



Universidad Nacional del Sur

TESIS DE DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN

*Integración de técnicas cualitativas y cuantitativas
en los sistemas de recomendación*

Cristian Emanuel Briguez

BAHÍA BLANCA

ARGENTINA

2019

Prefacio

Esta Tesis se presenta como parte de los requisitos para optar al grado Académico de Doctorado en Ciencias de la Computación, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otra. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el ámbito del Departamento de Ciencias e Ingeniería de la Computación durante el período comprendido entre el 1 de Abril de 2011 y el 18 de Junio de 2019, bajo la dirección de la Dra. Marcela Capobianco, Profesora Adjunta del Departamento de Ciencias e Ingeniería de la Computación y de la Dra. Ana Gabriela Maguitman, Profesora Asociada del Departamento de Ciencias e Ingeniería de la Computación.

.....
Cristian Emanuel Briguez

ceb@cs.uns.edu.ar

Departamento de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur

Bahía Blanca, 18 de Junio de 2019



UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el .../.../..., mereciendo la calificación de(.....)

Agradecimientos

Quiero agradecer a todos aquellos que han colaborado de alguna u otra manera a que haya logrado finalizar este camino, que ha resultado más largo y desafiante de lo esperado, no cabe duda que sin el aporte de todos ustedes esto no hubiera sido posible. En primer lugar, quiero agradecer a mis directoras, Ana y Marcela, por toda la paciencia y dedicación que han mostrado durante estos años, sobre todo cuando empecé con la escritura de esta tesis y las cosas no me estaban saliendo según lo planeado, siempre hicieron todo lo posible para que pudiera salir adelante. También agradezco su confianza por acompañarme una vez más en mi intento por concluir esta tesis. Les voy a estar siempre agradecido, no solo por haberme permitido hacer el doctorado, sino también porque me abrieron las puertas para muchísimas cosas.

Agradezco a mis compañeros de la salita de becarios: Ariel, Maxi, Santi, Carlitos, Juli, Gotti, Noni, Kcho, Anita, Jime, Diego, Lucho, Mauro, Ana y Martín, por hacerme vivir el trabajo en forma amena, generando diariamente un ambiente laboral placentero, por brindarme su apoyo y ayudarme incondicionalmente en todos estos años de trabajo. Además, por todo lo extra laboral, gracias por todas las juntadas, asados, fútbol, charlas y todos esos momentos de alegría vividos a lo largo de estos años. Particularmente quiero agradecer a Kcho y a Fer que fueron los primeros en ayudarme a iniciar este camino y por supuesto a Gotti, por todas las charlas que hemos tenido a lo largo de estos años, donde sus consejos siempre fueron de gran ayuda.

Quiero agradecer a todo el personal docente y no docente del Departamento de Ciencias e Ingeniería de la Computación (DCIC), por darme la posibilidad de trabajar junto a ellos todos estos años. Agradezco especialmente a los docentes que fueron profesores durante mi carrera de grado y a todos aquellos con los que tuve la oportunidad de trabajar como ayudante de cátedra, en especial a Laura, Seba, Clara y Pablo.

Especialmente quiero agradecer a Aldu, quien ha sido una de las grandes motivaciones para alcanzar este objetivo. Ella ha sido mi compañera durante esta última etapa, con la que he recorrido varias salas de lectura y bibliotecas en búsqueda de inspiración, con la que pude compartir el día a día de la escritura de esta tesis, haciendo que sea más fácil superar las frustraciones y permitiendo disfrutar más de los éxitos.

Uno de los principales agradecimientos es para mi familia, le quiero dedicar este trabajo a mis padres, Carlos y Graciela, y a mis hermanas, Gise, Nata y Dai. En primer lugar por darme la oportunidad de vivir una infancia, adolescencia y adultez felices. En segundo lugar por toda su dedicación en nuestra educación, por haberme enseñado a no bajar nunca los brazos y a seguir intentando siempre con honestidad y humildad. Gracias por el apoyo incondicional y por estar SIEMPRE!

A mis compañeros de trabajo de la AMBB también les quiero agradecer, porque siempre fueron muy considerados conmigo y más de una vez me dieron una mano para ayudarme a llegar hasta acá.

También les quiero agradecer a todos mis amigos, a pesar que últimamente los he dejado un poco de lado, son una parte fundamental de mi vida. Gracias por todos los momentos compartidos!

Por último quiero agradecer al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) y a la Universidad Nacional del Sur (UNS) por confiar en mí y brindarme los medios necesarios para poder llevar a cabo las investigaciones necesarias para el desarrollo de este trabajo doctoral. Esta tesis no hubiera sido posible sin el apoyo de estas instituciones.

¡Muchas gracias a todos!

Harry

Resumen

Los sistemas de recomendación se han hecho cada vez más frecuentes en los últimos años, ya que ayudan a los usuarios a acceder a elementos relevantes del vasto universo de posibilidades disponibles en estos días. La mayoría de las investigaciones existentes en el área se basan puramente en aspectos cuantitativos, tales como índices de popularidad o medidas de similitud entre elementos o usuarios.

En esta tesis se desarrollaron distintos modelos para la implementación de sistemas de recomendación, que incorporan aspectos que difieren de los más clásicos de un recomendador. Uno de los modelos desarrollados incorpora la noción de confiabilidad, mientras que otro fomenta la transparencia de la recomendación para los usuarios y la flexibilidad para instanciar las preferencias de recomendación en un sistema de recomendación. Los modelos propuestos incorporan el uso de argumentos a favor o en contra de las recomendaciones para determinar si una sugerencia debe ser presentada o no a un usuario. Para lograr esto, se adopta la Programación Lógica Rebatible (DeLP, del inglés, Defeasible Logic Programming) como el formalismo subyacente para modelar hechos y reglas sobre el dominio de recomendación y para computar el proceso de argumentación.

Este enfoque tiene varias ventajas sobre otras técnicas de recomendación existentes. En particular, las recomendaciones se pueden refinar fácilmente en cualquier momento agregando nuevas reglas pulidas. También, queda en evidencia la capacidad de inferencia de los modelos propuestos. Lo más importante es que las explicaciones que apoyan cada recomendación pueden proporcionarse de una manera que sea fácil de entender para el usuario, por medio de los argumentos calculados.

Abstract

Recommender systems have become prevalent in recent years as they help users to access relevant items from the vast universe of possibilities available these days. Most existing research in this area is based purely on quantitative aspects such as indices of popularity or measures of similarity between items or users.

In this thesis, different models were developed for the implementation of recommendation systems, which incorporate aspects that differ from those classically found in a recommender. One of the models developed incorporate the notion of trust, while another promotes the transparency of the recommendation for users and the flexibility to instantiate recommendation preferences in a recommendation system. The proposed models incorporate the use of arguments in favor or against recommendations to determine if a suggestion should be presented or not to a user. In order to accomplish this, Defeasible Logic Programming (DeLP) is adopted as the underlying formalism to model facts and rules about the recommendation domain and to compute the argumentation process.

This approach has a number of advantages over other existing recommendation techniques. In particular, recommendations can be easily refined at any time by adding new polished rules. Also, the inference capability of the proposed models is evident. Most importantly, explanations supporting each recommendation can be provided in a way that is easily understood by the user, by means of the computed arguments.

Índice general

1. Introducción	1
1.1. Motivaciones	5
1.2. Contribuciones	8
1.3. Publicaciones	9
1.4. Organización de la tesis	11
2. Sistemas de recomendación	13
2.1. Función de un sistema de recomendación	16
2.2. Datos y fuentes de conocimiento	20
2.3. Técnicas de recomendación	24
2.3.1. Basado en contenido	28
2.3.2. Filtrado colaborativo	34
2.3.3. Sistemas híbridos	42
2.4. Explicaciones	44
2.5. Evaluación de los resultados de un RS	48
2.5.1. Calidad de las predicciones	49
2.5.2. Calidad de un conjunto de recomendaciones	50
2.5.3. Otras métricas	51
2.6. Resumen	52

3. Sistemas argumentativos	55
3.1. Una caracterización de los sistemas argumentativos	56
3.2. Formalización de sistemas argumentativos	61
3.2.1. Programación en lógica rebatible (DeLP)	63
3.3. Argumentación con soporte de base de datos (DBI-DeLP)	90
3.4. Resumen	92
4. Confiabilidad	95
4.1. Definiendo confiabilidad	95
4.2. Confiabilidad y sistemas de recomendación	100
4.3. Confiabilidad y argumentación	104
4.4. Confiabilidad y noticias en la web	107
4.5. Resumen	109
5. Modelo de confiabilidad para la recomendación de noticias	111
5.1. Caracterización del modelo	112
5.1.1. Usando DeLP para modelar confiabilidad en noticias	115
5.2. Aptitud de DeLP para modelar confiabilidad	117
5.2.1. Algunos ejemplos seleccionados	117
5.2.2. Propiedades de la confiabilidad y su caracterización usando DeLP	119
5.2.3. Propiedades de DeLP y su aptitud para modelar confiabilidad	124
5.3. Validación por simulación	128
5.4. Conclusiones	131

6. Sistemas de recomendación mixtos basados en argumentación	133
6.1. RS basados en argumentación	135
6.1.1. RS de películas basado en argumentación	137
6.1.2. RS de música basado en argumentación	142
6.1.3. Decidiendo qué es importante: aspectos relevantes en la recomen- dación final	148
6.1.4. Expandingo aspectos: incorporando reglas	149
6.2. El rol de argumentación en recomendación	152
6.2.1. Recomendaciones basadas en argumentación	153
6.2.2. Ofreciendo explicaciones para las recomendaciones	155
6.3. Evaluación empírica de los enfoques	157
6.3.1. Definición del conjunto de datos	158
6.3.2. Configuración experimental	160
6.3.3. Métricas de rendimiento	161
6.3.4. Resultados	164
6.3.5. Análisis de los resultados	167
6.3.6. Comparación con algunos métodos del estado del arte usados en la recomendación de películas	171
6.4. Trabajo relacionado	178
6.5. Conclusiones	182
7. Conclusiones y trabajo a futuro	185
A. Simulaciones	193
B. Glosario	207
B.1. Acrónimos	207
B.2. Terminología	209

Capítulo 1

Introducción

Hoy en día, Internet está disponible prácticamente en todas partes debido a los avances de la tecnología. Hay teléfonos celulares, computadoras y relojes inteligentes que tienen la capacidad de conectarse a Internet. La gente usa estos dispositivos y proporciona más y más información a través de Internet. Gran parte de esta nueva información diaria es generada a partir de la interacción con las redes sociales. Analizando algunas estadísticas publicadas por *Brandwatch*¹, se observa cuál es el volumen de estas interacciones en las principales redes sociales de hoy en día. Por ejemplo en *Instagram*, en un día promedio, se comparten 80 millones de imágenes, en *YouTube* se suben 400 horas de video por minuto, en *Facebook* se publican 350 millones de fotos al día y en *Twitter*, también en el mismo período, se generan 500 millones de tweets. La información no solo se obtiene de computadoras personales y teléfonos móviles, sino también de cajas registradoras que las personas usan cuando realizan compras. Además, los centros de investigación tienen enormes bases de datos con fines científicos. Los observatorios astronómicos almacenan imágenes de galaxias y registran cada segundo del universo. Por lo tanto, existe una gran sobrecarga de información que está creciendo día a día. Como resultado, las bases de datos han crecido de gigabytes a terabytes y hasta petabytes. Es por esto que existen grandes conjuntos de datos que están disponibles en todas partes. Un gran conjunto de datos puede ser beneficioso si se supervisa y gestiona adecuadamente.

Todo esto conduce a que sea necesaria la existencia de mecanismos que permitan extraer información relevante de los conjuntos de datos, porque los datos por sí mismos

¹<https://www.brandwatch.com/>

no proporcionan mucha información. Estos se deben procesar, organizar, estructurar o presentar en un contexto dado para hacerlos realmente útiles.

Este tipo de mecanismos se ha aplicado en distintos dominios. Por ejemplo, los sitios de comercio electrónico utilizan uno de estos mecanismos, conocido como sistemas de recomendación, para ofrecer sugerencias efectivas de productos a sus clientes. Compañías como *Amazon* o *Netflix* brindan propuestas a sus clientes, también, a partir de sistemas de recomendación. De esta manera, están haciendo recomendaciones de productos en los que los clientes podrían estar interesados, además de los demás productos en su sitio web.

Los sistemas de recomendación no solo son populares en el sector del comercio electrónico, sino también en el sector de medios digitales. Los sitios web que ofrecen noticias, música, películas o fotografías usan sistemas de recomendación. Hay muchas razones para usar este tipo de sistemas. Un diario online, le puede proponer al usuario una noticia para leer, una radio por Internet puede elegir la siguiente canción para reproducir, o un sitio web de películas puede sugerir una película que a los usuarios les puede gustar. Hay demasiado para leer, escuchar o mirar. Es por esto, que las personas necesitan de estos mecanismos, para evitar tener que filtrar y seleccionar solo el contenido o la información que sea relevante o interesante para ellos directamente del conjunto de datos.

En esta tesis se aborda el estudio y desarrollo de mecanismos o herramientas conocidos como sistemas de recomendación. En particular, se propone una manera de modelar la confiabilidad de los usuarios del sistema basada en argumentación. Además se proponen sistemas de recomendación que tienen la particularidad de ser mixtos, combinando tanto aspectos cuantitativos como cualitativos al momento de realizar la recomendación. Como parte de la contribución, se han demostrado teóricamente ciertas propiedades que cumple la noción de confiabilidad. Por otra parte, otro aspecto novedoso es que el uso de la argumentación para generar recomendaciones no solo aporta poder predictivo al sistema sino también poder explicativo.

Lo expuesto en esta tesis incorpora características y elementos que aún no habían sido considerados en forma conjunta por otras propuestas desarrolladas hasta el momento. En consecuencia, los resultados obtenidos en esta tesis brindan una contribución importante a los desarrollos en la comunidad, tanto de los sistemas de recomendación como de los de argumentación, significando esto un aporte dentro de las áreas de Minería de Datos e Inteligencia Artificial en las Ciencias de la Computación.

En lo que resta de este capítulo se presentará brevemente el contexto junto con las motivaciones que guiaron el desarrollo de esta tesis. Seguido de esto se describirán las principales contribuciones y resultados obtenidos. Finalmente, se indicará la organización del resto de la tesis, detallando brevemente el contenido de los capítulos siguientes.

Sistemas de Recomendación

Los sistemas de recomendación son mecanismos de apoyo que ayudan a los usuarios en su proceso de toma de decisiones al interactuar con volúmenes de información grandes o complejos. La mayoría de los sistemas de recomendación están destinados a ayudar a los usuarios para hacer frente al problema de la sobrecarga de información, facilitando el acceso a los elementos pertinentes [Mae94, JZFF10, RRSK11, MS17]. Con elementos podemos referir tanto a productos como a servicios y los usuarios particulares pueden representar tanto a individuos como a empresas. Los recomendadores intentan generar un modelo del usuario o de la tarea del usuario y aplicar diversas heurísticas para anticipar qué tipo de información puede ser útil. De esta manera, tienen la capacidad de predecir si un elemento en particular puede ser de interés para un usuario.

