

1. S. Alonso, F. Chiclana, F. Herrera, E. Herrera-Viedma, J. Alcalá, and C. Porcel, "A consistency based procedure to estimate missing pairwise preference values", *International Journal of Intelligent Systems* **23** (2008) 155–175.
2. C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: Using social and contentbased information in recommendation", *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Nov 1998, pp. 714–720.
3. D. Ben-Arieh and Z. Chen, "Linguistic group decision-making: Opinion aggregation and measures of consensus", *Fuzzy Optimization and Decision Making* **5** (2006) 371–386.
4. G. Bordogna and G. Passi, "A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation", *Journal of the American Society for Information Science* **44** (1993) 70–82.
5. R. Burke, "Knowledge-based recommender systems", *Encyclopedia of Library and Information Systems* **69** 2000.
6. R. Burke. Hybrid recommender systems, "Survey and experiments", *User Modeling and User-Adapted Interaction* **12** (2002) 331–370.
7. R. D. Burke, K. J. Hammond, and B. C. Young, "Proc. Knowledge-based navigation of complex information spaces", Nov. 1996, pp. 462–468.
8. R. D. Burke, K. J. Hammond, and B. C. Young, "The findme approach to assisted browsing", *IEEE Expert* **12** (1997) 32–40.
9. P. Chang and Y. Chen, "A fuzzy multicriteria decision making method for technology transfer strategy selection in biotechnology", *Fuzzy Sets and Systems* **63** (1994) 131–139.
10. Z. Chen and D. Ben-Arieh, "On the fusion of multi-granularity linguistic lable sets in group decision making", *Computers and Industrial Engineering* **51** (2006) 526–541.
11. F. Chiclana, F. Herrera, and E. Herrera-Viedma, "Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations", *Fuzzy Sets and Systems* **97** (1998) 33–48.
12. R. Degani and G. Bortolan, "The problem of linguistic approximation in clinical decision making" *International Journal of Approximate Reasoning* **2** (1988) 143–162.

13. M. Delgado, J.L. Verdegay, and M.A. Vila, "Linguistic decision making models", *International Journal of Intelligent Systems* **7** (1992) 479–492.
14. M. Delgado, J.L. Verdegay, and M.A. Vila, "On aggregation operations of linguistic labels", *International Journal of Intelligent Systems* **8** (1993) 351–370.
15. D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications* (Academic Press, 1980).
16. J. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support* (Springer, 1994).
17. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry", *Communications of the ACM* **35** (1992) 61–70.
18. H.R. Guttman, "Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce", Ph.D. Thesis, 1998.
19. K.J. Hammond, *Case-Based Planning: Viewing Planning as a Memory Task*, (Academic Press Inc, 1989).
20. F. Herrera and E. Herrera-Viedma, "Linguistic decision analisys: Steps for solving decision problems under linguistic information", *Fuzzy Sets and Systems* **115** (2000) 67–82.
21. F. Herrera, E. Herrera-Viedma, and J.L. Verdegay, "A sequential selection process in group decision making with linguistic assessment", *Information Science* **85** (1995) 223–239.
22. F. Herrera and L. Martínez, "A 2-tuple fuzzy linguistic representation model for computing with words", *IEEE Transactions on Fuzzy Systems* **8** (2000) 746–752.
23. F. Herrera and L. Martinez, "A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making", *IEEE Transactions on Systems, Man and Cybernetics* **31** (2001) 227–234.
24. E. Herrera-Viedma, F. Herrera, F. Chiclana, and M. Luque, "Some issues on consistency of fuzzy preference relations", *European Journal of Operational Research* **154** (2004) 98–109.
25. E. Herrera-Viedma, L. Martínez, F. Mata, and F. Chiclana, "A consensus support system model for group decision-making problems with multigranular linguistic preference relations", *IEEE Transactions on Fuzzy Systems* **13** (2005) 644–658.
26. B. Krulwich, "Lifestyle finder: intelligent user profiling using large-scale demographic data", *AI Magazine* **18** (1997) 37–45.
27. M. Lalla, G. Facchinetti, and G. Mastroleo, "Ordinal scales and fuzzy set systems to measure agreement: An application to the evaluation of teaching activity", *Quality and Quantity* **38** (2004) 577–601.
28. C.K. Law, "Using fuzzy numbers in educational grading systems", *Fuzzy Sets and Systems* **83** (1996) 311–323.
29. H. Lee, "Group decision making using fuzzy sets theory for evaluating the rate of aggregative risk in software development", *Fuzzy Sets and Systems* **80** (1996) 261–271.
30. L. Martinez, L.G. Perez, and M. Barranco, "A multi-granular linguistic content-based recommendation model", *International Journal of Intelligent Systems* **22** (2007) 419–434.
31. F. McCarey, M. Cinnéide, and N. Kushmerick, "Rascal: A recommender agent for agile reuse", *Artificial Intelligence Review* **24** (2006) 253–276.
32. D. McSherry, "Increasing dialogue efficiency in case-based reasoning without loss of solution quality", *In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003, pp. 121–126.
33. M. J. Pazzani, J. Muramatsu, and D. Billsus, "Syskill webert: Identifying interesting

web sites", *AAAI/IAAI, Vol. 1*, Nov. 1996, pp. 54–61.
34. J.B. Schafer, J.A. Konstan, and J. Riedl, "E-commerce recommendation applications", *Data Mining and Knowledge Discovery* **5** (2001) 115–153.
35. T. Tanino, "Fuzzy preference orderings in group decision making", *Fuzzy Sets and Systems* **12** (1983) 117–131.
36. T. Tanino, *Non-Conventional Preference Relations in Decision Making chapter Fuzzy Preference Relations in Group Decision Making* (Springer-Verlag, 1988) pp. 54–71.
37. T. Tanino, *Multiperson Decision Making Using Fuzzy Sets and Possibility Theory chapter On Group Decision Making under Fuzzy Preferences* (Kluwer Academic Publisher, 1990) pp. 172–185.
38. M. Tong and P.P, "Bonissone. A linguistic approach to decision making with fuzzy sets", *IEEE Transactions on Systems, Man and Cybernetics* **10** (1980) 716–723.
39. Zeshui Xu, "Incomplete linguistic preference relations and their fusion", *Information Fusion* **7** (2006) 331–337.
40. R.R. Yager, *Fuzzy Logic: State of the Art chapter Fuzzy Screening Systems* (Kluwer Academic Publisher, 1993) pp. 251–261.
41. R.R. Yager, L.S. Goldstein, and E. Mendels, "FUZMAR: An approach to aggregating market research data based on fuzzy reasoning", *Fuzzy Sets and Systems* **68** (1994) 1–11.
42. R.R. Yager, "An approach to ordinal decision making", *International Journal of Approximate Reasoning* **12** (1995) 237–261.
43. R. R. Yager, "Induced aggregation operators", *Fuzzy Sets and Systems* **137** (2003) 59–69.
44. R.R. Yager, "Fuzzy logic methods in recommender systems", *Fuzzy Sets and Systems* **136** 133–149, 2003.
45. Y. Yang, "Expert network: Effective and efficient learning from human decisions in text categorization and retrieval", *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland, Jul. 1994) pp. 14–22.
46. L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning- I, II, III", *Information Sciences*, **8-9** (1975) 199–249, 301–357, 43–80.

D.8. REJA. REcommender System of Restaurants in JAén

In this chapter we will introduce a recommender system for restaurants in the province of Jaén that implements some of the proposals presented in our memory.

The development of this system was supported by a project of the Office of Turism, Commerce, and Athletics of the Andalusian Government (Consejería de Turismo, Comercio y Deporte de la Junta de Andalucía) as part of the 2006 official programme Subsidies for the development of campaigns to raise public awareness of the touristic quality of Andalusia and thanks to a grant published the 21st of March 2006, and granted the 16th of August 2006.

REJA is a highly functional hybrid recommender system implemented upon a database of restaurants in the province of Jaén. The recommender system is composed of two different recommender models:

1. A collaborative filtering recommender system
2. A knowledge based recommender system with linguistic preference relations.

Our aim with this system was to develop a recommender system that can make recommendations where, due to the features of the contexts, classic recommender systems are unable to make them. For example, it is hard in some situations to implement recommender systems of restaurants or leisure centres, since, they present they following problems:

- Many of their users are casual users that will not interact with the system again, or they will hardly ever interact. Classic recommender systems are unable to make recommendations as they do not have any historical information about them.

- Most of the users will have knowledge based on expectations about what they want or need.

- It is common that they system deals with users that do not want to receive recommendations based on their historical information. It could be caused because they want to celebrate something not related with their usual behaviour, for example, a birthday. In such cases, the historical information is not relevant, and it should not be used.

Following we will explain the recommender systems mentioned and the user interface of REJA.

D.8.1. Hybridation outline

In this recommender system, it was used the commutation mechanism for hybridizing the recommendations algorithm [31]. The commutation mechanism choose which recommendation algorithm is going to be used depending on the situation. This way of hybridation presents several advantages:

- It is easy to implement, since it is very straightforward.

- As the recommendation algorithms are independent, their implementation is the same as they were used by its own without any hybridation.

This hybridation was used, since it seemed to be the best option for our purposes.

If users want to receive recommendations, they must indentify themselves in the system. Depending on how they want to receive these recommendations, and the information they have provided, the system will choose the collaborative filtering algorithm or the knowledge based one:

- It will use the collaborative filtering one, when users want to receive recommendations based on their historical information.
- It will use the knowledge based one, when users want to receive recommendations by giving examples that represent their necessities. The model implemented here, is the same as the RRPI mode presented in section D.7.

Collaborative System

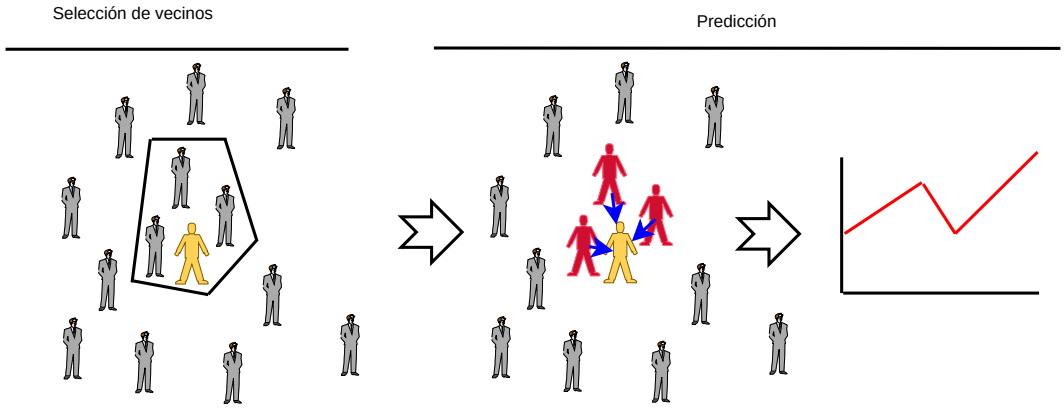
In the collaborative recommender system the recommendations are exclusively based on the resemblance terms among the users. The operation of this type of recommender system runs as follows (see figure D.5):

1. The system saves a profile for each user which includes the database items that the user knows and has evaluated.
2. The degree of resemblance among the different users is measured on the basis of their profiles. Hence, groups of users with similar characteristics are created.
3. The system will use all the information obtained in the previous actions in order to make the recommendations. It will recommend to each user items they have not evaluated yet and that have been positively evaluated by the remains of the group.

Consequently, these recommender systems do not take into account the contents and characteristics of the products they recommend, but the taste of those users who share a similar profile with the user who requires the service.

In REJA we have used a collaborative filtering engine called CoFE that could found in the url (http://eecs.oregonstate.edu/iis/CoFE/). Due to the fact, that

Figure D.5: How collaborative filtering algorithm works



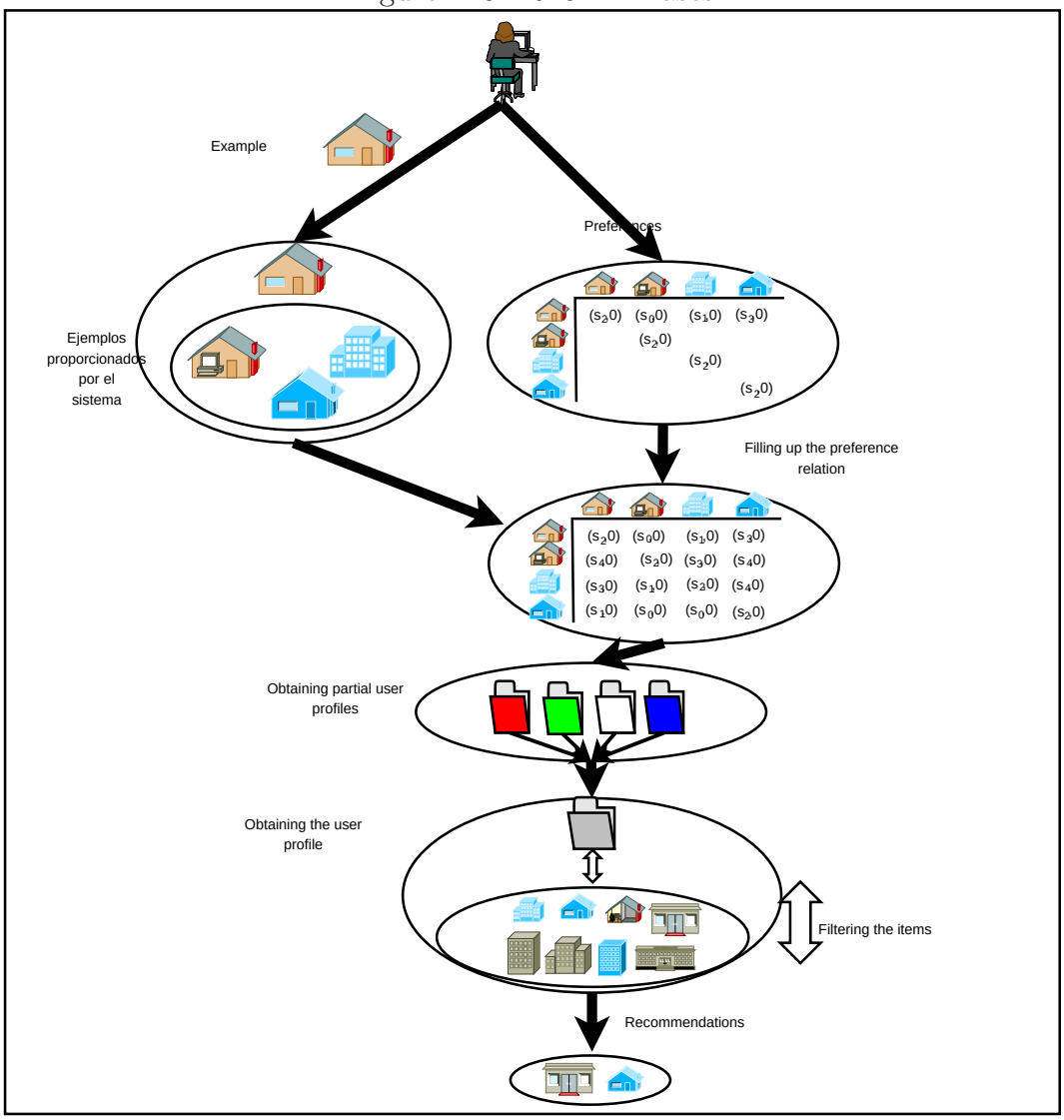
there is a great variety of colaborative filtering recommender system in the literature and we have used a standard implementation based on CoFE, we will not present in detail how this model works since all the information can be found in [178, 66, 70, 71, 73, 74, 72, 75, 112, 141, 149].

Knowledge based recommender system

The most commonly used recommender systems (collaborative and content-based) have a series of constraints that can have an influence on the quality of the generated recommendations. These recommender systems need historical ratings (or "information") of the past actions of the users in order to generate recommendations. However, this information is seldom useful, adequate, or available, when the users are under the circumstances as customers who visit the city for the first time -and, therefore, it is the first time they use our system-; users that have used the system before, but they wish to have lunch in a restaurant for a special occasion (birthday party, retirement celebration...) unrelated to any of their past actions; or users who, have not provided yet enough information to allow the collaborative system to make accurate recommendations.

This module is based on the recommender system model RRPI presented in section D.7. The phases of this model can be seen in figure D.6.

Figure D.6: REJA. Phases



This implementation presents a slightly difference with the model presented in section D.7. Meanwhile in the model users choose four examples of their necessities, in the implementation users choose the first one, and the others are chosen by the system. Two of them are the most recent user rated restaurants and the

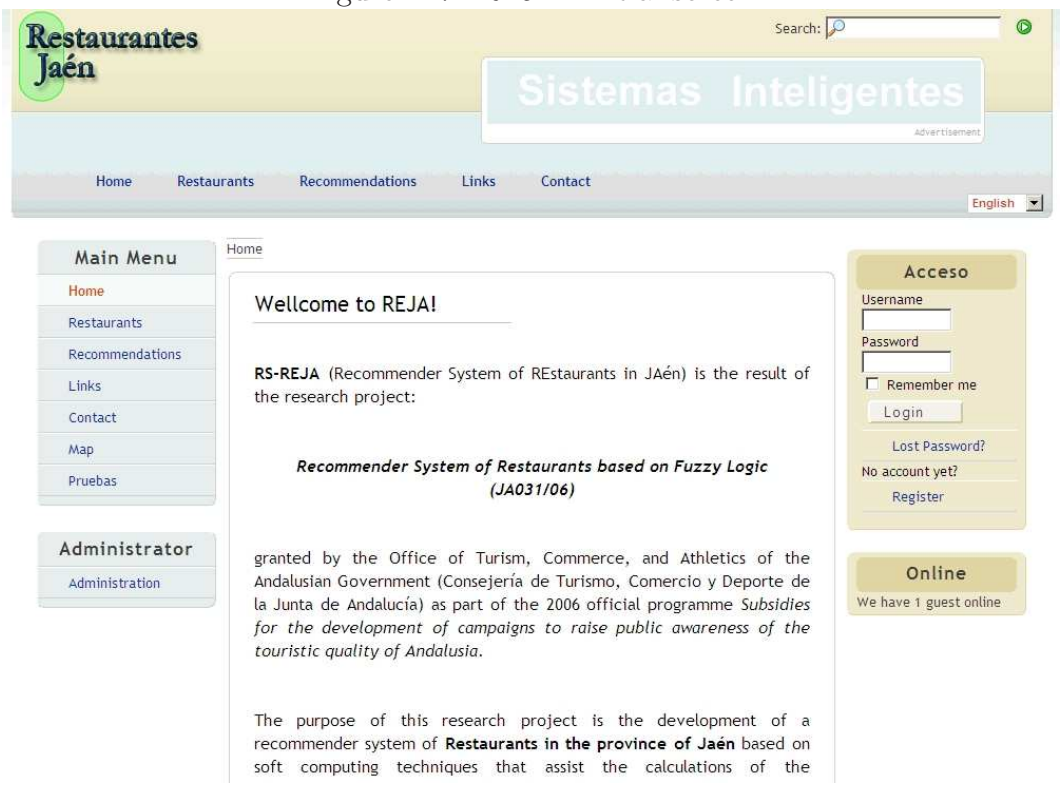
other one is the furthest similar to the selected one.

D.8.2. User interface

REJA is a web site with information about restaurants that, offers a hybrid recommender system of restaurants in the province of Jaén.