Los sistemas de recomendación convencionales se basan en los índices de popularidad o en medidas de similitud entre usuarios o contenido, calculados sobre la base de métodos procedentes de las ciencias sociales, la recuperación de información o el aprendizaje automático. En general, dos estrategias de recomendación han llegado a predominar. Los recomendadores basados en contenido (CB, del inglés *Content-Based*) se basan en el rico contenido de las descripciones de los elementos (productos o servicios) que se recomiendan. Por ejemplo, un recomendador de películas basado en contenido generalmente se basará en información como género, actores, director, etc., y combinará esto con las preferencias aprendidas del usuario para seleccionar un conjunto de recomendaciones de películas prometedoras para el usuario. Obviamente, esto impone una carga significativa de ingeniería de conocimiento a los diseñadores de recomendadores basados en contenido, ya que el conocimiento del dominio requerido puede no estar fácilmente disponible o ser fácil de mantener. Como alternativa, la estrategia de recomendación de filtrado colaborativo (CF, del inglés *Collaborative Filtering*) proporciona una posible solución. Esta opción está motivada por la observación de que en realidad a menudo buscamos recomendaciones de nuestros amigos. El conocimiento de los elementos no es requerido. En cambio, el fil-

trado colaborativo (a veces llamado filtrado social) se basa en la disponibilidad de perfiles de usuario que capturan los historiales de calificaciones anteriores de los usuarios. Las recomendaciones se generan para un usuario en particular teniendo en cuenta los historiales de calificaciones de un subconjunto de usuarios adecuado. Generalmente este subconjunto de usuarios es conformado por usuarios que comparten calificaciones similares o altamente correlacionadas con el usuario en cuestión.

Argumentación Rebatible

Argumentación es una forma de razonamiento en la cual, para una determinada afirmación (*conclusión*), se presta atención explícita a las justificaciones presentadas y a la resolución de los posibles conflictos entre ellas. En este tipo de razonamiento, una afirmación es aceptada o rechazada en función del análisis de los argumentos a su favor y en su contra. La forma en que los argumentos y las justificaciones para una afirmación son considerados permite definir un mecanismo de razonamiento automático, en el cual puede haber información contradictoria, incompleta e incierta [PV02, RS09]. En las últimas décadas, argumentación ha evolucionado como un atractivo paradigma para conceptualizar el razonamiento de sentido común [CML00, BD07, BH08, AyC14].

Existen diferentes aproximaciones para modelar argumentación, tales como sistemas argumentativos abstractos [Dun95, Pra10] o sistemas argumentativos basados en reglas [PS97, GS04]. Por otra parte, este tipo de razonamiento es particularmente atractivo para la toma de decisiones, y dentro del área de Inteligencia Artificial existe especial interés en utilizar este tipo de formalismos para modelar información tentativa y potencialmente contradictoria. Concretamente, los sistemas argumentativos han sido utilizados en diversos dominios y aplicaciones como el razonamiento legal [PS02, Ver03b, dCPLM⁺17], sistemas para la toma de decisiones y negociación [APM00, RRJ⁺03, BH09, AV12, FTG⁺17, BTG19], sistemas multi-agente [PSJ98, AMP02, RA06, RGS07, GGS10, CI15, GTGS18], planificación [SGC04, PG18], e-government [ABM05, CMGE16] y el procesamiento de lenguaje natural [CM04b, WvEH16].

Confiabilidad

Estudios anteriores han señalado que no existen criterios generales para emitir una valoración objetiva de la confiabilidad [Ben11], las definiciones de confiabilidad se dividen en varias categorías, y por lo general una definición sólida para esto, en muchos casos, puede ser bastante elusiva.

Aunque la confiabilidad es de suma importancia en los sistemas de recomendación, los mecanismos actuales de propagación de confiabilidad y preservación de la misma son poco comprendidos. Esto se debe en parte al hecho de que la confiabilidad es una noción compleja, que típicamente es dependiente del contexto, subjetiva, dinámica y no siempre transitiva o simétrica [ELV13, WSWW17, DRP17, AWBT18, MWZW18]. La noción de confiabilidad ha estado ganando una creciente atención en varias comunidades de investigación y, por supuesto, hay muchos puntos de vista diferentes sobre cómo medir y usar la confiabilidad [MC01, Wat05, Li07, EK09, SW14, MDM15].

1.1. Motivaciones

Como se ha mencionado, los sistemas de recomendación son una herramienta que acompaña a los usuarios en el proceso de toma de decisiones, mientras desligan al usuario de interactuar con una cantidad, masiva o compleja, de información. Es por esto, que los sistemas de recomendación son de suma utilidad para facilitar el acceso a los elementos de interés del usuario, evitándole tener que lidiar con el problema de la sobrecarga de información [Mae94, JZFF10, RRSK11, MS17].

Una observación importante es que la interacción hombre-máquina en los métodos de recomendación existentes es particularmente rígida. Estos servicios no ofrecen mecanismos para reformular fácilmente los criterios bajo los cuales se basará la recomendación. Esto significa que una vez que se adopta una técnica de recomendación en particular y se le asigna un conjunto de valores de parámetros (que normalmente se establecen por la configuración del sistema de recomendación), el usuario no tiene la posibilidad de introducir de forma natural nuevas preferencias al sistema para considerar o indicar cómo combinar diferentes hechos y reglas para generar recomendaciones. Las decisiones sobre las preferencias del usuario se basan principalmente en heurísticas, que dependen de la clasificación de las decisiones anteriores del usuario o de la recopilación de información

de otros usuarios con intereses similares. En otras palabras, los sistemas de recomendación existentes no proporcionan un mecanismo de interacción para expresar e inferir de una manera natural y metódica las relaciones de preferencias de los usuarios en entornos complejos.

Con frecuencia buscamos sugerencias de las personas en quienes confiamos para decidir el mejor lugar para adquirir algún servicio o la mejor fuente para obtener información sobre un tema determinado. Sugerecias de personas en quienes confiamos también nos pueden ayudar a decidir en quién más confiar. La Web ofrece nuevas oportunidades para crear servicios de recomendación basado en la confiabilidad.

En particular, los sistemas de gestión de noticias en la Web pueden tomar ventaja de la gran comunidad de lectores de noticias para clasificar dichas noticias, determinar la reputación de una fuente de información o propagar la confiabilidad entre los usuarios. Esto puede ayudar a decidir qué noticias son más interesantes o confiables para un determinado usuario, proporcionando servicios más personalizados.

La dinámica de la confiabilidad de las noticias se ha estudiado principalmente a través de enfoques cuantitativos (por ejemplo, [NSKA06]). Sin embargo, una perspectiva puramente cuantitativa de la confiabilidad de las noticias tiene varias limitaciones. En particular, los enfoques cuantitativos hacen que sea difícil proporcionar a los lectores una justificación de por qué ciertas noticias deberían ser confiables, o podrían ser incapaces de hacer frente a las nociones sutiles, como la desconfianza o la reparación de la confianza.

Un enfoque más atractivo consistiría en combinar los criterios cuantitativos y cualitativos para filtrar y rankear las noticias. En este sentido, los métodos cuantitativos pueden ayudar a determinar si el tema de las noticias es relevante para el interés del usuario, mientras que los criterios cualitativos podrían ayudar a decidir si la noticia proviene de una fuente confiable. En general, un enfoque cualitativo es más natural para hacer frente a algunos desafíos que presenta el modelado de la noción de confiabilidad, como ser la dependencia del contexto, la subjetividad, la asimetría y la característica de que la confiabilidad no siempre es transitiva.

Otra debilidad central en la mayoría de los métodos actuales de recomendación es la falta de transparencia. La mayoría de los sistemas de recomendación en línea actúan como cajas negras, no ofrecen al usuario idea alguna de la lógica del sistema o la justificación de las recomendaciones. Esto es debido al hecho de que los métodos cuantitativos adoptados

por la mayoría de los sistemas de recomendación existentes no tienen un modelo subyacente claro que sea fácilmente comprensible a un nivel de usuario final. Esto hace que sea difícil de proporcionar a los usuarios una explicación clara de los factores y procedimientos que llevaron al sistema a realizar ciertas recomendaciones. Como se ha demostrado en estudios anteriores (por ejemplo, [SS02, TM07]) los usuarios prefieren recomendaciones si pueden comprender las razones por las que se presentan estas recomendaciones. Ciertamente, una recomendación es más convincente si el usuario conoce y está de acuerdo con las razones que apoyan el por qué de la recomendación que se presentó que si la misma se desprende de una caja negra. Además, al ofrecer una justificación el sistema está proporcionando información útil a sus usuarios acerca de otros elementos del sistema de los cuales podrían no tener conocimiento.

Otro problema que enfrentan los sistemas de recomendación es que los criterios de preferencia de los usuarios en general, implican un conocimiento incompleto y potencialmente contradictorio sobre el dominio. Esto es debido al hecho de que las preferencias de los usuarios son dinámicas y típicamente cambian a medida que el tiempo transcurre o cuando nuevo material esta disponible para el análisis. Debido a que los enfoques cuantitativos adoptados por la mayoría de los sistemas de recomendación no están equipados con un mecanismo de revisión de las conclusiones anteriores, la naturaleza cambiante de las preferencias de los usuarios es tratada pobremente. La adopción de un enfoque más cualitativo mediante el aumento de recomendaciones con la inferencia de la lógica clásica no resultaría suficiente para resolver los problemas antes mencionados, ya que a menudo va a llevar a conclusiones contradictorias, que son problemáticas si no se tratan adecuadamente.

A pesar de su importancia, el uso de mecanismos de argumentación integrados a los sistemas de recomendación no ha recibido mucha atención por parte de la comunidad de los sistemas de recomendación. Sin embargo, en la última década se han hecho varios avances en los campos de la argumentación como en el de los sistemas de recomendación, aunque usualmente estudiados de manera independiente. El problema de fortalecer a la recomendación con la argumentación ha sido previamente tratado en [CMG09]. En ese trabajo, el foco se encuentra en una caracterización general de *sistemas de recomendación basados en argumentación*, descritos como herramientas de apoyo a los usuarios donde las recomendaciones se basan en argumentos.

A la luz de estos problemas, se propone modelar criterios de preferencia de los usuarios utilizando DeLP (por su nombre en inglés *Defeasible Logic Programming*) [GS04], marco para la argumentación rebatible [RS09, SL92] basado en la programación lógica que puede tratar eficazmente con información incompleta y contradictoria. El uso de DeLP permite integrar el razonamiento dialéctico en el proceso de recomendación, que proporciona una base razonada de los elementos sugeridos al usuario. Además de ofrecer un medio natural para codificar datos sobre el dominio de recomendación, DeLP puede hacer frente eficazmente a la naturaleza de los patrones de razonamiento rebatible que normalmente surgen en la recomendación. Además, una característica interesante de DeLP, que ha demostrado ser útil en entornos de recomendación, es que las conclusiones obtenidas a través de este formalismo se pueden explicar fácilmente por el proceso de razonamiento argumentativo.

1.2. Contribuciones

Las principales contribuciones de esta tesis pueden sintetizarse de la siguiente manera:

Confiabilidad basada en argumentación

Se presenta un modelo que permite representar la confiabilidad de un usuario con respecto a distintas entidades. Se realiza la investigación teórica identificando un conjunto de propiedades deseables de la confiabilidad y proporcionando una prueba formal de dos propiedades importantes del modelo. La primera propiedad hace referencia al hecho de que las premisas que establecen el estado de credibilidad del usuario con respecto a un elemento (producto o servicio) no pueden revocarse (*inclusión*) y la segunda asegura que el sistema no puede concluir que una entidad (un elemento u otro usuario), sea simultáneamente confiable y no confiable por el mismo usuario (*consistencia*). Un ingrediente clave del modelo propuesto es la propagación de confiabilidad basada en mecanismos de inferencia. En particular, se utiliza un intérprete de programación lógica rebatible (DeLP) para gestionar la confiabilidad y la desconfianza interpersonales. La confiabilidad se modela por medio de una serie de postulados básicos, que se pueden extender o relajar fácilmente. Los postulados se traducen en reglas rebatibles, que permiten inferir conclusiones tentativas sobre la credibilidad de un usuario sobre un determinado elemento.

Integración de aspectos cualitativos y cuantitativos en sistemas de recomendación

Se proponen sistemas de recomendación mixtos, es decir, sistemas de recomendación donde se combinan tanto aspectos cuantitativos como cualitativos al momento de realizar una recomendación. Los enfoques de recomendación basados en criterios puramente cuantitativos a menudo no proporcionan a los usuarios las razones detrás de las recomendaciones, lo que afecta la confianza de los usuarios en los resultados. En enfoques que consideran aspectos cualitativos, como el que propone esta tesis, las explicaciones naturalmente complementan las recomendaciones. Por lo tanto, el usuario recibirá una recomendación y un motivo que la respalde. Esto tiene una doble ventaja: el usuario tendrá más confianza en el resultado presentado, y el usuario puede darle al sistema comentarios explícitos que pueden ayudar a guiar el proceso de recomendación.

Se presenta un marco completo para lograr recomendaciones basadas en el razonamiento rebatible. Una característica particularmente interesante es que el marco ofrece un mecanismo para establecer criterios de preferencia entre argumentos, lo que permite ajustar fácilmente el comportamiento de los sistemas hacia diferentes aspectos. Para lograr definir sistemas de recomendación mixtos se modelan los criterios cuantitativos y cualitativos utilizando reglas DeLP. Este enfoque mixto permite generar recomendaciones razonadas basadas en conjuntos de datos masivos del mundo real.

1.3. Publicaciones

A continuación se indican los artículos publicados como resultado de los trabajos llevados a cabo durante la realización de esta tesis. En particular, para cada uno de estos trabajos, se detalla brevemente su contribución y se indican los capítulos de esta tesis con los que se vinculan sus resultados.

- En el trabajo “*System Architecture for Trust-Based News Recommenders on the Web*” [BSCM11], publicado en *XVII Congreso Argentino de Ciencias de la Computación (CACIC 2011)* se mostró cómo la noción de confiabilidad puede ser modelada cualitativamente e incorporada en la arquitectura de un sistema de recomendación de noticias. En el sistema propuesto, se permite a los usuarios expresar declaraciones

de confiabilidad explícita para las distintas entidades del sistema. La confiabilidad es modelada y propagada utilizando un proceso dialéctico respaldado por un intérprete de programación en lógica rebatible. Los resultados de este trabajo se hallan reflejados en el desarrollo del Capítulo 5.

- En el trabajo “*A Theoretical Framework for Trust-based News Recommender Systems and its Implementation using Defeasible Argumentation*” [BCM13], publicado en *International Journal on Artificial Intelligence Tools* se presentó un marco teórico que se puede aplicar para implementar un sistema de recomendación basado en confiabilidad para las noticias de la Web. Se realizó el análisis de un conjunto de propiedades deseadas de la confiabilidad, ya que sabemos que ésta puede ser subjetiva, dinámica, rebatible, dependiente del contexto y no siempre simétrica o transitiva. Además se mostró que estas propiedades se pueden modelar de forma natural utilizando el marco propuesto. Se demostraron dos teoremas fundamentales que caracterizan el comportamiento de cualquier sistema de recomendación basado en el modelo propuesto de propagación de confiabilidad. El primer teorema establece que el sistema no puede rebatir una premisa de confiabilidad realizada explícitamente. Más formalmente, establece que el conjunto de premisas de confiabilidad se incluye en el conjunto de conclusiones de confiabilidad. El segundo teorema establece la consistencia del sistema, lo que excluye la posibilidad de concluir que un usuario confíe y desconfíe simultáneamente de la misma entidad del sistema. Los resultados de este trabajo se hallan reflejados en el desarrollo del Capítulo 5.
- En el trabajo “*Towards an Argument-based Music Recommender System*” [BBD⁺12] publicado en *Fourth International Conference on Computational Models of Argument (COMMA 2012)* se describe un sistema de recomendación basado en argumentación para el dominio de la música. Se presenta un marco completo para lograr recomendaciones basadas en el razonamiento rebatible. Una característica particularmente interesante es que el marco propuesto ofrece un mecanismo para establecer criterios de preferencia entre argumentos, lo que permite ajustar fácilmente el comportamiento de los sistemas hacia diferentes aspectos. Este mecanismo permite pasar de enfoques basados en contenido a enfoques de filtrado colaborativo con bastante facilidad, simplemente cambiando las prioridades entre las reglas. Además, el uso de estructuras de razonamiento coherentes (argumentos) hace que dar explicaciones a

las recomendaciones sea un proceso sencillo. Los resultados de este trabajo se hallan reflejados en el desarrollo del Capítulo 6.