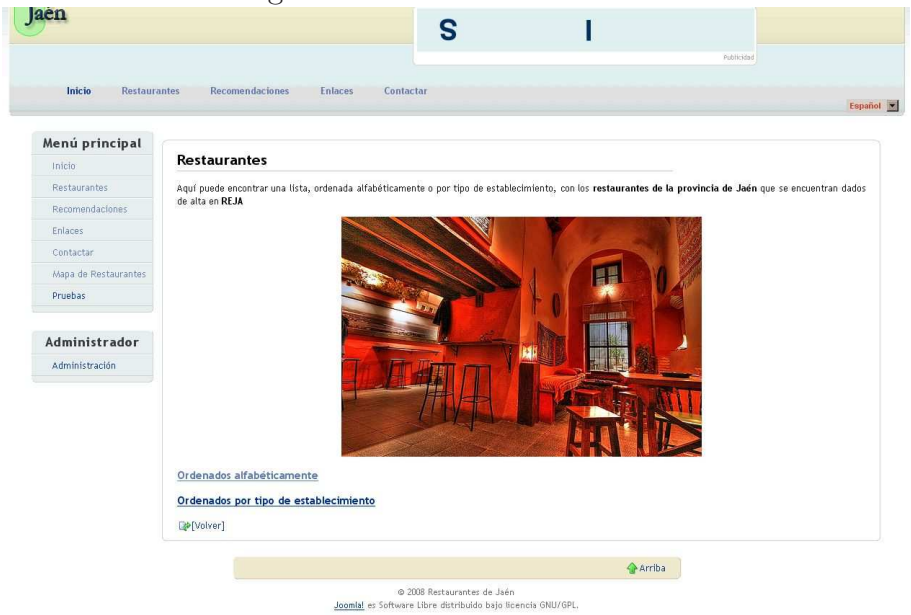
Here we present an example of how REJA works. Once, the user visits the web site, the system will show a screen as it appears in figure D.7.

Figure D.7: REJA. Initial screen



To obtain information about a restaurant, users can click on the option Restaurants on the left part main screen. Once the user has clicked this option, it will appear a screen (see figure D.8) where they can choose the restaurant by alphabetic order or by the type of restaurant.

Figure D.8: REJA. Restaurants



Once one of the options has been chosen, the user must choose the restaurant he/she wants to get information. The system then will show all the information about it (see figure D.9).

Figure D.9: REJA. Information about a restaurant

Moreover, this system provides a Geographic Information System module (see

figure D.10) that lets users know the location of the restaurant and show a route to go there. Optionally, it can show interesting places nearby the restaurant.

Figure D.10: REJA. Geographic Information Systems

Now, we will show, the use of the collaborative filtering and knowledge based modules.

Collaborative Filtering Module

First of all, the user has to be identified in our system. Hence, users introduce their login and their password in the right area of the initial screen (see figure D.7). This login and password have to be provided by the system after customer requirement.

Figure D.11: REJA. Main screen once users have been identified

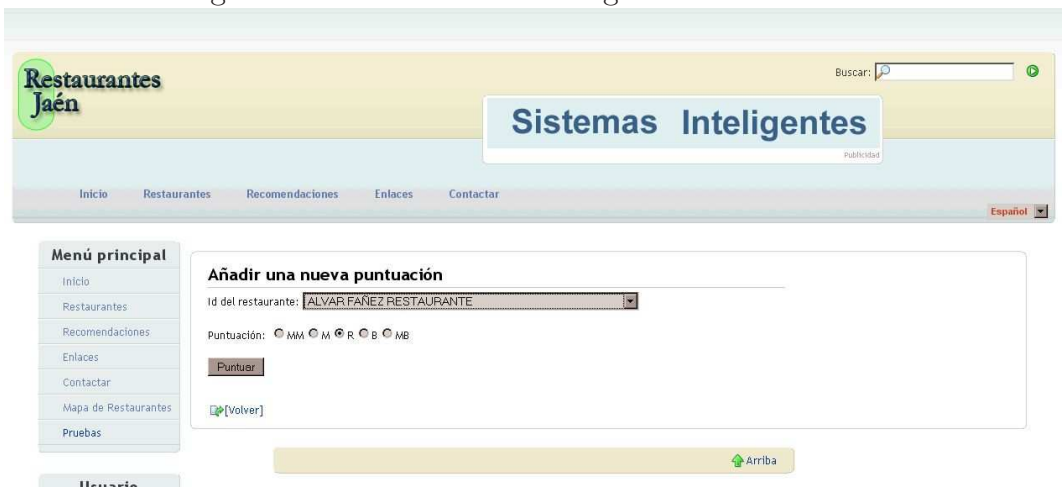
One we have logged in, the system will show a welcome screen as we see in the figure D.11. In this screen the system offers two options:

1. *To add a new rating:* in which we can provide more information about the restaurants we have visited by giving the opinion about them.

2. *To obtain recommendations:* in which the system will provide recommendations to the users.

When users choose the first option, *to add a new rating*, the system goes to the screen showed in figure D.12 where the ratings can be added. In order to do that, users must choose a restaurant, and then they will give a rating by using the scale: very bad, bad, regular, good, very good. This information will be used by the collaborative module in order to make recommendations.

Figure D.12: REJA. Add ratings to the restaurants

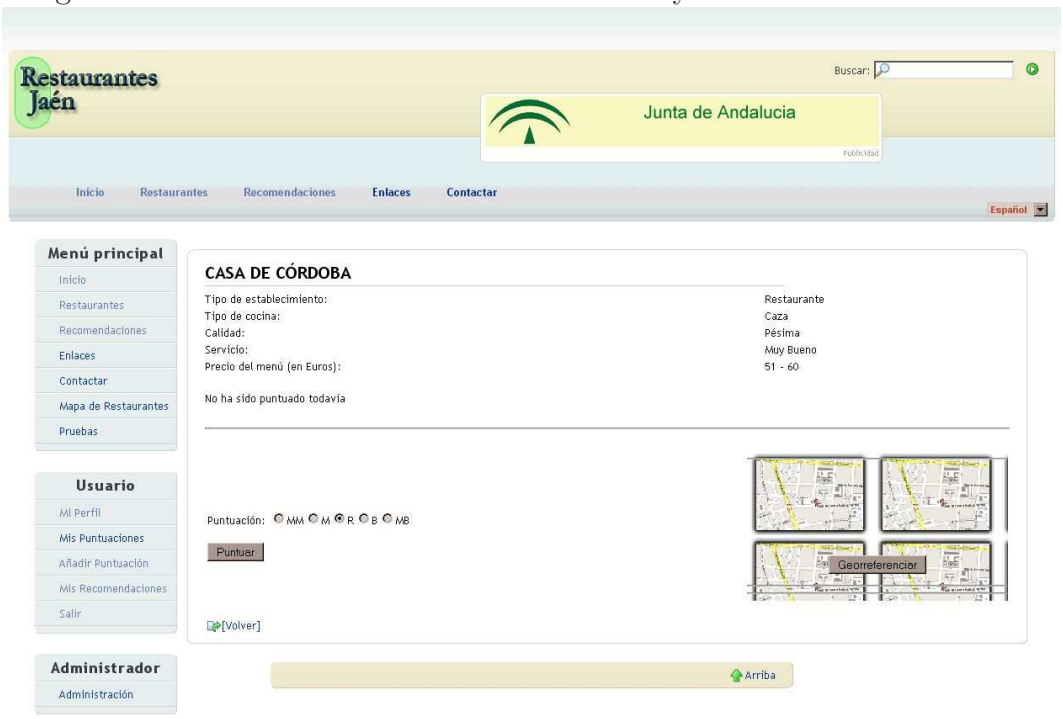


Once users have introduced enough ratings, they can recieve recommendations made by collavorative filtering algorithm clicking the option *Obtaining recommendations* in the wellcome screen (see figure D.11) and the system will provide recommendations, see figure D.13.

Figure D.13: REJA. Recommendations from the collaborative filtering module

When the users click in one of the restaurants, the system will show information about the restaurant (see figure D.14), and they can also provide their ratings.

Figure D.14: REJA. Recommendations made by the collaborative module



Moreover, REJA offers another option related to the collaborative filtering module. The option *My ratings* that users can find on the left part of the welcome screen. If an user click it, they system will show the user profile used by the collaborative filtering module, that is, all the ratings the user has provided about the restaurants the user has visited and rated (see figure D.15).

Figure D.15: REJA. User profile used by the collaborative filtering module

RRPI Module

To use this knowledge based recommendation system, the users should log in the system from the initial screen. Once in the welcome screen (figure D.11). They will click the option *Recomendations* on the left side and the users will see the interface for the knowledge based recommendation system (see figure D.16). In this screen, users have to provide a restaurant that represents their necessities.

Figure D.16: REJA. Recommendations to registered users

The system then will offer others three restaurants (two restaurants previously rated by the user and another the furthest similar to the selected one to the selected one) that users must compare with the first one (see figure D.18) in order to provide a incomplete preference relation. In order to do that, users provide the preference of the chosen restaurant over the others using a linguistic scale: much better, better, equal, worse and much worse. In the figure D.17, we can see in which phase the RRPI model is at this point.