- En el trabajo “*Argument-based Mixed Recommenders and their Application to Movie Suggestion*” [BBD⁺14], publicado en *Expert Systems With Applications* se propuso un enfoque basado en argumentación con la capacidad de mejorar las tecnologías de recomendación actuales. La argumentación se puede usar para realizar un análisis cualitativo de los usuarios y los posibles elementos a recomendar. Esto permite ir un paso más allá con respecto al enfoque clásico, dando lugar a sistemas mixtos de recomendación, es decir, sistemas que basan sus recomendaciones en aspectos tanto cualitativos como cuantitativos del dominio. En particular, este enfoque mixto facilita la consideración de varios aspectos antes de presentar una recomendación final al usuario. Estos aspectos se pueden aplicar para modelar algunas características útiles que son difíciles de tener en cuenta en enfoques puramente cuantitativos, tales como, la naturaleza rebatible de las preferencias de los usuarios en entornos complejos. Una característica útil del formalismo es que permite mucha flexibilidad tanto para incorporar nuevos aspectos a tener en cuenta al momento de realizar la recomendación, como para dar preferencia sobre un tipo de enfoque en particular. Nuevamente, gracias a la utilización de argumentos para realizar las recomendaciones, se pueden obtener explicaciones completas sobre las razones detrás de las recomendaciones. A través de la explotación de la estructura interna coherente de argumentos se puede generar una explicación final que se presenta a los usuarios como una sentencia en lenguaje natural. Los resultados de este trabajo se hallan reflejados en el desarrollo del Capítulo 6.

1.4. Organización de la tesis

El resto de la tesis se encuentra ordenada de la siguiente manera:

- En el Capítulo 2 se describen las principales ideas y conceptos de los sistemas de recomendación en general, incluyendo desde la definición de su función principal, los componentes fundamentales de un sistema de recomendación y las técnicas más populares de recomendación haciendo hincapié en sus fortalezas y debilidades. Finalmente se describen las principales formas de evaluación de un sistema de recomendación.

- En el Capítulo 3 se introducen los conceptos que caracterizan a los sistemas argumentativos, a través de una estructura conceptual. Adicionalmente se presenta el sistema argumentativo que se utilizará para el desarrollo de esta tesis: el sistema argumentativo de la programación en lógica rebatible, también conocido como DeLP. Además se hace mención al formalismo DBI-DeLP, el cual facilita la integración de información almacenada en bases de datos con un sistema basado en argumentación rebatible.
- En el Capítulo 4 se realiza una revisión de diferentes consideraciones para la definición de la noción de confiabilidad. Se exploran distintas alternativas del estado del arte de sistemas de recomendación y argumentativos considerando la noción de confiabilidad.
- En el Capítulo 5 se hace un análisis de distintas propiedades deseables de la confiabilidad. Además, se propone y analiza un modelo de representación y propagación de confiabilidad. Este modelo se utiliza para la implementación de un sistema de recomendación. El sistema es evaluado mediante la realización de una serie de simulaciones.
- En el Capítulo 6 se presenta una novedosa perspectiva sobre los sistemas de recomendación, que combina un método cuantitativo básico con un enfoque cualitativo, lo que da como resultado una familia de sistemas de recomendación de carácter mixto. Se analiza el aporte de argumentación en este enfoque, ofreciendo explicaciones que apoyan cada recomendación. Por último, se realizan pruebas empíricas para poder comparar el enfoque propuesto con otros exponentes del estado del arte.
- En el Capítulo 7 se presentan conclusiones y el trabajo a futuro.
- En el Apéndice A se presentan los resultados de las ejecuciones de las distintas simulaciones realizadas para validar el modelo propuesto en el Capítulo 5.
- En el Apéndice B se muestra el glosario de esta tesis.

Capítulo 2

Sistemas de recomendación

Los sistemas de recomendación (RS, del inglés *Recommender Systems*) son herramientas y técnicas de software que brindan sugerencias sobre los elementos que pueden ser de utilidad para un usuario [RV97, Bur07, MR09, JZFF10, RRSK11, MS17]. Las sugerencias se relacionan con varios procesos de toma de decisiones, como qué música escuchar, qué noticias leer o qué artículos comprar. El término general que se usa para denotar lo que el sistema recomienda a los usuarios se denomina “*elemento*”. Un RS normalmente se enfoca en un tipo específico de elemento (por ejemplo, películas o noticias) y, en consecuencia se personalizan su diseño, su interfaz gráfica de usuario y la técnica de recomendación principal utilizada para generar las recomendaciones, para así proporcionar sugerencias útiles y efectivas para ese tipo específico de elemento. Los RS se dirigen principalmente a personas que carecen de experiencia o competencia personal suficiente para evaluar un número potencialmente abrumador de elementos alternativos que, por ejemplo, un sitio web puede ofrecer [RV97]. Un ejemplo de ello es un sistema de recomendación de películas que ayuda a los usuarios a seleccionar una película para mirar. En la popular plataforma de contenido *Netflix*¹, se emplea un RS para personalizar el acceso al contenido en línea para cada cliente [GUH16]. Dado que las recomendaciones suelen ser personalizadas, diferentes usuarios o grupos de usuarios reciben diversas sugerencias. Otro ejemplo es el RS que presenta *Amazon.com*² en su tienda en línea para recomendar libros a sus clientes. Este también ofrece recomendaciones no personalizadas. Estas son mucho más simples de generar y normalmente aparecen en revistas o periódicos. Los ejemplos típicos incluyen

¹<https://www.netflix.com/>

²<https://www.amazon.com/>

las diez mejores selecciones de libros, álbumes de música, etc. Mientras que pueden ser útiles y efectivas en ciertas situaciones, estos tipos de recomendaciones no personalizadas no suelen abordarse en la investigación de RS.

En su forma más simple, las recomendaciones personalizadas se ofrecen como listas clasificadas de elementos. Al realizar esta clasificación, los RS intentan predecir cuáles son los productos o servicios más adecuados, según las preferencias y restricciones del usuario. Para completar dicha tarea computacional, los RS recopilan las preferencias de los usuarios, que se expresan explícitamente, por ejemplo, como calificaciones de productos, o se deducen al interpretar las acciones de los usuarios. Por ejemplo, un RS puede considerar la navegación a una página de un producto en particular como un signo implícito de preferencia por los elementos que se muestran en esa página. El desarrollo de RS comenzó a partir de una observación bastante simple: los individuos a menudo confían en las recomendaciones proporcionadas por otros para tomar decisiones rutinarias y diarias [MR09, MM09]. Por ejemplo, es común confiar en lo que sus compañeros recomiendan al seleccionar un libro para leer; los empleadores cuentan con cartas de recomendación en sus decisiones de reclutamiento; y al seleccionar una película para ver, los individuos tienden a leer y confiar en las reseñas de películas que un crítico de cine ha escrito y que aparecen en el periódico que habitualmente leen. Al tratar de imitar este comportamiento, los primeros RS aplicaron algoritmos para aprovechar las recomendaciones producidas por una comunidad de usuarios para entregar recomendaciones a un usuario activo, es decir, un usuario que busca sugerencias. Las recomendaciones fueron para elementos que les gustaron a usuarios similares (aquellos con gustos similares). Este enfoque se denomina filtrado colaborativo y su razonamiento es que si el usuario activo estuvo de acuerdo en el pasado con algunos usuarios, entonces las otras recomendaciones provenientes de estos usuarios similares también deberían ser relevantes y de interés para el usuario activo. A medida que los sitios web de comercio electrónico comenzaron a desarrollarse, surgió una necesidad apremiante de proporcionar recomendaciones derivadas del filtrado de toda la gama de alternativas disponibles. Los usuarios encontraban muy difícil llegar a las opciones más adecuadas de la inmensa variedad de elementos (productos y/o servicios) que estos sitios web estaban ofreciendo. El crecimiento explosivo y la variedad de información disponible en la Web y la rápida introducción de nuevos servicios de comercio electrónico (productos de compra, comparación de productos, subastas, etc.) con frecuencia abrumaron a los usuarios, lo que los llevó a tomar decisiones deficientes. La disponibilidad de opciones, en lugar de producir un beneficio, comenzó a disminuir el bienestar de los usuarios. Se entendió que

si bien tener opciones es bueno, no siempre tener más opciones es mejor.

Los RS han demostrado en los últimos años ser un medio valioso para hacer frente al problema de la sobrecarga de información. Fundamentalmente, un RS aborda este fenómeno apuntando a un usuario hacia elementos nuevos, aún no experimentados que pueden ser relevantes para la tarea actual de los usuarios. A solicitud del usuario, que puede expresarse, según el enfoque de recomendación, según el contexto y la necesidad del usuario, los RS generan recomendaciones utilizando diversos tipos de conocimiento y datos sobre los usuarios, los elementos disponibles y las transacciones anteriores almacenadas en bases de datos personalizadas. El usuario puede navegar por las recomendaciones, puede aceptarlas o no y puede proporcionar, inmediatamente o en una etapa siguiente, una retroalimentación implícita o explícita. Todas estas acciones y comentarios de los usuarios se pueden almacenar en la base de datos de recomendaciones y se pueden usar para generar nuevas recomendaciones en las siguientes interacciones entre el usuario y el sistema.

Como se señaló anteriormente, el estudio de los sistemas de recomendación es relativamente nuevo en comparación con la investigación en otras herramientas y técnicas clásicas de sistemas de información (por ejemplo, bases de datos o motores de búsqueda). Los sistemas de recomendación surgieron como un área de investigación independiente a mediados de la década del '90 [GNOT92, RIS⁺94, HSRF95, SM95, KMM⁺97]. En los últimos años, el interés en los sistemas de recomendación ha aumentado dramáticamente, existen hechos claros que así lo demuestran. Uno de estos hechos, es que los sistemas de recomendación desempeñan un papel importante en sitios de Internet altamente calificados como *Amazon.com*, *Netflix*, *YouTube*³, *Yahoo*⁴, *TripAdvisor*⁵, *Spotify*⁶, *Last.fm*⁷ e *IMDb*⁸. Además, muchas compañías de medios ahora están desarrollando e implementando RS como parte de los servicios que brindan a sus suscriptores. Por ejemplo, *Netflix*, el servicio de alquiler de películas en línea, otorgó un premio de un millón de dólares al equipo que primero logró mejorar sustancialmente el rendimiento de su sistema de recomendación [KBV09]. Otro de estos hechos es que hay conferencias dedicadas y talleres relacionados con el campo, por ejemplo, ACM Recommender Systems (*RecSys*), establecido en 2007 y actualmente siendo

³<https://www.youtube.com/>

⁴<https://www.yahoo.com/>

⁵<https://www.tripadvisor.com/>

⁶<https://www.spotify.com/>

⁷<https://www.last.fm/>

⁸<https://www.imdb.com/>

el evento anual principal en investigación y aplicación de tecnología de recomendación. Además, las sesiones dedicadas a los RS se incluyen frecuentemente en las conferencias más tradicionales en el área de bases de datos, sistemas de información y sistemas adaptativos. Entre estas conferencias, cabe mencionar a ACM SIGIR Special Interest Group on Information Retrieval (*SIGIR*), Adaptation and Personalization (*UMAP*), y al ACM's Special Interest Group on Management Of Data (*SIGMOD*).

En el resto de este capítulo, se describen las ideas y conceptos básicos de los sistemas de recomendación en general, abarcando desde su función principal, los componentes fundamentales de un RS, también se mencionan las principales técnicas de recomendación haciendo hincapié en los métodos más populares y sus fortalezas y debilidades. Finalmente se describen las principales formas de evaluación de un RS.

2.1. Función de un sistema de recomendación

Previamente, se definió a los RS como herramientas de software y técnicas que brindan sugerencias a los usuarios para los elementos que el usuario desee utilizar. Ahora es posible refinar esta definición para ilustrar un rango de posibles roles que un RS puede desempeñar. En primer lugar, se debe distinguir entre la función que desempeña el RS en nombre del proveedor de servicios y la del usuario del RS. Es común que para un intermediario de viajes (por ejemplo, *Expedia.com*⁹) o una organización de gestión de destinos (por ejemplo, *Visitfinland.com*¹⁰) se utilice un sistema de recomendación de viajes para aumentar su facturación (*Expedia*), es decir, vender más habitaciones de hotel o para aumentar el número de turistas al destino [Ric02, SA09, YH13]. Mientras que, las principales motivaciones del usuario para acceder a los dos sistemas es encontrar un hotel adecuado y eventos/atracciones interesantes cuando visite un destino. De hecho, hay varias razones por las cuales los proveedores de servicios pueden querer explotar esta tecnología:

Aumentar el número de artículos vendidos. Esta es probablemente la función más importante para un RS comercial, es decir, poder vender un conjunto adicional de artículos en comparación con los que se venden generalmente sin ningún tipo de recomendación [GRD⁺15]. Este objetivo se logra porque es probable que los elementos recomendados se adapten a las necesidades y deseos del usuario. Las aplicaciones no comerciales

⁹<https://www.expedia.com/>

¹⁰<https://www.visitfinland.com/>

tienen objetivos similares, incluso si el usuario no tiene ningún costo asociado con la selección de un artículo. Por ejemplo, una red de contenido apunta a aumentar el número de noticias leídas en su sitio. En general, podemos decir que, desde el punto de vista del proveedor de servicios, el objetivo principal para introducir un RS es aumentar la tasa de conversión, es decir, la cantidad de usuarios que aceptan la recomendación y consumen un elemento, en comparación con la cantidad de simples visitantes que solo navegan a través de la información.

Vender artículos más diversos. Otra función importante de un RS es permitir al usuario seleccionar elementos que puedan ser difíciles de encontrar sin una recomendación precisa. Por ejemplo, en un RS de alquiler de películas, el proveedor del servicio está interesado en ofrecer todos los títulos disponible en su catálogo, no solo los más populares. Esto podría ser difícil sin un RS, ya que el proveedor del servicio correría el riesgo de que las películas recomendadas no sean adecuadas para el gusto de un usuario en particular. Por lo tanto, un RS sugiere o anuncia películas impopulares solamente a los usuarios potencialmente interesados en las mismas.

Aumentar la satisfacción del usuario. Un RS bien diseñado también puede mejorar la experiencia del usuario con el sitio o la aplicación. El usuario encontrará las recomendaciones interesantes, relevantes y, con una interacción usuario-computadora adecuadamente diseñada, también disfrutará usando el sistema. La combinación de recomendaciones efectivas, es decir, precisas y una interfaz fácil de usar aumentará la evaluación subjetiva del sistema por parte del usuario. Esto, a su vez, aumentará el uso del sistema y la probabilidad de que se acepten las recomendaciones.

Aumentar la fidelidad del usuario. Un usuario debe ser leal a un sitio web que, cuando lo visita, reconoce al antiguo cliente y lo trata como un visitante valioso. Esta es una característica normal de un RS, ya que muchos RS computan recomendaciones, aprovechando la información adquirida por las interacciones anteriores del usuario, por ejemplo, sus calificaciones de elementos. En consecuencia, cuanto más tiempo interactúa el usuario con el sitio, más refinado se vuelve su modelo de usuario, es decir, la representación en el sistema de las preferencias del usuario, y esto permite personalizar, de manera más efectiva, la salida del recomendador para que coincida con las preferencias del usuario.

Entender mejor lo que quiere el usuario. Otra función importante de un RS, que se puede aprovechar para muchas otras aplicaciones, es la descripción de las preferencias del usuario, ya sea recopilada explícitamente o predicha por el sistema. El proveedor

de servicios puede entonces decidir reutilizar este conocimiento para una serie de otros objetivos, como mejorar la gestión del stock o la producción del objeto a recomendar. Por ejemplo, en el dominio de viajes, las organizaciones de gestión de destinos pueden decidir anunciar una región específica a nuevos sectores de clientes o anunciar un tipo particular de mensaje promocional derivado del análisis de los datos recopilados por el RS.

Anteriormente se mencionaron algunas motivaciones importantes en cuanto a por qué los proveedores de servicios electrónicos presentan RS. Pero los usuarios también pueden querer un RS, si lo apoyara efectivamente en sus tareas u objetivos. En consecuencia, un RS debe equilibrar las necesidades de estas dos partes intervinientes y ofrecer un servicio que sea valioso para ambos. En [HKTR04], un documento que se ha convertido en una referencia clásica en este campo, se definen las tareas más populares en las que un RS puede ayudar al usuario. Algunas pueden considerarse como las tareas principales o convencionales que normalmente se asocian con un RS, es decir, ofrecer sugerencias de elementos que pueden ser útiles para un usuario. Otras pueden ser consideradas como maneras más oportunistas de explotar un RS. De hecho, esta diferenciación de tareas es muy similar a lo que sucede con un motor de búsqueda. Su función principal es localizar documentos que sean relevantes para la necesidad de información del usuario, pero también se puede usar para verificar la importancia de una página web (mirando la posición de la página en la lista de resultados de una consulta) o para descubrir los diversos usos de una palabra en una colección de documentos. Entre las tareas en las que un RS puede ayudar al usuario se destacan:

Encontrar algunos elementos buenos. Recomendar a un usuario algunos elementos como una lista clasificada junto con predicciones de cuánto le gustaría al usuario (por ejemplo, en una escala de uno a cinco estrellas). Esta es la tarea de recomendación principal que abordan muchos sistemas comerciales. Algunos sistemas no muestran la calificación prevista.

Encontrar todos los elementos buenos. Recomendar todos los artículos que puedan satisfacer alguna necesidad particular del usuario. En tales casos, es insuficiente encontrar solo algunos buenos elementos. Esto es especialmente cierto cuando el número de elementos es relativamente pequeño o cuando el RS ayuda a realizar una tarea crítica, como en aplicaciones médicas o financieras. En estas situaciones, además del beneficio derivado de examinar cuidadosamente todas las posibilidades, el usuario también puede

beneficiarse de la clasificación de estos elementos o de las explicaciones adicionales que genera el RS.

Anotación en contexto. Dado un contexto existente, por ejemplo, una lista de elementos, resaltar algunos de ellos dependiendo de las preferencias a largo plazo del usuario. Por ejemplo, un sistema de recomendación de TV podría comentar qué programas de TV de los que se muestran en la guía de programas son dignos de ver.

Recomendar una secuencia. En lugar de centrarse en la generación de una sola recomendación, la idea es recomendar una secuencia de elementos que sea agradable en general. Los ejemplos típicos incluyen recomendar una serie de televisión, un libro sobre RS después de haber recomendado un libro sobre minería de datos o una compilación de canciones [HC01, SHB05, QCJ18].

Recomendar un paquete. Sugerir un grupo de elementos que sea adecuado agrupar. Por ejemplo, un plan de viaje puede estar compuesto por varias atracciones, destinos y servicios de alojamiento ubicados en un área delimitada. Desde el punto de vista del usuario, estas diversas alternativas pueden considerarse y seleccionarse como un único destino de viaje [YXYG15, BL16].

Solo navegación. En esta tarea, el usuario navega por el catálogo sin ninguna intención inminente de comprar un elemento. La tarea del recomendador es ayudar al usuario a navegar por los elementos que tienen más probabilidades de caer dentro del alcance de los intereses del usuario para esa sesión de navegación específica.

Encontrar recomendadores creíbles. Algunos usuarios no confían en los sistemas de recomendación, por lo que juegan con ellos para ver qué tan buenos son al hacer recomendaciones. Por lo tanto, algunos sistemas también pueden ofrecer funciones específicas para permitir que los usuarios prueben su comportamiento además de los que solo se requieren para obtener recomendaciones.

Mejorar el perfil. Esto se relaciona con la capacidad del usuario para proporcionar información (de entrada) al sistema de recomendación sobre lo que le gusta y lo que no le gusta. Esta es una tarea fundamental que es estrictamente necesaria para proporcionar recomendaciones personalizadas. Si el sistema no tiene un conocimiento específico sobre el usuario activo, entonces solo puede proporcionarle las mismas recomendaciones que se entregarían a un usuario “genérico”.

Expresarse. Algunos usuarios pueden no preocuparse en absoluto por las recomendaciones. Más bien, lo que es importante para ellos es que se les permita contribuir con sus calificaciones y expresar sus opiniones y creencias. La satisfacción del usuario por esa actividad puede actuar como una motivación para mantener al usuario firmemente en el sistema.

Ayudar a otros. Algunos usuarios están felices de contribuir con información, por ejemplo, con la evaluación de elementos (calificaciones), porque creen que la comunidad se beneficia de su contribución. Esta podría ser una motivación importante para ingresar información en un sistema de recomendación que no se usa de manera rutinaria. Por ejemplo, un RS de autos, un usuario que ya compró su nuevo auto es consciente de que es más probable que la calificación ingresada en el sistema sea útil para otros usuarios que para la próxima vez que compre un auto.

Influir en otros. En los RS basados en la web, hay usuarios cuyo objetivo principal es influir explícitamente en otros usuarios para que compren productos específicos. De hecho, también hay algunos usuarios malintencionados que pueden usar el sistema solo para promocionar o penalizar ciertos elementos.

Como lo indican estos diferentes puntos, el rol de un RS dentro de un sistema de información puede ser bastante diverso. Esta diversidad requiere la explotación de una variedad de distintas fuentes y técnicas de conocimiento, y en las siguientes secciones se analizan los datos que administra un RS y las técnicas principales utilizadas para identificar las recomendaciones correctas.

2.2. Datos y fuentes de conocimiento

Los RS recopilan activamente varios tipos de datos para elaborar sus recomendaciones. Los datos son principalmente sobre los elementos que se sugieren y los usuarios que recibirán estas recomendaciones. Pero, dado que los datos y las fuentes de conocimiento disponibles para los sistemas de recomendación pueden ser muy diversos, dependiendo de la técnica de recomendación utilizada, estos datos pueden ser explotados o no. En general, existen técnicas de recomendación que son deficientes en conocimiento, es decir, utilizan datos muy simples y básicos, como las calificaciones/evaluaciones de los usuarios para los elementos. Otras técnicas son mucho más dependientes del conocimiento, por ejemplo,

mediante el uso de descripciones ontológicas de los usuarios o los elementos, o las restricciones, o las relaciones sociales o las actividades de los usuarios. En cualquier caso, como una clasificación general, los datos utilizados por los RS se refieren a tres tipos de objetos: elementos, usuarios y transacciones, es decir, relaciones entre usuarios y elementos.

- **Elemento.** Los elementos son los objetos que se recomiendan. Los elementos pueden caracterizarse por su complejidad y su valor o utilidad. El valor de un elemento puede ser positivo si éste es útil para el usuario, o negativo si el elemento no es apropiado y el usuario tomó una decisión incorrecta al seleccionarlo. Cuando un usuario está adquiriendo un elemento, siempre se incurrirá en un costo, que incluye el costo cognitivo de la búsqueda del elemento y el costo monetario real, si fuera el caso, que finalmente se pagó por el elemento.

Por ejemplo, el diseñador de un RS de noticias debe tener en cuenta la complejidad de una noticia, es decir, su estructura, la representación textual y la importancia dependiente del tiempo de cualquier noticia. Pero, al mismo tiempo, el diseñador del RS debe entender que incluso si el usuario no paga por leer noticias, siempre hay un costo cognitivo asociado a la búsqueda y lectura de noticias. Si un elemento seleccionado es relevante para el usuario, este costo está dominado por el beneficio de haber adquirido una información útil, mientras que si el elemento no es relevante, el valor neto de ese elemento para el usuario y su recomendación es negativo. En otros dominios, por ejemplo, automóviles o inversiones financieras, el verdadero costo monetario de los elementos se convierte en un aspecto importante a considerar cuando se selecciona el enfoque de recomendación más apropiado. Los RS, de acuerdo con su tecnología central, pueden usar una variedad de propiedades y características de los elementos. Por ejemplo, en un sistema de recomendación de películas, el género (como comedia, suspenso, etc.), así como el director y los actores pueden usarse para describir una película y aprender cómo la utilidad de un elemento depende de sus características. Los elementos se pueden representar utilizando diversos enfoques de información y representación, por ejemplo, de manera minimalista como un código de identificación único, o en una forma más rica, como un conjunto de atributos, pero incluso como un concepto en una representación ontológica del dominio.

- **Usuario.** Los usuarios de un RS, como se mencionó anteriormente, pueden tener objetivos y características muy diversas. Con el fin de personalizar las recomendaciones y la interacción usuario-computadora, los RS explotan una gama de información sobre los usuarios. Esta información se puede estructurar de varias maneras y nuevamente la selección de qué información modelar depende de la técnica de recomendación. Por ejemplo, en el filtrado colaborativo, los usuarios se modelan como una lista simple que contiene las calificaciones proporcionadas por el usuario para algunos elementos. En un RS demográfico, se utilizan atributos sociodemográficos como la edad, el género, la profesión y la educación. Se dice que los datos de usuario constituyen el modelo de usuario [Fis01, WPB01]. El modelo de usuario perfila al usuario, es decir, codifica sus preferencias y necesidades. Se han utilizado varios enfoques de modelado de usuarios y, en cierto sentido, un RS se puede ver como una herramienta que genera recomendaciones al construir y explotar modelos de usuarios [BKR08, BKR09]. Dado que no es posible la personalización sin un modelo de usuario conveniente, a menos que la recomendación no sea personalizada, como en la selección de los 10 principales, el modelo de usuario siempre jugará un papel central. Por ejemplo, al considerar nuevamente un enfoque de filtrado colaborativo, al usuario se le perfila directamente por sus calificaciones a los elementos o, usando estas calificaciones, el sistema deriva un vector de valores de factores, donde los usuarios difieren en cómo cada factor es ponderado en su modelo. Los usuarios también pueden ser descritos por sus patrones de comportamiento, por ejemplo, patrones de navegación del sitio (en un sistema de recomendación basado en la web) [TKG07, LCK10], o patrones de búsqueda de viajes (en un sistema de recomendación de viajes) [MR09]. Además, los datos de usuario pueden incluir relaciones entre usuarios, como la relación de confiabilidad que puede existir entre usuarios. Un RS puede utilizar esta información para recomendar elementos a usuarios que fueron preferidos por usuarios similares o de confianza [HC07, GZT14, SKLB17].
- **Transacción.** En términos generales, una transacción hace referencia a una interacción registrada entre un usuario y el RS. Las transacciones son similares a un registro que almacena información importante generada durante la interacción usuario-computadora y que es útil para el algoritmo de generación de recomendaciones que utiliza el sistema. Por ejemplo, un registro de transacciones puede contener una referencia al elemento seleccionado por el usuario y una descripción del con-

texto (por ejemplo, objetivo del usuario/consulta del usuario) para una recomendación en particular. Si está disponible, esa transacción también puede incluir una retroalimentación explícita que el usuario ha proporcionado, como la calificación del elemento seleccionado. De hecho, las calificaciones son la forma más popular de transacciones que recopila un RS. Estas calificaciones pueden ser recogidas explícita o implícitamente. En la recolección explícita de calificaciones, se le pide al usuario que brinde su opinión sobre un elemento en una escala de calificación. Según [SFHS07], las calificaciones pueden tomar una variedad de formas:

- *Calificaciones numéricas*, como las estrellas de 1 a 5 provistas en el RS de libros asociado con *Amazon.com*.
- *Calificaciones ordinales*, como “totalmente de acuerdo, de acuerdo, neutral, en desacuerdo, totalmente en desacuerdo” donde se le pide al usuario que seleccione el término que mejor indique su opinión con respecto a un elemento (generalmente a través de un cuestionario).
- *Calificaciones binarias*, que modelan las elecciones en las que simplemente se le pide al usuario que decida si un determinado elemento es bueno o malo.
- *Calificaciones unarias*, que pueden indicar que un usuario ha observado o comprado un elemento, o lo ha calificado de manera positiva. En tales casos, la ausencia de una calificación indica que no se tiene información que relacione al usuario con el elemento (tal vez compró el elemento en otro lugar).

Otra forma de evaluación que tiene el usuario consiste en etiquetas asociadas por el mismo usuario a los elementos que presenta el sistema. Por ejemplo, en el RS de *MovieLens*¹¹, las etiquetas representan cómo se sienten los usuarios de *MovieLens* acerca de una película, por ejemplo: “*demasiado larga*”, “*basada en un libro*” o, haciendo referencia al tema de la película, “*futuro lejano*”.

En las transacciones que recopilan calificaciones implícitas, el sistema pretende inferir la opinión de los usuarios en función de las acciones del usuario. Por ejemplo, si un usuario ingresa cualquier palabra clave relacionada con su objetivo en *Amazon.com*, se le proporcionará una larga lista de libros. Tras observar dicha lista, el usuario puede hacer clic en un determinado libro para recibir información adicional.

¹¹<https://movielens.org/>

En este punto, el sistema puede inferir que el usuario está algo interesado en ese libro.

En los sistemas conversacionales, es decir, los sistemas que soportan un proceso interactivo, el modelo de transacción es más refinado. En estos sistemas, las solicitudes de los usuarios se alternan con las acciones del sistema. Es decir, el usuario puede solicitar una recomendación y el sistema puede producir una lista de sugerencias. Pero también puede solicitar preferencias del usuario adicionales para proporcionarle mejores resultados.

2.3. Técnicas de recomendación

Para implementar su función principal, identificando los elementos útiles para el usuario, un RS debe predecir que vale la pena recomendar un determinado elemento. Para hacer esto, el sistema debe poder predecir la utilidad de algunos de ellos, o al menos comparar la utilidad de algunos elementos, y luego decidir qué elementos recomendar basándose en esta comparación. El paso de predicción puede no ser explícito en el algoritmo de recomendación, pero aún se puede tener un modelo unificado para describir el papel general de un RS. Para ilustrar el paso de predicción de un RS, hay que considerar, por ejemplo, un algoritmo de recomendación simple, no personalizado, que recomienda solo las canciones más populares. La razón para usar este enfoque es que, en ausencia de información más precisa sobre las preferencias del usuario, una canción popular, es decir, algo que le gusta a muchos usuarios (utilidad alta), probablemente también le gustará a un usuario genérico, o al menos, más que otra canción seleccionada al azar. Por lo tanto, se predice que la utilidad de estas canciones populares será razonablemente alta para este usuario genérico. Esta vista del cálculo de la recomendación como la predicción de la utilidad de un elemento para un usuario se ha sugerido en [AT05]. Este valor de la utilidad de un elemento para un usuario, normalmente en un sistema basado en filtrado colaborativo, se obtiene considerando las calificaciones de los usuarios para los elementos. Entonces, la tarea fundamental de un RS basado en filtrado colaborativo es predecir el valor de la utilidad sobre pares de usuarios y elementos. Habiendo calculado esta predicción para el usuario activo en un conjunto de elementos, el sistema recomendará los elementos con la mayor utilidad predicha. La cantidad de elementos a recomendar, por lo general, suele ser un número pequeño, mucho más pequeño que la cardinalidad del conjunto de elementos,

y posiblemente, menor que la cantidad de elementos en los que se calcula la predicción de utilidad para el usuario, es decir, estos RS “filtran” los elementos que se recomiendan a los usuarios.

Como se mencionó anteriormente, algunos sistemas de recomendación no estiman completamente la utilidad antes de hacer una recomendación, pero pueden aplicar algunas heurísticas para suponer que un elemento es de utilidad para un usuario. Esto es típico, por ejemplo, en los sistemas basados en contenido. Estas predicciones de utilidad se calculan con algoritmos específicos y utilizan diversos tipos de información sobre los usuarios y los elementos. Por ejemplo, el sistema puede asumir que la función de utilidad es booleana y, por lo tanto, solo determinará si un elemento es o no útil para el usuario. Por consiguiente, si se asume que existe información disponible (posiblemente ninguna) sobre el usuario que solicita la recomendación, la información sobre los elementos y otros usuarios que recibieron recomendaciones, el sistema aprovechará este conocimiento con un algoritmo adecuado para generar varias predicciones de utilidad y, por lo tanto, recomendaciones [Bur07].