Figure D.17: Model RRPI in REJA. The user provides an example and a preference relation over the examples

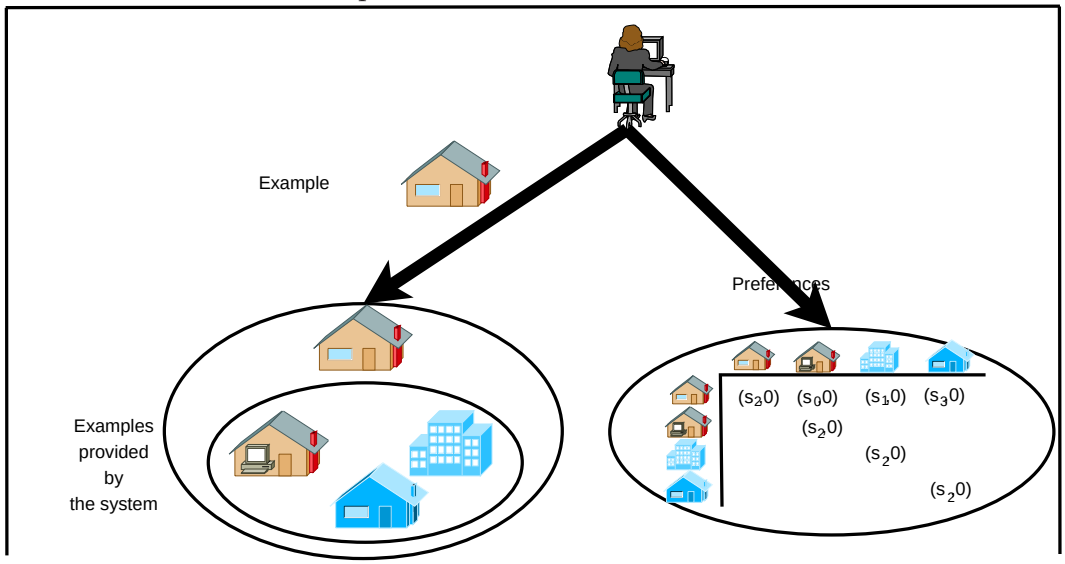
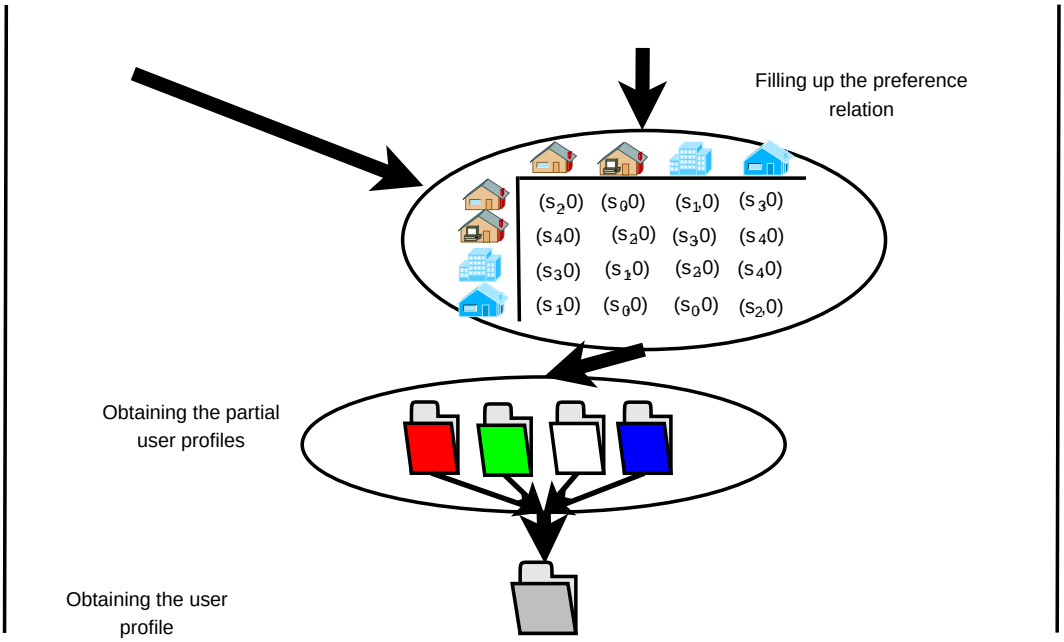


Figure D.18: REJA. Providing preference information to the knowledge based recommendation system

According to these preferences, the system will assemble a preference matrix which will be essential to compute the user's profile (see figure D.19).

Figure D.19: Model RRPI in REJA. Obtaining the user profile



From this user profile, the system will make the recommendation (see figure D.20) that will be given to the users (see figure D.21).

Figure D.20: Model RRPI in REJA. Making the recommendations

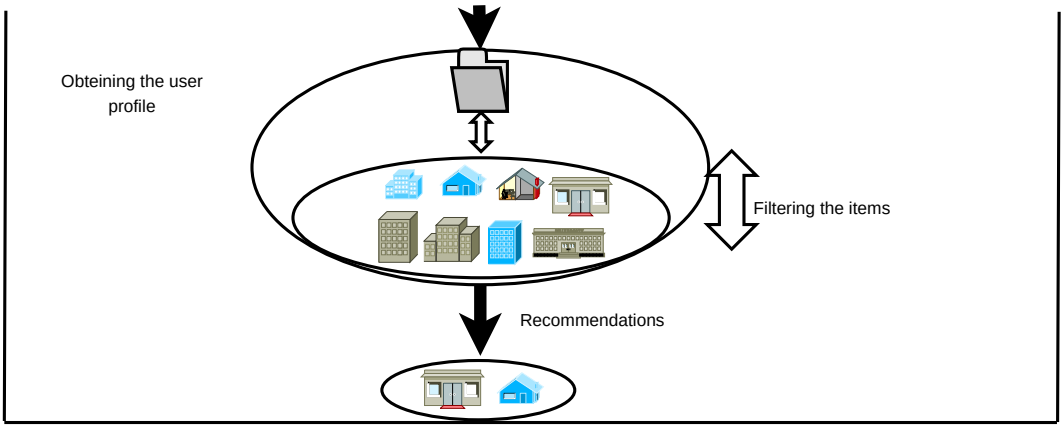
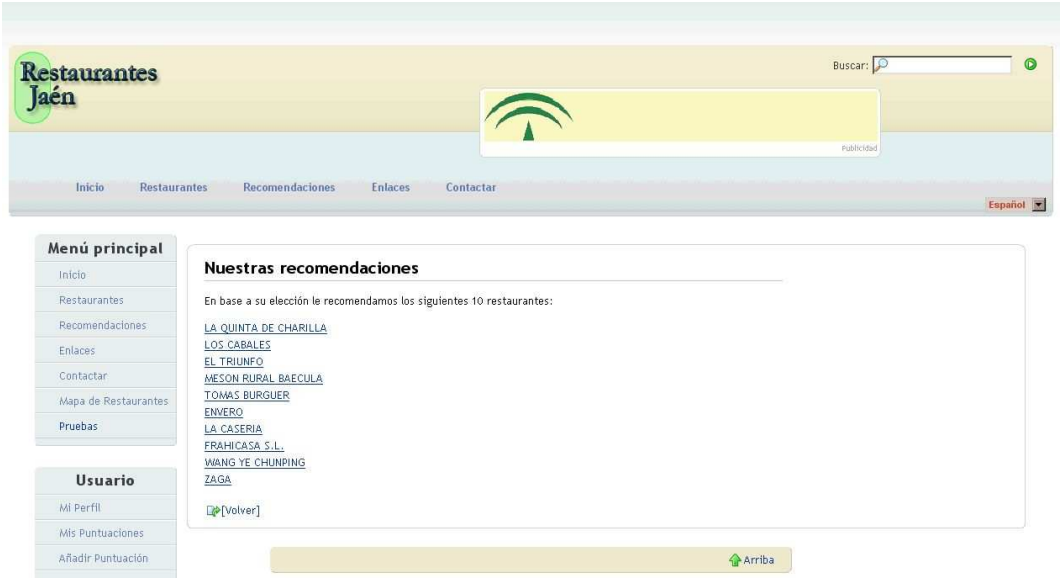


Figure D.21: REJA. Obtained recommendations



The same as the previous collaborative recommender module, when users click on the restaurant name, the system will show the information about such a restaurant (see figure D.22).

Figure D.22: REJA. Information about a restaurant

D.8.3. Conclusions

In this section, we have presented an implementation of a hybrid recommender system for making recommendations of restaurants in the province of Jaén. The hybridization technique used by this system is known as commutation. This system is composed of two modules:

1. A collaborative filtering module.
2. A knowledge based recommendation module based on the RRPI model presented in section D.7.

In the beginning of this summary, we stated our aims. The main was to develop a prototype of a recommender system.

In this chapter, we have presented this prototype, a hybrid recommender system of restaurants in the province of Jaén. The main contributions and features are the following ones:

- It offers a greater flexibility than the classic recommender system, since it lets the users to provide their preferences by using linguistic assessments.
- It improves the information gathering processes in order to define the user profile. The knowledge based recommender system builds the user profile from an incomplete preference relation, that will be completed automatically by the system, over a small set of restaurants.
- It is able to make recommendations, even when there is no historical information or it is irrelevant by means of the knowledge based module.
- This prototype was designed to make recommendations in the tourist sector, spreading this type of tools in a sector which lacked them because of varied

typical difficulties and features of the items of this sector.

D.9. Conclusions and future works

Finally, we will review our main proposals and the results we have obtained in this report. At the end, we will review our future works and research areas we will develop from these results.

D.9.1. Proposal and obtained results

Recommender systems make personalized recommendations to the customers, making the purchase processes in e-shop more enjoyable, faster, and personal. The aim of these systems is to offer their users the most interesting and suitable products according to their tastes and necessities. To learn these tastes or preferences, many of these systems use the users' historical information about which items they have bought or rated. Although this systems have been used successfully in many situations, as this information was available, there are other situations where the information is not available or relevant, but, however, customers would like to receive recommendations.

For these reasons, I established the following aims for this research:

- To study how to model the information in this kind of problems.
- To improve the users' information gathering process in recommender system in order to define the user profile.
- To design recommender system models that are able to make recommendations in situations where there is no historical information about their past opinion, tastes, rates or purchase or the historical information is scarce.
- To implement a prototype of a recommender system that can be used in the touristic sector.

According to these aims, we have developed three proposals of recommender system models. Now we will give a brief explanation about them:

1. **Multigranular linguistic Content-based recommender model:** The main features of this model are the following:

   - It makes recommendations without needing historical information. Our proposal let the users define their own user profiles, and hence, the historical information is not needed now.
   - It deals with linguistic information. As the linguistic domain is more suitable as we are modeling subjective information, tastes and preferences and/or gathered by means of perceptions, and thus, they have got a high degree of uncertainty
   - It deals with multigranular information. This way, users and experts could use the most suitable linguistic scale according to their degree of knowledge and nature of the feature that is being evaluated.

2. **Knowledge based recommender system models:** when we reviewed the literature about recommender system we realized that there were new models that made recommendation without historical information. Our following step was to improve these models by defining new ones:

   a) *Knowledge based recommender system model with multigranular linguistic information (or RML):* in this model the user profile is defined from an example, given by the user, that represents his/her necessities. Sometimes this example does not represent exactly their necessities and it is required a refinement phase. The main improvements with regards to classic knowledge based recommender systems are:

- It deals with linguistic information.
- It deals with multigranular linguistic information.

Moreover it offers some advantages over the content-based model such as:

- It is easier the use profile to be defined, it is more complete, and therefore, better results are expected to be obtained.
- The computations to make recommendations are more efficient, and hence, this model can deal with bigger database and still make recommendations within acceptable limits of times.

b) *Knowledge based recommender system model with linguistic preference relations (or RRPI:)* one of the inconvenience of knowledge based recommender systems is the refinement phase of user profile. This phase can be time-consuming and complex since the profile can be composed of many features and some of them cannot be understandable by the users. Due to the fact that our first studies were focused on preference relations and how to fill them, we realize that these ones could be useful to improve the knowledge based recommender system models. The main improvements of this model are:

- There is no refinement phase. The user profile is defined from four examples of the user's necessities.
- It minimizes the information needed to make recommendations. The user profile is defined from a preference relation provide by the user, but it does not expect to be complete. In real situations it is not easy for a user, to provide complete and consistent preference relations since it can be time-consuming. In this model

we have included mathematical tools to allow the users to provide incomplete relations that the model is able to fill up by itself.

- It deals with linguistic information.

This last model, the RRPI, has been one of the model used to implement a hybrid recommender system of Restaurants in the province of Jaen.

Regarding the scientific spreading and publications of our scientific results, we will highlight the following contributions:

- Luis Martínez, Luis G. Pérez and Manuel Barranco, A Multi-granular Linguistic Content Based Recommendation Model. International Journal of Intelligent Systems. 22:5, 2007, pp.419-434.

- Luis Martínez, Luis G. Pérez, Manuel Barranco and Macarena Espinilla. A Knowledge Based Recommender System with Multigranular Linguistic Information. International Journal of Computational Intelligence Systems, 1(2), 2008, pp. 148-158.

- Luis Martínez, Luis G. Pérez, Manuel Barranco and Macarena Espinilla. Improving the Effectiveness of Knowledge Based Recommender Systems Using Incomplete Linguistic Preference Relations. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (Accepted for publication).

- Luis Martínez, Luis G. Pérez and Manuel Barranco. A Knowledge Recommender System based on Fuzzy Consistent Preference Relations. In the book: Intelligent Decision and Policy Making Support Systems. Serie: Studies in Computational Intelligence , Vol. 117. Editors: Da Ruan, Frank Hardeman and Klaas van der Meer (Springer, 2008).

- Luis G. Pérez, Manuel J. Barranco and Luis Martínez. Exploiting Linguistic Preference Relations in Knowledge Based Recommendation Systems. Eurofuse 2007. Jaén.
- Luis G. Pérez, Manuel Barranco and Luis Martínez. Building User Profiles for Recommender Systems from incomplete preference relations. FUZZ-IEEE 2007, IEEE International Conference on Fuzzy Systems. London.
- L. Martínez, M. Barranco, L. Pérez, M. Espinilla and F. Siles. A Knowledge Based Recommendation System with Multigranular Linguistic Information. In the 2007 International Conference on Intelligent Systems and Knowledge Engineering (ISKE2007), Oct 15-16, 2007, Chengdu, Chinna.

D.9.2. Future works and research

Taking into account these previous results, our future works and research will focus on the following research areas:

1. To offer a greater flexibility in the preference modeling. In order to do that, we will study how to adapt the models proposed in this summary to let them deal with heterogeneous information: numerical, interval and linguistic.
2. To study hybridization models more advanced than the proposed one, with the aim of making the matter of obtaining recommendations easier.
3. To study in depth how to deal with incomplete information in order to improve the information gathering processes in recommender systems.
4. To progress in the implementation of REJA to obtain a fully functional and commercial recommender system.

Bibliografía

[1] G.I. Adamopoulos and C.P. Pappis. A fuzzy linguistic approach to a multicriteria sequencing problem. *European Journal of Operational Research*, 92(3):628–636, 1996.

[2] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transaction on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] H. Akdag, I. Truck, A. Borgi, and N. Mellouli. Linguistic modifiers in a symbolic framework. *Journal Title: International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(Suppl):49–61, 2001.

[4] S. Alonso, F. Chiclana, F. Herrera, E. Herrera-Viedma, J. Alcalá, and C. Porcel. A consistency based procedure to estimate missing pairwise preference values. *International Journal of Intelligent Systems*, 23:155–175, 2008.

[5] J. Alspector, J. Koicz, and N. Karunanithi. Feature-based and clique-based user models for movie selection: A comparative study. *User Modeling and User-Adapted Interaction*, (7):279–304, 1997.

[6] B. Arfi. Fuzzy decision making in politics: A linguistic fuzzy-set approach. *Political Analysis*, 13(1):23–56, 2005.

[7] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12, 1995.

[8] W. Armstrong. Uncertainity and utility function. *Economics Journal*, 58:1–10, 1948.

[9] C. Avery and R. Zeckhauser. Recommender systems for evaluating computer messages. *Communications of the ACM*, 40(3):88–89, 1997.

[10] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[11] M. Balabanovic. An adaptive web page recommendation service. In W. Lewis Johnson and Barbara Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 378–385, New York, 1997. ACM Press.

[12] M. Balabanovic. Exploring versus exploiting when learning user models for text representation. *User Modeling and User-Adapted Interaction*, 8(1-2):71–102, 1998.

[13] M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[14] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of*

the Fifteenth National Conference on Artificial Intelligence, pages 714–720, 1998.

[15] A. Bechara, D. Tranel, and H. Damasio. Characterization of the decision-making deficit of patients with ventromedial prefrontal cortex lesions. *Brain*, 123:2189–2202, 2000.

[16] N.J. Belkin and W.B. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[17] B.L.D. Bezerra and F. de A. T. de Carvalho. A symbolic approach for content-based information filtering. *Information Processing Letters*, 92(1), 2004.

[18] H.K. Bhargava, S. Sridhar, and C. Herrick. Beyond spreadsheets: tools for building decision support systems. *IEEE Computer*, 3(32):31–39, 1999.

[19] D. Billsus and M.J. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000.

[20] A. Blum, L. Hellerstein, and N. Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *Journal of Computer and System Sciences*, 50(1):32–40, 1995.

[21] P.P. Bonissone. *Approximate Reasoning in Decision Analysis*, chapter A fuzzy sets based linguistic approach: Theory and applications, pages 329–339. North-Holland Publishing Company, North-Holland, 1982.

[22] P.P. Bonissone and K.S. Decker. *Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-Off Precision and Complexity*, chapter Uncertainty in Artificial Intelligence, pages 217–247. North-Holland, 1986.

[23] G. Bordogna, M. Fedrizzi, and G. Pasi. A linguistic modeling of consensus in group decision making based on OWA operators. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27:126–132, 1997.

[24] G. Bordogna and G. Pasi. A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation. *Journal of the American Society for Information Science*, (44):70–82, 1993.

[25] P. Bosc, D. Kraft, and F. Petry. Fuzzy sets in database and information systems: Status and opportunities. *Fuzzy Sets and Systems*, 3(156):418–426, 2005.

[26] B. Bouchon-Meunier and M. Rifqi. Resemblance in database utilization. In *Proceeding 6th IFSA World Congress, Sao Paulo*, 1995.

[27] B. Bouchon-Meunier, M. Rifqi, and S. Bothorel. Towards general measures of comparison of objects. *Fuzzy Sets and Systems*, 84(2):143–153, 1996.

[28] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Uncertainty in Artificial Intelligence. Proceedings of the Fourteenth Conference*, pages 43–52, 1998.

[29] Z. Bubnicki. *Analysis and Decision Making in Uncertain Systems*. Springer-Verlag, 2004.

[30] R. Burke. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, 69(32), 2000.

[31] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[32] R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *AAAI/IAAI,Vol. 1*, pages 462–468, 1996.

[33] R. D. Burke, K. J. Hammond, and B. C. Young. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40, 1997.

[34] H. Bustince and P. Burillo. Perturbation of intuitionistic fuzzy relations. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, 9(1):81–103, 2001.

[35] E. Capurso and A. Tsoukias. Decision aiding and psychotherapy. *Bulletin of the EURO Working Group on MCDA*, 2003.

[36] C. Carlsson and R. Fuller. *Fuzzy Reasoning in Decision Making and Optimization*. Springer-Verlag, 2001.

[37] G. Chang, M. Healey, J.A.M. McHugh, and T.L. Wang. *Mining the World Wide Web - An Information Search Approach*. Springer, 2001.

[38] S.M. Chen. A new method for tool steel materials selection under fuzzy environment. *Fuzzy Sets and Systems*, (92):265–274, 1997.

[39] Y.H. Cho and J.K. Kim. Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce. *Expert Systems with Applications*, 26(2):233–246, 2004.

[40] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D.Netes, and M. Sartin. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, 1999.

[41] M. Condliff, D. Madigan D.D. Lewis, and C. Posse. Bayesian mixed-effects models for recommender systems. In *Proeedings ACM SIGIR 99 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[42] C. Coombs and J. Smith. On the detection of structures in attitudes and developmental processes. *Psychological Reviews*, 80(5):337–351, 1973.

[43] U.S. Information Technology Industry Council. The protection of personal data in electronic commerce. *Public Policy Document*, 1997.