También es importante tener en cuenta que a veces se observa que la utilidad de un elemento para el usuario depende de otras variables, generalmente llamadas “contextuales” [ASST05, CMVGRG⁺15, Twa16, CPGPSM17, BCJ⁺18]. Por ejemplo, la utilidad de un elemento para un usuario puede verse influida por el conocimiento del dominio por parte del usuario (por ejemplo, los usuarios expertos frente a los usuarios principiantes con respecto al uso de una cámara fotográfica), o puede depender del momento en que se solicita la recomendación. O también, el usuario puede estar más interesado en los elementos (por ejemplo, un restaurante) más cercanos a su ubicación actual. En consecuencia, las recomendaciones deben adaptarse a estos detalles adicionales específicos y, como resultado, cada vez es más difícil estimar correctamente cuáles son las recomendaciones correctas. En general existen varios tipos diferentes de sistemas de recomendación que varían en términos del dominio en cuestión, la información utilizada, pero especialmente en relación con el algoritmo de recomendación, es decir, cómo se realiza la predicción de la utilidad de una recomendación, como se menciona al principio de esta sección. A continuación se proporciona una descripción general de los diferentes tipos de RS. Ésta taxonomía, proporcionada por [Bur07], es una forma clásica de distinguir entre sistemas de recomendación y referirse a ellos.

Basado en contenido: el sistema aprende a recomendar elementos similares a los que le gustaron al usuario en el pasado [Lan95, VMVS00, SA06, Jan08, GZHS16, BHM17]. La similitud de los elementos se calcula en función de las características asociadas de los elementos comparados. Por ejemplo, si un usuario ha calificado positivamente una película que pertenece al género de comedia, entonces el sistema puede aprender a recomendar otras películas de este género. Este tipo de RS presenta problemas como la sobre-especialización, el problema de nuevo usuario (arranque en frío) o el análisis de contenido limitado [RRSK11].

Filtrado colaborativo: la implementación más simple y característica de este enfoque [SFHS07] recomienda al usuario activo los elementos que a otros usuarios, con gustos similares, le gustaron en el pasado. La similitud en el gusto de dos usuarios usualmente se calcula en función de la similitud en el historial de calificación de los usuarios. Esta es la razón por la que en [SKR01] se refiere al filtrado colaborativo como “correlación entre personas”. El filtrado colaborativo es considerado la técnica más popular y ampliamente implementada en RS, y aún se pueden apreciar avances en esta técnica [KB15, YWZ⁺16]. Los métodos basados en el “vecindario” se centran en las relaciones entre elementos o, alternativamente, entre usuarios [BOvS09, BP12, KK17]. Un enfoque de elemento-elemento modela la preferencia de un usuario a un elemento según las calificaciones de elementos similares del mismo usuario. Los métodos de los vecinos más cercanos disfrutaron de una considerable popularidad debido a su simplicidad, eficiencia y su capacidad para producir recomendaciones precisas y personalizadas. Este tipo de RS presentan problemas como la escasez de datos y cobertura limitada, que a menudo se observan en los grandes sistemas de recomendación comercial.

Basado en conocimiento: estos sistemas recomiendan elementos basados en el conocimiento del dominio, sobre cómo ciertas características de los elementos satisfacen las necesidades y preferencias de los usuarios y, de esta manera, cómo el elemento es útil para el usuario [MSR04, CNHAVGGS12, CMVGRG⁺15]. Existen diferentes tipos de estos RS, entre ellos los basados en casos [BGMS05, Smy07] y los basados en restricciones [FB08], si bien son diferentes, en términos del uso del conocimiento son similares. En ambos se recopilan los requerimientos del usuario, se proponen compensaciones automáticamente cuando existen requisitos inconsistentes en situaciones donde no se pueden encontrar soluciones y se explican los resultados de las recomendaciones. La principal diferencia

radica en la forma en que se calculan las soluciones. Los recomendadores basados en casos determinan las recomendaciones basándose en métricas de similitud, mientras que los recomendadores basados en restricciones explotan predominantemente las bases de conocimiento predefinidas que contienen reglas explícitas sobre cómo relacionar los requisitos del usuario con las características del elemento. Los sistemas basados en conocimiento tienden a funcionar mejor que otros al comienzo de su implementación, pero si no están equipados con componentes de aprendizaje, pueden ser superados por otros métodos superficiales que pueden explotar los registros de la interacción usuario-computadora (como el filtrado colaborativo). Este tipo de RS es muy utilizado para ofrecer sugerencias de recursos de aprendizaje electrónico [TNM18].

Basado en comunidad: este tipo de sistemas recomiendan elementos según las preferencias de los amigos de los usuarios. Esta técnica sigue el refrán “dime con quién andas y te diré quién eres” [BP06, BTR⁺07]. La evidencia sugiere que las personas tienden a confiar más en las recomendaciones de sus amigos que en las recomendaciones de personas similares pero anónimas [SS⁺01a]. Esta observación, combinada con la creciente popularidad de las redes sociales abiertas, está generando un interés creciente en los sistemas basados en comunidad o, como se suele hacer referencia, en los sistemas de recomendación social [Gol06, Guy15, DHX⁺16, SKLB17]. Este tipo de RS modela y adquiere información sobre las relaciones sociales de los usuarios y las preferencias de sus amigos. La recomendación se basa en las calificaciones proporcionadas por los amigos del usuario. De hecho, estos RS están siguiendo el auge de las redes sociales y permiten una adquisición simple y completa de datos acerca de las relaciones sociales de los usuarios. La investigación en esta área aún se encuentra en su fase inicial y los resultados sobre el rendimiento de los sistemas son variados. Por ejemplo, [Gol06, MA04] informa que, en general, las recomendaciones basadas en redes sociales no son más precisas que las derivadas de los enfoques tradicionales de filtrado colaborativo, excepto en casos especiales, como cuando las calificaciones de los usuarios de un elemento específico son muy variadas (es decir, elementos controvertidos) o para situaciones de arranque en frío, es decir, donde los usuarios no proporcionaron calificaciones suficientes para calcular la similitud con otros usuarios. Otros han demostrado que, en algunos casos, los datos de redes sociales ofrecen mejores recomendaciones que los datos de similitud por perfil [GZC⁺09] y que la adición de datos de redes sociales al filtrado colaborativo tradicional mejora los resultados de recomendaciones [GE07]. En [EGVT18] se muestran algunos de los retos que enfrentan este tipo de RS.

Demográfico: este tipo de sistema recomienda elementos según el perfil demográfico del usuario [Kru97, Paz99, VM07, SA09]. El supuesto es que se deben generar diferentes recomendaciones para diferentes nichos demográficos. Muchos sitios web adoptan soluciones de personalización simples y efectivas basadas en datos demográficos. Por ejemplo, los usuarios son enviados a sitios web particulares en función de su idioma o país. Además, las sugerencias pueden ser personalizadas de acuerdo a la edad del usuario. Si bien estos enfoques han sido bastante populares en la literatura de mercadeo, ha habido relativamente poca investigación adecuada de RS en sistemas demográficos [MR07].

Sistemas de recomendación híbridos: estos RS se basan en la combinación de las técnicas mencionadas anteriormente [Bur02, BMCMB⁺10, CYKS12, PTLMHV12, TNY17, PBW17, LS19]. Un sistema híbrido que combina las técnicas A y B trata de usar las ventajas de A para solucionar las desventajas de B. Por ejemplo, los métodos de filtrado colaborativo tienen problemas con los nuevos elementos, es decir, no pueden recomendar elementos que no tengan calificaciones. Esto no limita los enfoques basados en contenido, ya que la predicción de nuevos elementos se basa en su información (características) que generalmente están fácilmente disponibles. Es posible combinar de diversas formas dos o más técnicas básicas de RS para crear un nuevo sistema híbrido (en [Bur07] se pueden ver las descripciones precisas).

En las siguientes secciones se describen en mayor detalle los principales tipos de recomendadores mencionados, haciendo hincapié en los basados en filtrado colaborativo, en contenido e híbridos.

2.3.1. Basado en contenido

En los métodos de recomendación basados en contenido, la utilidad $u(c, s)$ del elemento s para el usuario c se estima en función de las utilidades $u(c, s_i)$ asignadas por el usuario c a los elementos $s_i \in S$ que son “similares” al elemento s . Por ejemplo, en un sistema de recomendación de películas, para recomendar películas al usuario c , el RS basado en contenido trata de comprender las características comunes entre las películas que el usuario c ha calificado con puntaje alto en el pasado (actores específicos, directores, géneros, temas, etc.). Entonces, solo se recomendarán las películas que tengan un alto grado de similitud con las preferencias del usuario.

El enfoque de la recomendación basado en contenido tiene sus raíces en la recuperación de información (IR, del inglés *Information Retrieval*) [BR99, Sal89] y en la investigación de filtrado de información [BC92]. Debido a los importantes y tempranos avances realizados por las comunidades de recuperación y filtrado de información y por la importancia de varias aplicaciones basadas en texto, muchos sistemas actuales basados en contenido se centran en recomendar elementos que contienen información textual, como documentos, sitios web (URL), o mensajes de noticias de Usenet. La mejora con respecto a los enfoques tradicionales de IR proviene del uso de perfiles de usuario que contienen información sobre los gustos, preferencias y necesidades de los usuarios. La información de perfil se puede obtener de los usuarios de manera explícita, por ejemplo, a través de cuestionarios, o se puede aprender de forma implícita de su comportamiento transaccional a lo largo del tiempo.

Más formalmente, *Contenido(s)* es el perfil de un elemento, es decir, un conjunto de atributos que caracterizan al elemento s . Por lo general, se calcula mediante la extracción de un conjunto de características del elemento s (su contenido) y se utiliza para determinar si el elemento es apropiado para fines de recomendación. Como se mencionó anteriormente, los sistemas basados en contenido están diseñados principalmente para recomendar elementos basados en texto, el contenido en estos sistemas generalmente se describe con palabras clave. Por ejemplo, un componente basado en contenido del sistema Fab [BS97], que recomienda páginas web a los usuarios, representa el contenido de la página web con las 100 palabras más importantes. De manera similar, el sistema Syskill & Webert [PB97] representa documentos con las 128 palabras más informativas. La “importancia” (o “informatividad”) de la palabra k_i en el documento d_j se determina con alguna medida de peso, w_{ij} , que puede definirse de varias maneras diferentes.

Una de las medidas más conocidas para especificar pesos de palabras clave en IR es: frecuencia de término/frecuencia inversa de documento (TF-IDF, del inglés *Term Frequency/Inverse Document Frequency*) [Sal89] que se define de la siguiente manera. Supongamos que N es el número total de documentos que se pueden recomendar a los usuarios y que la palabra clave k_i aparece en n_i de ellos. Además, suponiendo que $f_{i,j}$ es el número de veces que aparece la palabra clave k_i en el documento d_j . Entonces $TF_{i,j}$, la frecuencia del término (o frecuencia normalizada) de la palabra clave k_i en el documento d_j , se define como:

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \quad (2.1)$$

donde el máximo se calcula sobre las frecuencias $f_{z,j}$ de todas las palabras clave k_z que aparecen en el documento d_j . Sin embargo, las palabras clave que aparecen en muchos documentos no son útiles para distinguir entre un documento relevante y uno no relevante. Por lo tanto, la medida de la frecuencia inversa del documento (IDF_i) se usa a menudo en combinación con la simple frecuencia de término ($TF_{i,j}$). La frecuencia inversa del documento para la palabra clave k_i se define generalmente como:

$$IDF_i = \log \frac{N}{n_i} \quad (2.2)$$

Luego, el peso de TF-IDF para la palabra clave k_i en el documento d_j se define como:

$$w_{i,j} = TF_{i,j} \times IDF_i \quad (2.3)$$

y el contenido del documento d_j se define como $Contenido(d_j) = (w_{1j}, \dots, w_{kj})$.

Como se indicó anteriormente, los sistemas basados en contenido recomiendan elementos similares a los que le gustaban a un usuario en el pasado [DGLM⁺15, MSdGL16, GZHS16, BHM17]. En particular, varios elementos candidatos se comparan con elementos previamente calificados por el usuario, y se recomiendan los elementos que mejor coinciden. Más formalmente, *PerfilBasadoEnContenido(c)* es el perfil del usuario c que contiene los gustos y preferencias de este usuario. Estos perfiles se obtienen al analizar el contenido de los elementos vistos y calificados anteriormente por el usuario y, por lo general, se construyen utilizando técnicas de IR de análisis de palabras clave. Por ejemplo, *PerfilBasadoEnContenido(c)* se puede definir como un vector de ponderaciones (w_{c1}, \dots, w_{ck}) , donde cada ponderación w_{ci} denota la importancia de la palabra clave k_i para el usuario c y se puede calcular a partir de vectores de contenido con calificación individual utilizando una variedad de técnicas. Por ejemplo, en [PB97] para poder calcular *PerfilBasadoEnContenido(c)* se usa un clasificador bayesiano para estimar la probabilidad de que un documento sea de su agrado. El algoritmo de Winnow [LW94] también se ha demostrado que funciona bien para este propósito, especialmente en las situaciones en las que hay muchas características posibles [Paz99]. En los sistemas basados en contenido, la función de utilidad $u(c, s)$ generalmente se define como:

$$u(c, s) = \text{Calificación}(\text{PerfilBasadoEnContenido}(c), \text{Contenido}(s)) \quad (2.4)$$

Utilizando el paradigma basado en IR mencionado anteriormente para recomendar páginas web, URLs de sitios web o mensajes de noticias de *Usenet*, tanto *PerfilBasadoEnContenido(c)* del usuario c como *Contenido(s)* del documento s pueden representarse como vectores TF-IDF \vec{w}_c y \vec{w}_s de pesos de palabras clave. Además, la función de utilidad $u(c, s)$ suele estar representada en la literatura de IR mediante una heurística de calificación definida en términos de vectores \vec{w}_c y \vec{w}_s , como la medida de similitud por coseno [BR99, Sal89, BMCMB⁺10, ASA⁺10]:

$$u(c, s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2} = \frac{\sum_{i=1}^K w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^K w_{i,c}^2} \sqrt{\sum_{i=1}^K w_{i,s}^2}} \quad (2.5)$$

donde K es el número total de palabras clave en el sistema.

Por ejemplo, si el usuario c lee muchos artículos sobre el tema de la bioinformática, las técnicas de recomendación basadas en contenido podrán recomendar otros artículos de bioinformática al usuario c . Esto se debe a que estos artículos tendrán más términos relacionados con la bioinformática (por ejemplo, “genoma”, “secuenciación”, “proteómica”) que artículos sobre otros temas, por lo tanto, *PerfilBasadoEnContenido(c)*, definido por vector \vec{w}_c , representará dichos términos k_i con pesos altos $w_{i,c}$. En consecuencia, un sistema de recomendación que utilice el coseno o una medida de similitud relacionada asignará una mayor utilidad $u(c, s)$ a aquellos artículos s que tengan términos de bioinformática de alto peso en \vec{w}_s y una utilidad más baja a aquellos en los que los términos de bioinformática que pesen menos.