[44] R. Degani and G. Bortolan. The problem of linguistic approximation in clinical decision making. *International Journal of Approximate Reasoning*, (2):143–162, 1988.

[45] M. Delgado, F. Herrera, E. Herrera-Viedma, and L. Martínez. Combining numerical and linguistic information in group decision making. *Information Sciences*, 107:177–194, 1998.

[46] M. Delgado, J.L. Verdegay, and M.A Vila. Linguistic decision making problems. *International Journal of Intelligent Systems*, 7:479–492, 1992.

[47] M. Delgado, J.L. Verdegay, and M.A. Vila. On aggregation operations of linguistic labels. *International Journal of Intelligent Systems*, (8):351–370, 1993.

[48] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.

[49] L. Dombi. *Fuzzy Logic and Soft Computing*, chapter A General Framework for the Utility-Based and Outranking Methods, pages 202–208. World Scientific, 1995.

[50] J. Doyle. Prospects for preferences. *Computational Intelligence*, 20(2):111–136, 2004.

[51] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, 1980.

[52] D. Dubois and H. Prade. Ranking fuzzy numbers in the setting of possibility theory. *Information Sciences*, (30):183–224, 1983.

[53] M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, 2003.

[54] F. Estepa, P. García, L. Godó, E. Ruspini, and L. Valverde. On similarity logic and the generalized modus ponens. In *Proceedings of the FUZZY-IEEE'94*, 1994.

[55] Z-P. Fan, J. Ma, and Q. Zhang. An approach to multiple attribute decision making based on fuzzy preference information alternatives. *Fuzzy Sets and Systems*, 131(1):101–106, 2002.

[56] Z.P Fan and X. Chen. Consensus measures and adjusting inconsistency of linguistic preference relations in group decision making. *Lecture Notes in Artificial Intelligence*, 3613:130–139, 2005.

[57] Z.P. Fan, S.H. Xiao, and G.F. Hu. An optimization method for integrating two kinds of preference information in group decision-making. *Computers Industrial Engineering*, 46(2):329–335, 2004.

[58] J. Fodor and M. Roubens. *Fuzzy preference modelling and multicriteria decision support*. Kluwer Academic Publishers, Dordrecht, 1994.

[59] P.W. Foltz. Using latent semantic indexing for information filtering. In *Proceedings of the ACM SIGOIS and IEEE CS TC-OA conference on Office information systems*, pages 40–47, 1990.

[60] P. Fortemps and R. Slowinski. A graded quadrivalent logic for ordinal preference modelling: Loyola-like approach. *Fuzzy Optimization and Decision Making*, 1:93–111, 2002.

[61] C. García, F. Herrera, L. Martínez, L.G. Pérez, and P.J. Sánchez. Diagrama uml para modelos de toma de decisión con información heterogénea. In *XII Congreso Español sobre Tecnologías y Lógica Fuzzy*, pages 473–478, 2004.

[62] C. García, L.G. Pérez, L. Martínez, B. Montes, and P.J. Sánchez. An automatic eeducational quality eevaluation fuzzy system. In *IADAT- e 2004 International Conference in Education*, pages 28–32, 2004.

[63] RA. Gheorghe, A. Bufardi, and P. Xirouchakis. Fuzzy multicriteria decision aid method for conceptual design. *Cirp Annals-Manufacturing Technology*, 54(1):151–154, 2005.

[64] D. Goldberg, D.Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61 – 70, 1992.

[65] J. González-Pachóna, D. Gómez, J. Montero, and J. Yáñez. Searching for the dimension of valued preference relations. *International Journal of Approximate Reasoning*, 33(2):133–157, 2003.

[66] N. Good, J.B. Schafer, J.A. Konstan, A. Borchers, B. Sarwar, J.L. Herlocker, and J. Riedl. Combining collaborative filtering with personal agents for better recommendations. In *In Proceedings of AAAI-99, AAAI Press*, pages 439–446, 1999.

[67] X. Guo and J. Lu. Intelligent e-government services with recommendation techniques. *International Journal of Intelligent Systems on the special issue on E-Service intelligence*, 22(5):1–17, 2007.

[68] R.H. Guttman. *Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce*. PhD thesis, School of Architecture and Planning, Massachusetts Institute of Technology, 2006.

[69] K.J. Hammond. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press Inc, 1989.

[70] J.L. Herlocker. Understanding and improving automated collaborative filtering systems, 2000.

[71] J.L. Herlocker and J.A. Konstan. Content-independent, task-focused recommendation. *IEEE Internet Computing*, 5(6), 2001.

[72] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237. ACM Press, 1999.

[73] J.L. Herlocker, J.A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work*, pages 241–250, 2000.

[74] J.L. Herlocker, J.A. Konstan, and J. Riedl. *Information Retrieval, 5*, chapter An Empirical Analysis of Design Choices in Neighborhood-based Collaborative Filtering Algorithms, pages 287–310. Kluwer Academic Publishers, 2002.

[75] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information System*, 22(1):5–53, 2004.

[76] F. Herrera, E. Herrera-Viedma, and L. Martínez. A fusion approach for managing multi-granularity linguistic term sets in decision making. *Fuzzy Sets and Systems*, (114):43–58, 2000.

[77] F. Herrera, E. Herrera-Viedma, L. Martínez, L.G. Pérez, A.G.López, and S. Alonso. *Fuzzy Applications in Industrial Engineering Series: Studies in Fuzziness and Soft Computing , Vol. 201*, chapter A Multi-granular Linguistic Hierarchical Model To Evaluate The Quality Of Web site Services. Springer, 2006.

[78] F. Herrera, E. Herrera-Viedma, and J.L. Verdegay. Direct approach processes in group decision making using linguistic owa operators. *Fuzzy Sets and Systems*, (79):175–190, 1996.

[79] F. Herrera, E. Herrera-Viedma, and J.L. Verdegay. A model of consensus in group decision making under linguistic assessments. *Fuzzy Sets and Systems*, 79:73–87, 1996.

[80] F. Herrera, E. López, C. Mendaña, and M.A. Rodríguez. A linguistic decision model for personnel management solved with a linguistic biobjective genetic algorithm. *Fuzzy Sets and Systems*, 118(1):47–64, 2001.

[81] F. Herrera and L. Martínez. A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on Fuzzy Systems*, 8(6):746–752, 2000.

[82] F. Herrera and L. Martínez. The 2-tuple linguistic computational model. Advantages of its linguistic description, accuracy and consistency. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(Suppl.):33–49, 2001.

[83] F. Herrera and L. Martínez. A model based on linguistic 2-tuples for dealing with multigranular hierarchical linguistic contexts in multi-expert decision-making. *IEEE Transactions on Systems, Man and Cybernetics*, 31(2):227–234, 2001.

[84] F. Herrera, L. Martínez, and P.J. Sánchez. Managing heterogeneous information in group decision making. In *Proceedings Ninth International Conference IPMU 2002*, pages 439–446, Annecy (France), 2002.

[85] F. Herrera, L. Martínez, and P.J. Sánchez. *A Multi-Granular Linguistic Decision Model for Evaluating the Quality of Network Services.*, pages 71–92. In: Intelligent Sensory Evaluation: Methodologies and Applications. Springer, 2004.

[86] F. Herrera, L. Martínez, and P.J. Sánchez. Managing non-homogeneous information in group decision making. *European journal of operational research*, 166:115–132, 2005.

[87] E. Herrera-Viedma, F. Herrera, and F. Chiclana. A consensus model for multiperson decision making with different preference structures. *IEEE Tran-*

sactions on Systems, Man and Cybernetics-Part A: Systems and Humans, 32(3):394–402, 2002.

[88] E. Herrera-Viedma, F. Herrera, F. Chiclana, and M. Luque. Some issues on consistency of fuzzy preference relations. *European Journal of Operational Research*, (154):98–109, 2004.

[89] E. Herrera-Viedma and A.G. López-Herrera. A model of information retrieval system with unbalanced fuzzy linguistic information. *International Journal Of Intelligent Systems*, 22(11):1197–1214, 2007.

[90] E. Herrera-Viedma, F. Mata, L. Martínez, F Chiclana, and L.G. Pérez. Measurements of consensus in multi-granular linguistic group decision making. *Modeling Decisions for Artificial Intelliegence, Proceedings Lecture Notes in Computer Science*, (3131):194–204, 2004.

[91] E. Herrera-Viedma, F. Mata, L. Martínez, and L.G. Pérez. An adaptive module for the consensus reaching process in group decision making problems. In *Proceedings MDAI*, 2005.

[92] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194 – 201, 1995.

[93] D. Hinkle and C.N. Toomey. Clavier: Applying case-based reasoning on to composite part fabrication. In *Proceeding of the Sixth Innovative Application of AI Conference, Seattle, WA, AAAI Press*, pages 55–62, 1994.

[94] H.Nurmi. *Assumptions of Individual Preferences in the Theory of Voting Procedures*, pages 142–155. In: J. Kacprzyk and M. Roubens, Eds., Non conventional Preference Relations in Decision Making. Springer-Verlag, 1988.

[95] U. Hohle. *Mathematics of fuzzy sets*. Kluwer Academic Publishers, 1998.

[96] M. Hsu, M. Bahtt, R. Adolfs, D. Tranel, and C.F. Camarer. Neural systems responding to degrees of uncertainty in human decision-making. *Science*, 310(5754):1680–1683, 2005.

[97] V.-N. Huynh and Y.Ñakamori. A satisfactory-oriented approach to multi-expert decision-making with linguistic assessments. *IEEE Transactions on systems, man, and Cybernetics - Part B: Cybernetics*, 35(2):184–196, 2005.

[98] B.J. Pine II. *Mass Customization*. Harvard Business School Press. Boston, Massachusetts, 1993.

[99] B.J. Pine II. *The Experience Economy*. Harvard Business School Press. Boston, Massachisetts, 1999.