Además de las heurísticas tradicionales que se basan principalmente en los métodos de IR, también se han utilizado otras técnicas para la recomendación basada en el contenido, como los clasificadores bayesianos [PB97, MBR98, PHC07, Agg16] y varias técnicas de aprendizaje automático (ML, del inglés *Machine Learning*), incluido el agrupamiento (en inglés, *clustering*) [KA08, SK12, GPB14], los árboles de decisión y las redes neuronales artificiales (ANN, del inglés *Artificial Neural Network*) [PB97, LCW02, CVS07]. Estas técnicas difieren de los enfoques basados en IR en que calculan predicciones de utilidad no basándose en una fórmula heurística, como una medida de similitud por coseno, sino que se basan en un modelo aprendido de los datos subyacentes mediante el aprendizaje

estadístico y las técnicas de ML. Por ejemplo, basado en un conjunto de páginas web previamente calificadas como “relevantes” o “irrelevantes” por el usuario, en [PB97] se utiliza un clasificador bayesiano naïve para clasificar nuevas páginas web. Más específicamente, el clasificador bayesiano naïve [DHS01] se usa para estimar la probabilidad de que la página p_j pertenezca a una determinada clase C_i (por ejemplo, relevante o irrelevante) dado el conjunto de palabras clave $k_{1,j}, \dots, k_{n,j}$ en esa página:

$$P(C_i | k_{1,j}, \dots, k_{n,j}) \quad (2.6)$$

Además, [PB97] utiliza el supuesto de que las palabras clave son independientes y, por lo tanto, la probabilidad anterior es proporcional a

$$P(C_i) \prod_x P(k_{x,j} | C_i) \quad (2.7)$$

Si bien el supuesto de independencia de las palabras claves no se aplica necesariamente en muchas aplicaciones, los resultados experimentales demuestran que los clasificadores bayesianos naïve producen una alta precisión de clasificación [PB97]. Además, tanto, $P(k_{x,j} | C_i)$ como $P(C_i)$ se pueden estimar a partir de los datos de entrenamiento subyacentes. Por lo tanto, para cada página p_j , la probabilidad $P(C_i | k_{1,j}, \dots, k_{n,j})$ se calcula para cada clase C_i , y la página p_j se asigna a la clase C_i que tiene la mayor probabilidad [PB97].

Si bien, la comunidad de recuperación de texto (en inglés *text retrieval*), no se ocupa explícitamente de proporcionar recomendaciones, ha contribuido con varias técnicas que se utilizan en los sistemas de recomendación basados en contenido. Un ejemplo de estas técnicas sería la investigación sobre el filtrado adaptativo (en inglés *adaptive filtering*) [SH01, ZCM02], que se centra en ser más precisos en la identificación de documentos relevantes de forma incremental, mediante la observación de los documentos uno por uno en un flujo continuo de documentos. Otro ejemplo sería el trabajo sobre la configuración del umbral (en inglés *threshold setting*) [RW00, ZC01], que se centra en determinar la medida en que los documentos deben coincidir con una consulta determinada para que sean relevantes para el usuario. Otros métodos de recuperación de texto se describen en [Hul98] y también se pueden encontrar en distintas ediciones de la Conferencia de Recuperación de Texto (TREC, del inglés *Text Retrieval Conference*)¹².

¹²<http://trec.nist.gov>

Como se observó en [SM95, BS97, RRSK11, BOHG13], los sistemas de recomendación basados en contenido tienen varias limitaciones que se describen en el resto de esta sección.

Análisis de contenido limitado. Las técnicas basadas en contenido están limitadas por las características que están asociadas explícitamente con los elementos que estos sistemas recomiendan. Por lo tanto, para tener un conjunto suficiente de características, el contenido debe estar en una forma que pueda ser analizada automáticamente por una computadora (por ejemplo, texto), o las características deben asignarse a los elementos manualmente. Si bien las técnicas de IR funcionan bien en la extracción de características de documentos de texto, en algunos otros dominios tienen un problema inherente con la extracción automática de características. Por ejemplo, los métodos de extracción automática de características son mucho más difíciles de aplicar a los datos multimedia, por ejemplo, imágenes gráficas, audio y secuencias de video. Además, a menudo no es práctico asignar atributos manualmente debido a las limitaciones de recursos [SM95]. Otro problema con el análisis de contenido limitado es que, si dos elementos diferentes están representados por el mismo conjunto de características, son indistinguibles.

Sobre-especialización. Cuando el sistema solo puede recomendar elementos que tienen una puntuación alta en relación con el perfil de un usuario, el usuario se limita a recibir recomendaciones de elementos similares a los que ya ha calificado [AT05]. Por ejemplo, una persona sin experiencia con la cocina griega nunca recibiría una recomendación ni siquiera para el mejor restaurante griego de la ciudad. Este problema, que también se ha estudiado en otros dominios, a menudo se resuelve introduciendo algo de aleatoriedad. Por ejemplo, se ha propuesto el uso de algoritmos genéticos como una posible solución en el contexto del filtrado de información [SM93]. Además, el problema con la especialización excesiva no es solo que los sistemas basados en contenido no pueden recomendar elementos que sean diferentes de cualquier cosa que el usuario haya visto antes. En ciertos casos, los elementos no deben recomendarse si son demasiado similares a algo que el usuario ya ha visto, como un artículo de noticias diferente que describe el mismo evento. Por lo tanto, algunos sistemas de recomendación basados en contenido, como DailyLearner [BP00], filtran los elementos no solo si son demasiado diferentes de las preferencias del usuario, sino también si son demasiado similares a algo que el usuario ha visto anteriormente. Además, [ZCM02] proporciona un conjunto de cinco medidas de redundancia para evaluar si un documento que se considera relevante también contiene información novedosa. En resumen, la novedad y la diversidad de recomendaciones suelen ser carac-

terísticas deseables en los sistemas de recomendación [HZ11, BHOB11, KP17, NHTK18]. Idealmente, al usuario se le debe presentar una gama de opciones y no un conjunto homogéneo de alternativas. Por ejemplo, no es necesariamente una buena idea recomendar todas las películas de Woody Allen a un usuario al que le haya gustado solo una de ellas.

Problema de usuario nuevo. El usuario debe calificar un número suficiente de elementos antes de que un sistema de recomendación basado en contenido pueda entender realmente las preferencias del usuario y presentarle recomendaciones confiables. Por lo tanto, un nuevo usuario, con muy pocas calificaciones, no podría obtener recomendaciones precisas.

2.3.2. Filtrado colaborativo

Los sistemas de recomendación colaborativos (o sistemas de filtrado colaborativo) intentan predecir la utilidad de los elementos para un usuario en particular en función de los elementos previamente calificados por otros usuarios. De manera más formal, la utilidad $u(c, s)$ del elemento s para el usuario c se estima en función de las utilidades $u(c_j, s)$ asignadas al elemento s por aquellos usuarios $c_j \in C$ que son “similares” al usuario c . Por ejemplo, en una aplicación de recomendación de películas, para recomendar películas al usuario c , el sistema de recomendación colaborativa intenta encontrar los “compañeros” del usuario c , es decir, otros usuarios que tienen gustos similares para las películas (es decir, califican las mismas películas de manera similar). Entonces, solo se recomendarán las películas que más les gustan a los “compañeros” del usuario c . Han habido muchos sistemas de recomendación colaborativos desarrollados en la academia y la industria. Se puede mencionar al sistema *Grundy* [Ric79], que fue el primer sistema de recomendación, que proponía utilizar estereotipos como un mecanismo para construir modelos de usuarios basados en una cantidad limitada de información sobre cada usuario individual. Usando estereotipos, el sistema *Grundy* construiría modelos de usuario individuales y los usaba para recomendar libros relevantes para cada usuario. Posteriormente, el sistema *Tapestry* se basó en cada usuario para identificar a los usuarios afines manualmente [GNOT92]. *GroupLens* [RIS⁺94, KMM⁺97], *Video Recomennder* [HSRF95] y *Ringo* [SM95] fueron los primeros sistemas que utilizaron algoritmos de filtrado colaborativo para automatizar la predicción. Pueden nombrarse otros ejemplos de sistemas de recomendación colaborativos como el sistema de recomendación de libros de *Amazon.com*, el sistema *PHOAKS* que ayuda a las personas a encontrar información relevante en la Web [THA⁺97], el

sistema de *Jester* que recomienda bromas [GRGP01], o el sistema de *Last.fm* que se encarga de recomendar música [DYD16]. De acuerdo con [BOHG13], los algoritmos para recomendaciones colaborativas pueden agruparse en dos clases generales: *basados en memoria* (o basados en heurística) y *basados en modelos*. Los algoritmos basados en memoria [SKY05, AT05, CMB07, SNM09] utilizan esencialmente heurísticas que realizan predicciones de calificación basadas en la colección completa de elementos previamente calificados por los usuarios. Es decir, el valor de la calificación desconocida $r_{c,s}$ para el usuario c y el elemento s generalmente se calcula como un agregado de las calificaciones de otros usuarios (generalmente los N más similares) para el mismo elemento s :

$$r_{c,s} = \text{agregación}_{(c' \in \hat{C})} r_{c',s} \quad (2.8)$$

donde \hat{C} representa el conjunto de los N usuarios que son más similares al usuario c y además calificaron al elemento s (N puede tomar el valor desde 1 hasta la cantidad total de usuarios). Algunos ejemplos de una función de agregación son:

$$r_{c,s} = \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s} \quad (2.9)$$

$$r_{c,s} = k \sum_{c' \in \hat{C}} \text{sim}(c, c') \times r_{c',s} \quad (2.10)$$

$$r_{c,s} = \bar{r}_c + k \sum_{c' \in \hat{C}} \text{sim}(c, c') \times (r_{c',s} - \bar{r}_{c'}) \quad (2.11)$$

donde k sirve como factor de normalización y generalmente se selecciona como:

$$k = 1 / \sum_{c' \in \hat{C}} |\text{sim}(c, c')|$$

y donde la calificación promedio del usuario c , \bar{r}_c , en 2.11 se define como:

$$\bar{r}_c = (1/|S_c|) \sum_{s' \in S_c} r_{c,s'}, \text{ donde } S_c = \{s \in S | r_{c,s} \neq \emptyset\}. \quad (2.12)$$

La notación $r_{c,s} = \emptyset$ indica que el elemento s no fue calificado por el usuario c .

En el caso más simple, la agregación puede ser un promedio simple, como se define en la expresión 2.9. Sin embargo, el enfoque de agregación más común es utilizar la suma ponderada, que se muestra en la expresión 2.10. La medida de similitud entre los usuarios c y c' , $sim(c, c')$, es esencialmente una medida de distancia y se usa como un peso, es decir, cuanto más similares son los usuarios c y c' , $r_{c',s}$ tendrá mayor peso en la calificación de la predicción de $r_{c,s}$. Hay que tener en cuenta que $sim(x, y)$ es un artefacto heurístico que se introduce para poder diferenciar niveles de similitud entre usuarios (es decir, para poder encontrar un conjunto de “pares más cercanos” o “vecinos más cercanos” para cada usuario) y al mismo tiempo simplificar el procedimiento de estimación de calificación. Como se muestra en la expresión 2.10, diferentes aplicaciones de recomendación pueden usar su propia medida de similitud de usuario, siempre que los cálculos se normalicen utilizando el factor de normalización k , como se muestra arriba. A continuación se describen las dos medidas de similitud más utilizadas. Un problema con el uso de la suma ponderada, como en la expresión 2.10, es que no tiene en cuenta el hecho de que diferentes usuarios pueden usar la escala de calificación de manera diferente. La suma ponderada ajustada, que se muestra en la expresión 2.11, se ha utilizado ampliamente para abordar esta limitación. En este enfoque, en lugar de utilizar los valores absolutos de las calificaciones, la suma ponderada utiliza sus desviaciones de la calificación promedio del usuario correspondiente. Otra forma de superar los diferentes usos de la escala de calificación es implementar el filtrado basado en preferencias [FISS98, CSS99, JSZ03, JSZC03, Smy07, YST⁺13], que se centra en predecir las preferencias relativas de los usuarios en lugar de los valores de calificación absolutos.

Se han utilizado varios enfoques para calcular la similitud $sim(c, c')$ entre usuarios en sistemas de recomendación colaborativos. En la mayoría de estos enfoques, la similitud entre dos usuarios se basa en las calificaciones de los elementos que ambos usuarios han calificado. Los dos enfoques más populares se basan en la correlación y el coseno. Para presentarlos, sea S_{xy} el conjunto de todos los elementos calificados por los usuarios x e y , es decir, $S_{xy} = \{s \in S | r_{x,s} \neq \emptyset \ \& \ r_{y,s} \neq \emptyset\}$. En los sistemas de recomendación colaborativos, S_{xy} se utiliza principalmente como un resultado intermedio para calcular los “vecinos más cercanos” del usuario x y a menudo se calcula de manera directa, es decir, calculando la intersección de los conjuntos S_x y S_y . Sin embargo, algunos métodos, como el enfoque teórico-gráfico para el filtrado colaborativo [AWWY99], pueden determinar los vecinos más cercanos de x sin calcular S_{xy} para todos los usuarios y . En el enfoque basado en correlación, el coeficiente de correlación de Pearson se usa para medir la similitud [RIS⁺94,

SM95, ASA⁺10, RRSK11]:

$$\text{sim}(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)(r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}} \quad (2.13)$$

En el enfoque basado en coseno [BHK98, SKKR01], los dos usuarios x e y se tratan como dos vectores en el espacio m -dimensional, donde $m = S_{xy}$. Luego, la similitud entre dos vectores se puede medir calculando el coseno del ángulo entre ellos:

$$\text{sim}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \times \|\vec{y}\|_2} = \frac{\sum_{s \in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} r_{x,s}^2} \sqrt{\sum_{s \in S_{xy}} r_{y,s}^2}} \quad (2.14)$$

donde $\vec{x} \cdot \vec{y}$ denota el producto entre los vectores \vec{x} e \vec{y} . Otro enfoque más para medir la similitud entre usuarios utiliza la medida de la diferencia de la media al cuadrado y se describe en [SM95]. Hay que tener en cuenta que los diferentes sistemas de recomendación pueden adoptar diferentes enfoques para implementar los cálculos de similitud de los usuarios y las estimaciones de calificación de la manera más eficiente posible. Una estrategia común es calcular todas las similitudes de usuario $\text{sim}(x, y)$ (incluido el cálculo de S_{xy}) por adelantado y recalcularlas solo de vez en cuando (ya que la red de pares generalmente no cambia drásticamente en poco tiempo). Luego, cuando el usuario solicita una recomendación, las calificaciones se pueden calcular de manera eficiente según la demanda utilizando similitudes previamente calculadas. Tanto el enfoque basado en contenido como el colaborativo utilizan la misma medida de coseno de la literatura de recuperación de información. Sin embargo, en los sistemas de recomendación basados en contenido, se utiliza para medir la similitud entre los vectores de ponderaciones TF-IDF, mientras que en los sistemas colaborativos mide la similitud entre los vectores de las calificaciones reales especificadas por el usuario. Se han propuesto muchas modificaciones para mejorar el rendimiento de estas técnicas estándar basadas en la correlación y en el coseno, como la calificación predeterminada, la frecuencia inversa del usuario, la amplificación de casos [BHK98] y la predicción de mayoría ponderada [NA98, DI99]. Por ejemplo, la calificación predeterminada [BHK98] es una extensión de los enfoques basados en memoria descritos anteriormente. Se observó que siempre que haya relativamente pocas calificaciones especificadas por el usuario, estos métodos no funcionarán bien para calcular la similitud entre los usuarios x e y [RRSK11], ya que la medida de similitud se basa en la intersección de los conjuntos de elementos, es decir, conjuntos de elementos calificados

por ambos usuarios x e y . Se demostró empíricamente que la precisión de la predicción de calificación podría mejorar si se asume algún valor de calificación predeterminado para las calificaciones faltantes [BHK98]. Además, mientras que las técnicas anteriores tradicionalmente se han utilizado para calcular similitudes entre usuarios, en [ASA⁺10, KBGM15] se observa otra alternativa, que es la de calcular similitudes entre elementos. Por ejemplo, en [SKKR01] propusieron usar las mismas técnicas basadas en correlación y coseno para calcular similitudes entre elementos y obtener las calificaciones de ellos. Esta idea se ha ampliado aún más en [DK04] para las recomendaciones de los primeros N elementos. Además, [SKKR01, DK04] presenta evidencia empírica de que los algoritmos basados en elementos pueden proporcionar un mejor rendimiento computacional que los métodos colaborativos tradicionales basados en el usuario, mientras que al mismo tiempo proporcionan una calidad comparable o mejor que los mejores algoritmos disponibles basados en el usuario. En contraste con los métodos basados en memoria, los algoritmos basados en modelos [Hof03, PP03, Mar04, AT05, SK09, OMDG17] utilizan la colección de calificaciones para definir un modelo, que luego se usa para hacer predicciones de calificación. Por ejemplo, [BHK98] propone un enfoque probabilístico para el filtrado colaborativo, donde las calificaciones desconocidas se calculan como:

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^n i \times \Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c) \quad (2.15)$$

y se asume que los valores de calificación son números enteros entre 0 y n , y la expresión de probabilidad es la probabilidad de que el usuario c otorgue una calificación particular al elemento s , dadas las calificaciones previas de ese usuario para otros elementos. Para estimar esta probabilidad, [BHK98] propone dos modelos probabilísticos alternativos: modelos de agrupamiento (en inglés *cluster*) y redes bayesianas. En el primer modelo, los usuarios afines se agrupan en clases. Dada la clase a la que pertenece el usuario, se asume que las calificaciones del usuario son independientes, es decir, la estructura del modelo es la de un modelo bayesiano naïve. El número de clases y los parámetros del modelo se aprenden de los datos. El segundo modelo representa cada elemento en el dominio como un nodo en una red bayesiana, donde los estados de cada nodo corresponden a los posibles valores de calificación para cada elemento. Tanto la estructura de la red como las probabilidades condicionales se aprenden de los datos. Una limitación de este enfoque es que cada usuario puede agruparse en una sola categoría, mientras que algunos sistemas de recomendación pueden beneficiarse de la capacidad de agrupar usuarios en varias categorías a la vez. Por

ejemplo, en un sistema de recomendación de libros, un usuario puede estar interesado en un tema (por ejemplo, programación) para fines de trabajo y un tema completamente diferente (por ejemplo, pedestrismo) para el tiempo libre. Además, en [BP98a] se propone un método de filtrado colaborativo en un marco de aprendizaje automático, en el que varias técnicas de ML son usadas. Técnicas como las redes neuronales artificiales o técnicas de extracción de características, como la descomposición en valores singulares (SVD, del inglés *Singular Value Decomposition*) [CRR⁺13, NbII14, NIB18], una técnica algebraica para reducir la dimensionalidad de las matrices. Han habido varios otros enfoques de recomendación colaborativa basados en modelos propuestos en la literatura. Un modelo estadístico para el filtrado colaborativo se propuso en [UF98], y se compararon varios algoritmos diferentes para estimar los parámetros del modelo, incluyendo el agrupamiento de K-means y el muestreo de Gibbs. Otros métodos de filtrado colaborativo incluyen un modelo bayesiano [CG99], un modelo relacional probabilístico [GS⁺99], una regresión lineal [SKKR01], y un modelo de máxima entropía [PP03]. Se ha realizado una cantidad significativa de investigación para intentar modelar el proceso de recomendación utilizando modelos probabilísticos más complejos. Por ejemplo, [SHB05] ve el proceso de recomendación como un problema de decisión secuencial y propone utilizar los procesos de decisión de Markov (una técnica estocástica bien conocida para modelar decisiones secuenciales) para generar recomendaciones. Otras técnicas de modelado probabilístico para sistemas de recomendación incluyen el análisis semántico probabilístico latente [Hof03, Hof04] y una combinación de mezcla multinomial y modelos de aspecto usando semántica generativa de asignación latente de Dirichlet [Mar04]. De manera similar, [SJ03] también utiliza el análisis semántico latente probabilístico para proponer un modelo de mezcla flexible que permita modelar las clases de usuarios y elementos explícitamente con dos conjuntos de variables latentes. Además, [KRRT01] utiliza un modelo probabilístico simple para demostrar que el filtrado colaborativo es valioso con relativamente pocos datos sobre cada usuario, y que, en ciertas configuraciones restringidas, los algoritmos de filtrado colaborativo simple son casi tan efectivos como los mejores algoritmos posibles en términos de utilidad. Al igual que en el caso de las técnicas basadas en contenido, la principal diferencia entre las técnicas basadas en modelos colaborativos y los enfoques basados en heurística es que las técnicas basadas en modelos calculan predicciones de utilidad (*calificación*) basadas no en algunas reglas heurísticas ad hoc, sino más bien basadas en un modelo aprendido de los datos subyacentes utilizando técnicas estadísticas y de aprendizaje automático. En [PHLG00] se propuso un método que combinaba los enfoques basados

en memoria y los basados en modelos, donde se demostró empíricamente que el uso de este enfoque combinado puede proporcionar mejores recomendaciones que los enfoques de colaboración puramente basados en memoria o en modelos.

Se adoptó un enfoque diferente para mejorar el rendimiento de los algoritmos de filtrado colaborativo existentes en [YXT⁺02], donde el conjunto de entrada de clasificaciones especificadas por el usuario se selecciona cuidadosamente utilizando varias técnicas que excluyen el ruido, la redundancia y explotan la escasez de datos de calificaciones. Los resultados empíricos demuestran el aumento de la precisión y la eficiencia de los algoritmos de filtrado colaborativo basados en modelos. También se sugiere que las técnicas de selección de entrada propuestas pueden ayudar a los algoritmos basados en modelos para abordar el problema de aprender de grandes bases de datos [YXT⁺02]. Además, en [YST⁺04] proponen un enfoque probabilístico para el filtrado colaborativo que constituye otra forma de combinar las técnicas basadas en memoria y modelos. En particular, [YST⁺04] propone (a) utilizar un enfoque de aprendizaje activo para aprender el modelo probabilístico de las preferencias de cada usuario y (b) utilizar los perfiles de usuario almacenados en un modelo mixto para calcular recomendaciones. El último aspecto del enfoque propuesto implementa algunas de las ideas utilizadas en los algoritmos tradicionales basados en memoria.

Los sistemas de recomendación colaborativa pura no tienen algunas de las deficiencias que tienen los sistemas basados en contenido. En particular, ya que los sistemas colaborativos utilizan recomendaciones de otros usuarios (calificaciones), pueden tratar cualquier tipo de contenido y recomendar cualquier elemento, incluso los que son diferentes a los que se vieron en el pasado. Sin embargo, los sistemas colaborativos tienen sus propias limitaciones [BS97, Lee01, CCFF11, BOHG13, OMDG17], como se describe a continuación.

Problema de usuario nuevo. Es el mismo problema que con los sistemas basados en contenido. Para hacer recomendaciones precisas, el sistema debe primero conocer las preferencias del usuario a partir de las calificaciones que el usuario realiza [dBRB05, RKR08]. Se han propuesto varias técnicas para abordar este problema [SL12, ML13]. La mayoría de ellas utilizan un enfoque de recomendación híbrida, que combina técnicas basadas en contenido y colaborativas. En una sección posterior se describe a los sistemas de recomendación híbridos con más detalle. Un enfoque alternativo se presenta en [RAC⁺02, YST⁺04], donde se exploran varias técnicas para determinar los mejores elementos (es

decir, los más informativos para el sistema de recomendación) para que los califique un nuevo usuario. Estas técnicas utilizan estrategias que se basan en la popularidad de los elementos, la entropía de los elementos, la personalización del usuario y las combinaciones de los anteriores [RAC⁺02, YST⁺04].

Problema de elemento nuevo. Nuevos elementos se añaden regularmente a los sistemas de recomendación. Los sistemas colaborativos dependen únicamente de las preferencias de los usuarios para hacer recomendaciones. Por lo tanto, hasta que el nuevo elemento no sea calificado por un número sustancial de usuarios, el RS no podrá recomendarlo [WHC⁺17]. Este problema también se puede abordar utilizando enfoques de recomendación híbridos, que se describen en una sección posterior. Por ejemplo en [OMDG17], se utilizan modelos de factores latentes para la recomendación y se predicen estos factores a partir de las descripciones de los elementos que utilizan una red neuronal convolucional profunda cuando no se pueden obtener a partir de los datos de uso.

Escasez. En cualquier sistema de recomendación, la cantidad de calificaciones ya obtenidas es generalmente muy pequeña en comparación con la cantidad de calificaciones que se deben predecir. La predicción efectiva de las calificaciones a partir de un pequeño número de ejemplos es importante. Además, el éxito del sistema de recomendación colaborativo depende de la disponibilidad de una masa crítica de usuarios. Por ejemplo, en el sistema de recomendación de películas puede haber muchas películas que solo hayan sido calificadas por pocas personas y estas películas se recomendarían muy raramente, incluso si esos pocos usuarios les dieran calificaciones altas. Además, para el usuario cuyos gustos son inusuales en comparación con el resto de la población, no habrá otros usuarios que sean particularmente similares, lo que dará lugar a recomendaciones deficientes [BS97]. Una forma de superar el problema de la escasez de clasificaciones es utilizar la información del perfil del usuario al calcular la similitud del usuario. Es decir, dos usuarios podrían considerarse similares no solo si calificaban las mismas películas de manera similar, sino también si pertenecían al mismo segmento demográfico. Por ejemplo, [SL12] incorpora información adicional para aliviar este problema. Esta información se obtiene de la red de confianza social de los usuarios y del conocimiento semántico de los elementos del dominio. En otra propuesta, [Paz99] utiliza información de género, edad, código de área, educación y empleo de los usuarios en la aplicación de recomendación de restaurantes. Esta extensión de las técnicas tradicionales de filtrado colaborativo a veces se denomina “filtrado demográfico” [Paz99]. Otro enfoque que también explora las similitudes entre los

usuarios se ha propuesto en [HCZ04], donde el problema de la escasez se aborda mediante la aplicación de un marco de recuperación asociativa y algoritmos de activación de propagación relacionados para explorar asociaciones transitivas entre los consumidores a través de sus transacciones y comentarios anteriores. Se utilizó un enfoque diferente para tratar con matrices de calificación escasas en [BP98a, SKKR00a, NbII14, NIB18], donde se usó una técnica de reducción de dimensionalidad, descomposición en valores singulares, para reducir la dimensionalidad de matrices de calificación escasas. La SVD es un método bien conocido para la factorización matricial que proporciona las mejores aproximaciones de rango inferior de la matriz original [SKKR00a].

2.3.3. Sistemas híbridos

Varios sistemas de recomendación utilizan un enfoque híbrido al combinar métodos colaborativos y basados en contenido, lo que ayuda a evitar ciertas limitaciones de los sistemas basados en contenido y colaborativos [Bur07, BMCMB⁺10, SK12, TNY17, LS19]. Las diferentes formas de combinar métodos colaborativos y basados en contenido en un sistema de recomendación híbrido se pueden clasificar de la siguiente manera: (1) implementando métodos colaborativos y basados en contenido por separado y combinando sus predicciones, (2) incorporando algunas características basadas en contenido en un enfoque colaborativo, (3) incorporando algunas características colaborativas en un enfoque basado en contenido, y (4) construyendo un modelo unificador general que incorpore características tanto de contenido como colaborativas. Todos los enfoques anteriores han sido utilizados en el área de RS, como se describe a continuación.

1. Combinación de recomendadores separados. Una forma de crear sistemas de recomendación híbridos es implementar sistemas colaborativos y basados en contenido separados. Entonces podemos tener dos escenarios diferentes. Primero, podemos combinar los resultados (*calificaciones*) obtenidos de los sistemas de recomendación individuales en una recomendación final utilizando una combinación lineal de calificaciones [MCG⁺99] o un esquema de votación [Paz99]. Alternativamente, podemos usar uno de los recomendadores individuales, en cualquier momento, eligiendo usar el que es “mejor” que otros, en base a alguna métrica de “calidad” de recomendación. Por ejemplo, el sistema DailyLearner [BP00] selecciona el sistema de recomendación que puede dar la recomendación con el mayor nivel de confianza, mientras que [TC00] elige el sistema cuya recomendación es más consistente con las calificaciones anteriores del usuario.

2. Agregar características basadas en contenido a los modelos colaborativos. Varios sistemas de recomendación híbridos, incluidos Fab [BS97] y el enfoque de “colaboración a través del contenido”, descrito en [Paz99], se basan en técnicas colaborativas tradicionales pero también mantienen los perfiles basados en contenido para cada usuario. Estos perfiles basados en contenido, y no los elementos calificados comúnmente, se utilizan para calcular la similitud entre dos usuarios. Como se mencionó en [Paz99], esto permite superar algunos problemas relacionados con la escasez de un enfoque puramente colaborativo, ya que normalmente no muchos pares de usuarios tendrán un número significativo de elementos calificados comúnmente. Otro beneficio de este enfoque es que a los usuarios se les puede recomendar un elemento no solo cuando lo califican positivamente los usuarios con perfiles similares, sino también de manera directa, es decir, cuando este elemento tiene una alta calificación respecto al perfil del usuario [BS97]. En [GSK⁺99] se emplea un enfoque algo similar usando una variedad de diferentes tipos de filtro, en forma de agentes especializados de análisis de contenido que actúan como participantes adicionales en una comunidad de filtrado colaborativo. Como resultado, los usuarios cuyas calificaciones concuerden con algunas de las calificaciones de los agentes de filtrado podrán recibir mejores recomendaciones [GSK⁺99]. De manera similar, [MMN02] utiliza un enfoque colaborativo en el que el vector de calificaciones del usuario tradicional se aumenta con calificaciones adicionales, que se calculan utilizando un predictor basado puramente en el contenido.

3. Agregar características colaborativas a los modelos basados en contenido. El enfoque más popular en esta categoría es usar alguna técnica de reducción de dimensionalidad en un grupo de perfiles basados en contenido. Por ejemplo, [SN99] utiliza la indexación semántica latente (LSI, del inglés *Latent Semantic Indexing*) para crear una vista colaborativa de una colección de perfiles de usuario, donde los perfiles de usuario están representados por vectores de términos (como se explica en la Sección 2.3.1), lo que resulta en una mejora del rendimiento en comparación con el enfoque basado en contenido estrictamente puro.

4. Desarrollar un modelo único de recomendación unificado. Algunos investigadores han seguido este enfoque en los últimos años. Por ejemplo, [BHC98] propone usar características basadas en contenido y colaborativas (por ejemplo, la edad o el género de los usuarios o el género de películas) en un único clasificador basado en reglas. En [PPL01, SPUP02] se proponen métodos probabilísticos unificados para combinar re-

comendaciones colaborativas y basadas en contenido, que se basa en el análisis semántico latente probabilístico [Hof99].

Los sistemas de recomendación híbridos también pueden aumentarse mediante técnicas basadas en conocimiento [Bur00], como el razonamiento basado en casos, con el fin de mejorar la precisión de la recomendación y abordar algunas de las limitaciones (por ejemplo, los problemas de nuevos usuarios o nuevos elementos) de los sistemas de recomendación tradicionales. Por ejemplo, Entrée [Bur00], es un sistema de recomendación basado en conocimiento, utiliza conocimientos del dominio sobre restaurantes, cocinas y alimentos (por ejemplo, que la “comida de mar” no es “vegetariana”) para recomendar restaurantes a sus usuarios. El principal inconveniente de los sistemas basados en conocimiento es la necesidad de la adquisición de conocimiento, un conocido cuello de botella para muchas aplicaciones de inteligencia artificial. Sin embargo, se han desarrollado sistemas de recomendación basados en conocimiento para dominios de aplicación donde el conocimiento del dominio está fácilmente disponible en alguna forma estructurada y legible por máquinas, por ejemplo, como una ontología. Por ejemplo, los sistemas Quickstep y Foxtrot [MSR04] utilizan la ontología del dominio de los artículos de investigación para hacer recomendaciones en línea de artículos a los usuarios. Además, varios artículos, como [BS97, Paz99, SN99, MMN02], comparan empíricamente el rendimiento del enfoque híbrido con los métodos colaborativos y basados en contenido completamente puros y demuestran que los métodos híbridos pueden proporcionar recomendaciones más precisas que los enfoques puros.