[100] B.J. Pine II, D. Peppers, and M. Rogers. Do you want to keep your customers forever? *Harvard Business School Review*, (2):103–114, 1995.

[101] R. Jain. Tolerance analysis using fuzzy sets. *International Journal Systems Science*, 12(7):1393–1401, 1976.

[102] R. Jin, J.Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 337–344, 2004.

[103] J. Kacprzyk. *The Analysis of Fuzzy Information*, chapter On Some Fuzzy Cores and "Soft" Consensus Measures in Group Decision Making, pages 119–130. In: J. Bezdek. (Ed.). CRC Press, 1987.

[104] J. Kacprzyk, M. Fedrizzi, and H.Nurmi. Group decision making and consensus under fuzzy preferences and fuzzy majority. *Fuzzy Sets and Systems*, 49:21–31, 1992.

[105] D. Kahneman, P. Slovic, and A. Tversky. *Judgement under uncertainty: Heuristics and biases.* Cambridge University Press, Cambridge, 1981.

[106] F. Kalseth. Developing a restaurant recommender system. *Report submitted in partil fulfillment of the requirement for the degree of Bachelor of Computer Science*, 2005.

[107] Y. Klein and G. Langholz. Multicriteria scheduling optimization using fuzzy logic. In *In Proceedings of the IEEE Int Conference on Systems, Man and Cybernetics*, 1998.

[108] G.J. Klir and B. Yuan. *Fuzzy sets an fuzzy logic: Theory and Applications.* Prentice-Hall PTR, New Jersey, 1995.

[109] L.T. Koczy and K. Hirota. Ordering, distance and closeness of fuzzy sets. *Fuzzy Sets and Systems*, 59:291–293, 1993.

[110] J. Kolodner. Reconstructive memory: A computer model. *Cognitive Science*, 7(4):281–328, 1983.

[111] J. Kolodner. *Case-Based Reasoning.* Morgan Kaufmann, 1993.

[112] J.A. Konstan, B. Miller, D. Maltz, J.L. Herlocker, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[113] J.A. Konstan, J. Riedl, A. Borchers, and J.L. Herlocker. Recommender systems: A grouplens perspective. In *AAAI Workshop Recommender Systems 98, Papers from the 1998 Workshop Technical report WS-98-08*, 1998.

[114] B. Krulwich. Lifestyle finder: intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.

[115] S. Kundu. Min-transitivity of fuzzy leftness relationship and its application to decision making. *Fuzzy Sets and Systems*, 86:357–367, 1997.

[116] 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.

[117] R.D. Lawrence, G.S. Almasi, V. Kotlyar, M.S. Viveros, and S.S. Duri. Personalization of supermarket product recommendations. *IBM Research Division, T.J. Watson Research Center, Yorktown Heights, New York. IBM Research Report*, 2000.

[118] J. Lawry. An alternative approach to computing with words. *International Journal of Uncertainty, Fuzziness and Knowledge Based Systems*, 9(Suppl):3–16, 2001.

[119] M. Lebowitz. Memory-based parsing. *Artificial Intelligence*, (21):363–404, 1983.

[120] H. Lee. Group decision making using fuzzy sets theory for evaluating the rate of aggregative risk in software development. *Fuzzy Sets and Systems*, (80):261–271, 1996.

[121] H.S. Lee. On fuzzy preference relation in group decision making. *International Journal of Computer Mathematics*, 82(2):133–140, 2005.

[122] E. Levrat, A. Voisin, S. Bombardier, and J. Bremont. Subjective evaluation of car seat comfort with fuzzy set techniques. *International Journal of Intelligent Systems*, (12):891–913, 1997.

[123] D. Li and J.B. Yang. A multiattribute decision making approach using intuitionistic fuzzy sets. In *Proceedings Eusflat 2003*, pages 183–186, Zitau, 2003.

[124] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 2(108):212–261, 1994.

[125] C.M. Liu, M.J. Wang, and Y.S. Pang. A multiple criteria linguistic decision model (mcldm) for human decision making. *European Journal of Operational Research*, (76):466–485, 1994.

[126] L.Niu, X.-W. Yan, C.-Q. Zhang, and S.-C. Zhang. Product hierarchy-based customer profiles for electronic commerce recommendation. In *Proceedings of 2002 International Conference on Machine Learning and Cybernetics*, volume 2, pages 1075–1080, 2002.

[127] R.D. Luce and P. Suppes. *Handbook of Mathematical Psychology*, chapter Preferences, Utility and Subject Probability, pages 249–410. Wiley, New York, 1965.

[128] H. Mak, I. Koprinska, and J. Poon. Intimate: a web-based movie recommender using text categorization. In *Proceedings of IEEE/WIC International Conference on Web Intelligence, 2003. WI 2003*, pages 602–605, 2003.

[129] M. Marimin, M. Umano, I. Hatono, and H. Tamura. Linguistic labels for expressing fuzzy preference relation in fuzzy group decision making. *IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics*, 28:205–218, 1998.

[130] L. Martínez. Sensory evaluation based on linguistic decision analysis. *International Journal of Aproximated Reasoning*, 44(2):148–164, 2007.

[131] L. Martínez, M. Espinilla, and L.G. Pérez. Linguistic sensory evaluation model in multigranular linguistic contexts. In *10th Joint Conference on Information Sciences*, 2007.

[132] L. Martínez, L.G.Pérez, J. Liu, and J-B Yang. A multi-granular linguistic evaluation model for engineering systems. In *IFSA 2005 World Congress, Fuzzy Logic, Soft Computing and Computational Intelligence Theories and Applications*, 2005.

[133] L. Martínez, L. G. Pérez, P. J. Sánchez, F. Mata, and M. Barranco. *Procesos de Toma de Decisiones, Modelado y Agregación de Preferencia*, chapter Toma de Decisión en Grupo en Contexto Heterogéneos, pages 41–51. Copicentro Granada S.L., 2005.

[134] L. Martínez, L.G. Pérez, and M. Barranco. Models for managing incomplete information in recommeder systems. In *IADIS e-Commerce, Lisbon (Portugal)*, 2004.

[135] L. Martínez, L.G. Pérez, and M. Barranco. A multi-granular linguistic content-based recommendation model. *International Journal of Intelligent Systems*, 22(5), 2007.

[136] L. Martínez, L.G. Pérez, and J. Liu. A fuzzy evaluation process for general purpose web sites. In *IADIS International Conference WWW/Internet 2006*, pages 95–102, 2006.

[137] L. Martínez, L.G. Pérez, J. Liu, J-B Yang, and F. Herrera. *Modern Information Processing: From theory to Application*, chapter A linguistic hierarchical evaluation model for engineering systems. Elsevier, 2006.

[138] L. Martínez, L.G. Pérez, P.J. Sánchez, and B. Montes. A heterogeneous multi-criteria hierarchical evaluation model for web site services. In *10th International Conference on Fuzzy Theory and Technology*, 2005.

[139] L. Martínez, P. Sánchez, B. Montes C. García, F. Mata, and L.G. Pérez. *Un sistema de evaluación basado en técnicas de decisión difusas*, chapter Unidad para la Calidad de Las Universidades Andaluzas. 2005.

[140] F. Mata, L. Martínez, L.G. Pérez, E. Herrera-Viedma, and F. Herrera. A model of consensus reaching process for group decision problems with multigranular linguistic information. In *EUROFUSE ANNIVERSARY WORKSHOP on Fuzzy for Better*, pages 156–165, 2005.

[141] M. McLaughlin and J.L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 2004 Conference on Research and Development in Information Retrieval (SIGIR 2004)*, 2004.

[142] G. Miller. The magical number sever or minus two: Some limits on our capacity of processing information. *Psychological Review*, (63):81–97, 1956.

[143] M.Nowakoswska. Methodological problems of measurement of fuzzy concepts in the social sciences. *Behavioral Science*, 22:107–115, 1977.

[144] F.J. Montero and J. Tejada. Fuzzy preferences in decision-making. *Lecture Notes in Computer Science*, 286:144–150, 1987.

[145] R.J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *In SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[146] R.J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, 2000.

[147] J.N. Morderson and P.S. Nair. *Fuzzy mathematics*. Physica-Verlag, 1998.

[148] M.Pazzani and D.Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, (27):313–331, 1997.

[149] M. O'Connor and J.L. Herlocker. Clustering items for collaborative filtering. In *Presented at the Recommender Systems Workshop at 1999 Conference on Research and Development in Information Retrieval*, 1999.

[150] M. O'Hagan. Aggregating template rule antecedents in real-time expert systems with fuzzy set logic. In *22th Annual IEEE Asilomar Conference on Signals, Systems and Computers*, 1988.

[151] M-N Omri. *Systeme interactif flou dáide a lútilisation de dispositifs techniques: le syteme SIFADE.* PhD thesis, Universite Pierre et Marie Curie, Paris, France, 1994.

[152] C. O'Riordan and J. Griffith. Providing personalised recommendations in a web-based education system. *Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Computer Science*, 2774:245–251, 2003.

[153] S.A. Orlovsky. Decision-making with a fuzzy preference relation. *Fuzzy Sets Systems*, 1:155–167, 1978.

[154] M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61, 1996.

[155] M.J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

[156] W. Pedrycz. *Fuzzy modeling: Paradigms and practice.* Kluwer Academic, 1996.

[157] W. Pedrycz and F. Gomide. *An introduction to fuzzy sets: Analysis and Design (Complex Adaptive Systems).* Bradford Book, 1998.

[158] P. Perny and A. Tsoukias. On the continuous extension of a four Valued logic for preference modelling. pages 302–309, Paris, 1998. IPMU.

[159] P. Perny and J.D. Zucker. Collaborative filtering methods based on fuzzy preference relations. In *In Proceedings of the EUROFUSE-SIC'99 conference*, pages 279–285, 1999.

[160] A.L. Ralescu and D.A. Ralescu. Probability and fuzziness. *Information Science*, 34:85–92, 1984.

[161] M.H. Rasmy, S.M. Lee, W.F. Abd El-Wahed, A.M. Ragab, and M.M. El-Sherbiny. An expert system for multiobjective decision making: Application of fuzzy linguistic preferences and goal programming. *Fuzzy Sets and Systems*, 127:209–220, 2002.

[162] M.M. Recker, A. Walker, and K. Lawless. What do you recommend? implementation and analyses of collaborative information filtering of web resources for education. *Instructional Science*, 31(4-5):299–316, 2003.

[163] F. Reichheld. Loyalty-based management. *Harvard Business School Review*, (2):64–73, 1993.

[164] F. Reichheld and W.E. Sasser. Zero defections: Quality comes to services. *Harvard Business School Review*, (5):105–111, 1990.

[165] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[166] P. Resnick and H.R. Varian. Recommender systems. *Association for Computing Machinery. Communications of the ACM.*, 40(3):56, Mar 1997.

[167] E. Rich. User modeling via stereotypes. *Cognitive Science*, (3):329–354, 1979.

[168] C.K. Riesbeck and R.C. Schank. *Inside Case-Based Reasoning*. Lawrence Erlbaum, 1989.

[169] M. Rifqi, V. Berger, and B. Bouchon-Meunier. Discrimination power of measures of comparison. *Fuzzy Sets and Systems*, 110(2):189–196, 2000.

[170] J.J. Rocchio. *The SMART retrieval system. Experiments in automatic document processing*, chapter Relevance feedback in information retrieval, pages 313–323. Prentice-Hall, 1971.

[171] J-P Rossazza. *Utilisation de hierarchie de classes floues pour la representation des connaissances imprecises et sujettes a exception: le systeme*. PhD thesis, Universite Paul Sabatier, Toulouse, France, 1990.

[172] M. Roubens and Ph. Vincke. *Preference modelling*. Springer-Verlag, 1985.

[173] E.H. Ruspini. On the semantics of fuzzy logic. *International Journal of Approximate Reasoning*, (5):45–88, 1991.

[174] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[175] E. Sanchez. Inverse of a fuzzy relations. applications to possibility distributions and medical diagnosis. *Fuzzy Sets and Systems*, 2(1):75–86, 1979.

[176] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *In Proceedings of the 10th International World Wide Web Conference (WWW10)*, pages 285–295, 2001.

[177] B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–167, 2000.

[178] B.M. Sarwar, J.A. Konstan, A. Borchers, J.L. Herlocker, B. Miller, and J. Riedl. Using filtering agents to improve prediction quality in the grou-

plens research collaborative filtering system. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, 1998.

[179] J.B. Schafer, J.A. Konstan, and J. Riedl. Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166, 1999.

[180] J.B. Schafer, J.A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, (5):115–153, 2001.

[181] R. Schank. *Dynamic Memory: A Theory of Learning in Computers and People*. New York: Cambridge University Press, 1982.

[182] I. Schwab, A. Kobsa, and I. Koychev. Learning user interests through positive examples using content analysis and collaborative filtering. *Internal Memo, GMD, St. Augustin, Germany*, 2001.

[183] I. Schwab, W. Pohl, and I. Koychev. Learning to recommend from positive evidence. In *Proceedings of Intelligent User Interfaces*, pages 241–247, 2000.

[184] F. Seo and M. Sakawa. Fuzzy multiattribute utility analysis for collective choice. *IEEE Transactions on Systems, Man and Cybernetics*, 15:45–53, 1985.

[185] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.

[186] B. Smyth and P. Cotter. A personalised tv listings service for the digital tv age. *Journal of Knowledge-Based Systems*, 13(2-3):53–59, 2000.

[187] T. Tanino. Fuzzy preference orderings in group decision making. *Fuzzy Sets and Systems*, 12:117–131, 1984.

[188] T. Tanino. *Non-Conventional Preference Relations in Decision Making*, chapter Fuzzy Preference Relations in Group Decision Making, pages 54–71. Springer-Verlag, 1988.

[189] T. Tanino. *On Group Decision Making Under Fuzzy Preferences*, pages 172–185. in: J. Kacprzyk and M. Fedrizzi, Eds., Multiperson Decision Making Using Fuzzy Sets and Possibility Theory. Kluwer Academic Publishers, 1990.

[190] M. Taschuk. A hybrid knowledge-based/content-based recommender system in the bluejay genome browser, 2007.

[191] J.F. Le Téno and B. Mareschal. An interval version of PROMETHEE for the comparison of building products'design with ill-defined data on environmental quality. *European Journal of Operational Research*, 109:522–529, 1998.

[192] L. Terveen and W. Hill. *Human-Computer Interaction in the New Millennium*, chapter Human-Computer Collaboration in Recommender Systems. Addison-Wesley, 2002.

[193] M. Tong and P.P. Bonissone. A linguistic approach to decision making with fuzzy sets. *IEEE Transactions on Systems, Man and Cybernetics*, (10):716–723, 1980.

[194] M. Tong and P.P. Bonissone. Linguistic solution to fuzzy decision problems. *Studies in the Management Science*, 20:323–334, 1984.

[195] V. Torra. Negation function based semantics for ordered linguistic labels. *International Journal of Intelligent Systems*, 11:975–988, 1996.

[196] V. Torra. Linguistic aggregation in non-unified domains. In *EUROFUSE-SIC 99*, pages 188–193, Budapest, 1999.

[197] V. Torra. Aggregation of linguistic labels when semantics is based on antonyms. *International Journal of Intelligent Systems*, 16:513–524, 2001.

[198] B. Towle and C. Quinn. Knowledge-based recommender systems using explicit user models. In *Knowledge-Based Electronic Markets, Paper from the AAAI Workshop, AAAI Technical Report WS-00-04*, pages 74–77, 2000.

[199] T. Tran and R. Cohen. Hybrid recommender systems for electronic commerce. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-00) Workshop on Knowledge-Based Electronic Markets*, pages 78–84, 2000.

[200] I. Truck and H. Akdag. A tool for aggregation with words. *Information Sciences, Special Issue: Linguistic Decision Making: Tools and Applications*, in press.

[201] J.D. Ullman. *Principles of database and Knowledge-base systems, Vol. I.* Computer Science Press, Inc., 1988.

[202] T. Walsh. Representing and reasoning with preferences. *AI Magazine*, Winter:59–69, 2007.

[203] J.-H. Wang and J. Hao. A new version of 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on Fuzzy Systems*, 14(3):435–445, 2006.

[204] Z.Q. Wang and B.Q. Feng. Collaborative filtering algorithm based on mutual information. *Advanced Web Technologies and Applications, Lecture Notes in Computer Science*, 3007:405–415, 2004.

[205] J. Wolf, C. Aggarwal, K-L. Wu, and P. Yu. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA.*, pages 201–212, 1999.

[206] Z.S. Xu. A method based on linguistic aggregation operators for group decision making with linguistic preference relations. *Information Science*, 166:19–30, 2004.

[207] Z.S. Xu. Uncertain linguistic aggregation operators based approach to multiple attribute group decision making under uncertain linguistic environment. *Information Sciences*, 168:171–184, 2004.

[208] Z.S. Xu. Incomplete linguistic preference relations and their fusion. *Information Fusion*, 7(3):331–337, 2006.

[209] Z.S. Xu. *Fuzzy Sets and Their Extensions: Representation, Aggregation and Models*, chapter Linguistic Aggregation Operators: An Overview, pages 163–181. Springer, 2008.

[210] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183 – 190, 1988.

[211] R.R. Yager. *Fuzzy Logic: State of the Art*, chapter Fuzzy Screening Systems, pages 251–261. Kluwer Academic Publisher, 1993.

[212] R.R. Yager. An approach to ordinal decision making. *International Journal of Approximate Reasoning*, (12):237–261, 1995.

[213] R.R. Yager. Induced aggregation operators. *Fuzzy Sets and Systems*, 137(1):59 – 69, 2003.

[214] R.R. Yager, L.S. Goldstein, and E. Mendels. Fuzmar: An approach to aggregating market research data based on fuzzy reasoning. *Fuzzy Sets and Systems*, (68):1–11, 1994.

[215] Y. Yang. Expert network: effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–186, Zitau, 1994.

[216] A. Yazici and R. George. *Fuzzy Database Modeling*. Phisyca-Verlag, 1999.

[217] C.Y. Yue, S.B Yao, and P. Zhang. Rough approximation of a preference relation for stochastic multi-attribute decision problems. *Lecture Notes in Artificial Intelligence*, 3613:1242–1245, 2005.

[218] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, 8-9(I, II, III):199–249, 301–357, 43–80, 1975.

[219] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[220] L.A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers and Mathematics with Applications*, 9(1):149–184, 1983.

[221] L.A. Zadeh. Fuzzy logic = computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2):103–111, 1996.

[222] C. Zeng, C.-X. Xing, L.-Z. Zhou, and X.-H. Zheng. Similarity measure and instance selection for collaborative filtering. *International Journal of Electronic Commerce*, 8(4):115–129, 2004.

[223] Q. Zhang, J.C.H. Chen, and P.P. Chong. Decision consolidation: Criteria weight determination using multiple preference formats. *Decision Support Systems*, 38(2):247–258, 2004.

[224] H.J. Zimmermann. *Fuzzy sets: Theory and its applications*. Kluwer Academic, 1996.

[225] R. Zwick, E. Carlstein, and D.V. Budescu. Measures of similarity among fuzzy concepts: A comparative analysis. *International Journal of Approximate Reasoning*, 1(2):221–242, 1987.


ESPAÑA
OPATB2
CZECH AIRLINES CSA
NOT TRANSPORTATION
OK NOT ENDORSABLE OKZ/CAY
MAD OK PRG OK MAD
EUR
EUR
EUR
EUR
9548313465
This portion of the ticket
should be retained as
evidence of your journey.
parte del billete
conservarse como
ante de su viaje.