2.4. Explicaciones

Un RS tiene como objetivo principal proporcionar recomendaciones personalizadas para los usuarios, pero opcionalmente algunos son capaces de generar explicaciones del porqué de la recomendación. Según [FZ11] las explicaciones en RS se caracterizan por tener dos propiedades. Primero, una explicación debe dar información sobre la recomendación. Segundo, deben estar alineadas con los objetivos del diseño del sistema. La capacidad de un sistema para explicar sus recomendaciones es un aspecto importante de su utilidad y puede contribuir significativamente a que los usuarios acepten sus sugerencias [TM12].

El concepto de explicación es frecuentemente explotado en la comunicación humana y en las tareas de razonamiento. En consecuencia, la investigación dentro de la intelligen-

cia artificial, y en particular, en el desarrollo de sistemas que imiten el comportamiento humano, ha mostrado gran interés en la naturaleza de las explicaciones. En [BCS98] se distingue entre explicaciones funcionales, causales, intencionales y científicas. Las explicaciones funcionales tratan sobre las funciones de los sistemas. Las explicaciones causales proporcionan relaciones causales entre eventos. Las explicaciones intencionales dan razones para el comportamiento humano. Las explicaciones científicas se explotan para expresar relaciones entre los conceptos formulados en diversos campos científicos y se basan típicamente en teorías refutables. Desafortunadamente, no hay una teoría unificada aceptada que describa el concepto de explicación. En consecuencia, no está claro cómo diseñar un método general para generar explicaciones. Al enfrentar tales desafíos fundamentales, surge la pregunta de por qué los sistemas de recomendación deberían tratar con las explicaciones. La respuesta está relacionada con las dos partes que proporcionan y reciben recomendaciones. Por ejemplo, un agente de ventas puede estar interesado en promocionar productos particulares, mientras que un agente de compras está preocupado por tomar la decisión de compra correcta. Las explicaciones son piezas importantes de información que pueden ser explotadas por ambos agentes a lo largo del proceso de comunicación para aumentar su rendimiento. Diferentes agentes formularán explicaciones con diferentes intenciones, por ejemplo, un agente de compras busca ofertas y explicaciones que justifiquen las decisiones, mientras que un agente de ventas intenta mejorar las ganancias al proporcionar argumentos convincentes al agente de compras. Pero en definitiva, lo importante es considerar el fenómeno de las explicaciones desde un punto de vista pragmático.

A pesar de la diversidad de propuestas para caracterizar el concepto de explicación, casi todas las fuentes están de acuerdo en que una explicación es una información intercambiada en un proceso de comunicación. En el contexto de los sistemas de recomendación, estas piezas de información complementan una recomendación con diferentes objetivos. La Figura 2.1 muestra los distintos objetivos que puede perseguir una explicación, para beneficio del usuario o del sistema.

Siguiendo una visión pragmática, como se propone en [TM07], se pueden identificar diferentes objetivos para proporcionar explicaciones en un proceso de recomendación. A continuación se explican algunos de los posibles objetivos a perseguir.

Transparencia. Las explicaciones que respaldan la transparencia de las recomendaciones tienen como objetivo proporcionar información para que el usuario pueda compren-

Objetivo	Definición
Transparencia	Explicar cómo funciona el sistema.
Validez	Ayudar a los usuarios a verificar la calidad de una recomendación.
Escrutabilidad	Permitir a los usuarios corregir una recomendación incorrecta.
Confianza	Aumentar la confianza de los usuarios en el sistema.
Persuasión	Convencer a los usuarios a probar o adquirir un producto o servicio.
Eficacia	Ayudar a los usuarios a tomar buenas decisiones.
Eficiencia	Ayudar a los usuarios a tomar decisiones más rápido.
Satisfacción	Aumentar la facilidad de uso o disfrute.
Pertinencia	Refinar el perfil del usuario mejorando la recomendación.
Comprensibilidad	Complementar el conocimiento del usuario con el del recomendador.
Educación	Ayudar a los usuarios a entender el dominio.

Figura 2.1: Objetivos de una explicación.

der el razonamiento utilizado para generar una recomendación específica. En particular, la explicación puede proporcionar información sobre por qué se prefirió un elemento sobre otro. Por ejemplo, considere el caso en el que se pregunta porqué un recomendador de películas asume que al usuario gustan los westerns, cuando en realidad no es así. Las explicaciones de transparencia pueden indicar, por ejemplo, que el usuario compró canciones country y esta información se está explotando para recomendar películas western, lo que le da la oportunidad de cambiar suposiciones falsas.

Validez. Se pueden generar explicaciones para permitir que un usuario verifique la validez de una recomendación. Por ejemplo, “te recomiendo un automóvil tipo *van* porque en tu familia hay cuatro niños y el vehículo tiene una cantidad de siete asientos”. Debido a la cantidad de hijos de la familia, no es posible recomendar un tipo de automóvil que solo posea cuatro asientos. La capacidad de verificar la validez no está necesariamente relacionada con la transparencia. Por ejemplo, una red neuronal puede haber decidido que un producto es casi perfecto para un conjunto de requisitos de los clientes. La transparencia en el proceso de cálculo, al revelar cómo la red neuronal calculó la recomendación, no ayudará a un cliente a validar la recomendación. Sin embargo, mostrará una comparación de las características requeridas y ofrecidas del producto que permite al cliente validar la calidad de la recomendación del producto.

Escrutabilidad. Significa que los usuarios pueden informar que el sistema se ha equivocado. Esta capacidad está estrechamente relacionada con la transparencia, permitiendo

a los usuarios corregir el razonamiento realizado por el sistema. Los usuarios confiarán en un sistema que proporciona transparencia en el funcionamiento al ofrecer explicaciones que contienen un mecanismo para gestionar los errores dentro de las recomendaciones.

Confianza. Siguiendo a [GKK03], la creación de confianza se puede ver como un mecanismo para reducir la complejidad de la toma de decisiones humanas en situaciones inciertas. Las explicaciones que buscan generar confianza en las recomendaciones reducen la incertidumbre acerca de la calidad de una recomendación, por ejemplo, “el medicamento *Pacural* curó a miles de personas con su enfermedad; por lo tanto, esta droga también te ayudará”.

Persuasión. La tecnología computacional [Fog99] se considera persuasiva si el sistema está diseñado intencionalmente para cambiar la actitud o el comportamiento de una persona de una manera predeterminada. En este sentido, las explicaciones persuasivas de las recomendaciones apuntan a cambiar el comportamiento, por ejemplo de compra, del usuario. De esta manera, un recomendador puede detenerse intencionalmente en los aspectos positivos de un producto y guardar silencio sobre varios aspectos negativos.

Eficacia. En el contexto de los sistemas de recomendación, el término eficacia se refiere al apoyo que recibe un usuario para tomar decisiones de alta calidad. Las explicaciones para mejorar la efectividad generalmente ayudan al cliente a descubrir sus preferencias y tomar decisiones que maximizan la satisfacción con respecto al producto seleccionado. Los recomendadores efectivos ayudan a los usuarios a tomar mejores decisiones.

Eficiencia. En el contexto de los sistemas de recomendación, el término eficiencia se refiere a la capacidad de un sistema para ayudar a los usuarios a reducir el esfuerzo de toma de decisiones. Por lo tanto, las explicaciones que buscan aumentar la eficiencia generalmente intentan reducir el tiempo necesario para la toma de decisiones. Sin embargo, una medida de la eficiencia también podría ser el esfuerzo cognitivo percibido, que podría ser diferente de la eficiencia en función del tiempo necesario para realizar la recomendación y seleccionar un producto.

Satisfacción. Las explicaciones pueden intentar mejorar la satisfacción general derivada del uso de un sistema de recomendación. Este objetivo no puede estar vinculado a ningún otro objetivo de explicación, como la persuasión. La motivación detrás de este objetivo puede ser múltiple, por ejemplo, para aumentar la tasa de retorno de los clientes.

Pertinencia. Las explicaciones se pueden utilizar para explicar por qué, en algunos casos, es necesario que el usuario proporcione información adicional al sistema. Esto es

cierto en el caso de los sistemas de recomendación conversacionales, donde los usuarios interactúan con el sistema, durante el proceso de recomendación, para refinar sus requerimientos de información [JZFF10].

Comprensibilidad. Los recomendadores nunca pueden estar seguros sobre el conocimiento de sus usuarios. Las explicaciones dirigidas a la comprensión apoyan al usuario a relacionar los conceptos conocidos del usuario con los conceptos empleados por el recomendador.

Educación. Las explicaciones pueden tener como objetivo educar a los usuarios para ayudarles a comprender mejor el dominio del producto. El conocimiento profundo sobre el dominio ayuda a los clientes a repensar sus preferencias y evaluar los pros y los contras de las diferentes soluciones. Eventualmente, a medida que los clientes se informan más, pueden tomar decisiones de compra más inteligentes.

Los objetivos antes mencionados para generar explicaciones pueden estar interrelacionados. Por ejemplo, una explicación generada para mejorar la transparencia de una recomendación, puede tener un efecto positivo en la confianza. Por el contrario, las explicaciones dirigidas a la persuasión pueden resultar en una pérdida de confianza. Para evaluar la utilidad de las explicaciones, todos los efectos en los diversos objetivos de comunicación del proceso de recomendación deben ser verificados. Como se señaló anteriormente, las explicaciones se utilizan en un proceso de comunicación. Por lo tanto, la idoneidad de una explicación depende de los objetivos tanto del remitente como del receptor. Como consecuencia, la calidad de las explicaciones se puede mejorar modelando el agente receptor. Por ejemplo, para que las explicaciones sean comprensibles, es necesario tener información sobre el nivel de conocimiento del receptor. En general, cuanto mejor sea el modelo de un agente receptor, más efectivos serán los argumentos generados por un agente persuasivo.

2.5. Evaluación de los resultados de un RS

Desde que comenzó la investigación de RS, la evaluación de predicciones y recomendaciones se ha vuelto importante [SKKR00b, HKTR04]. La investigación en el campo de RS requiere medidas de calidad y métricas de evaluación [GS09] para conocer la calidad de las técnicas, métodos y algoritmos para predicciones y recomendaciones. Las métricas de evaluación [HKTR04, HdOG08] y los marcos de evaluación [BHOB11, HKBR17]

facilitan la comparación de varias soluciones para el mismo problema y la selección de diferentes líneas de investigación prometedoras que generan mejores resultados. Debido a las medidas de evaluación, las recomendaciones de los RS se han probado y mejorado gradualmente [CCFF11].

Un conjunto representativo de medidas de evaluación existentes tiene formulaciones estándar y además se han generado bases de datos públicas para RS. Estos dos avances han facilitado las comparaciones de calidad para los nuevos métodos de recomendación propuestos y los métodos publicados anteriormente, por lo tanto, los métodos y algoritmos investigados en el área de RS han progresado continuamente.

A continuación se presenta una selección representativa de las medidas de calidad de evaluación de RS más utilizadas.

2.5.1. Calidad de las predicciones

Para medir la precisión de los resultados de un RS, es habitual utilizar el cálculo de algunas de las métricas de error de predicción más comunes, entre las que se encuentran el error absoluto medio (MAE, del inglés *Mean Absolute Error*) y sus métricas relacionadas destacadas: error cuadrático medio, raíz del error cuadrático medio (RMSE, del inglés *Root Mean Squared Error*), y el error absoluto medio normalizado.

Se define U como el conjunto de usuarios del RS, I como el conjunto de elementos del RS, $r_{u,i}$ la calificación del usuario u para el elemento i , \emptyset la falta de calificación ($r_{u,i} = \emptyset$ significa que el usuario u no calificó al elemento i), y $p_{u,i}$ la predicción del elemento i para el usuario u . Sea $O_u = \{i \in I | p_{u,i} \neq \emptyset \wedge r_{u,i} \neq \emptyset\}$, el conjunto de elementos calificados por el usuario u que tienen valores de predicción. Se observa que la diferencia absoluta entre la predicción y el valor real, $|p_{u,i} - r_{u,i}|$, informa sobre el error en la predicción. Las métricas MAE y RMSE se definen promediando los errores de predicción para todos los elementos y usuarios como sigue:

$$MAE = \frac{1}{\#U} \sum_{u \in U} \left(\frac{1}{\#O_u} \sum_{i \in O_u} |p_{u,i} - r_{u,i}| \right) \quad (2.16)$$

$$RMSE = \frac{1}{\#U} \sum_{u \in U} \sqrt{\frac{1}{\#O_u} \sum_{i \in O_u} (p_{u,i} - r_{u,i})^2} \quad (2.17)$$

2.5.2. Calidad de un conjunto de recomendaciones

La confianza de los usuarios para un determinado RS no depende directamente de la precisión para el conjunto de posibles predicciones. Un usuario gana confianza en el RS cuando este está de acuerdo con un conjunto reducido de recomendaciones hechas por el RS. Por lo general, cuando se realiza un experimento “*offline*”, la respuesta del recomendador puede clasificarse en alguna de las clases que se muestran en la Figura 2.2.

		Predicción	
		+	-
Observación	+	TP	FN
	-	FP	TN

Figura 2.2: Clasificación de la respuesta de un RS.

Utilizamos TP para representar a los verdaderos positivos (del inglés, *True Positive*), FN a los falsos negativos (del inglés, *False Negative*), FP a los falsos positivos (del inglés, *False Positive*), y TN a los verdaderos negativos (del inglés, *True Negative*).

En esta sección, a partir de las posibles respuestas de un RS, se definen las siguientes medidas de calidad de recomendación más utilizadas: (1) precisión (del inglés *precision*), que indica la proporción de elementos relevantes recomendados del número total de elementos recomendados, (2) cobertura (del inglés *cobertura*), que indica la proporción de elementos relevantes recomendados del número de elementos relevantes, y (3) exactitud (o tasa de aciertos, del inglés *accuracy*), que indica la fracción de los recursos predichos como positivos o negativos para los que la predicción fue correcta, (4) F-Calificación (del inglés *F-Score*), que es la media armónica de precisión y cobertura.

$$\text{Precisión} = \frac{TP}{TP + FP} \quad (2.18)$$

$$\text{Cobertura} = \frac{TP}{TP + FN} \quad (2.19)$$

$$\text{Exactitud} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.20)$$

$$F_\beta - \text{calificación} = \frac{(1 + \beta^2) \times (\text{precisión} \times \text{cobertura})}{(\beta^2 \times \text{precisión} + \text{cobertura})} \quad (2.21)$$

Por lo general, es esperable una compensación entre estas cantidades, mientras que permitir listas de recomendaciones más largas generalmente mejora la cobertura, también es probable que reduzca la precisión. En aplicaciones donde las recomendaciones que se presentan al usuario están ordenadas, la medida de interés más útil es *precisión a un rango n ($p@n$)*.

2.5.3. Otras métricas

Existe otro conjunto de métricas, no tan populares o estandarizadas para su cálculo como las anteriormente mencionadas [KB17]. De este conjunto hay diferentes aproximaciones que se han utilizado para medir distintas características de las sugerencias de un sistema de recomendación. A continuación se listan algunas de ellas.

Diversidad. Se define generalmente como una medida de disimilitud entre los elementos de la lista de recomendaciones con respecto a una métrica de similitud. Esta diversidad “dentro de la lista” ha sido presentada y utilizada en varias formas [ZCM02, ZMKL05, ZH08, KP17, NHTK18].

Serendipia. Es la cualidad de ser inesperado y útil [KWV16]. Para medir la serendipia de una lista de recomendaciones, por ejemplo, se podría medir la utilidad de los elementos (por lo general considerando solo la precisión) penalizando los elementos más populares [MMO07, GDBJ10].

Novedad. Es una medida de calidad de recomendación que parece estar estrechamente relacionada con la serendipia [MRK06] y hasta pueden superponerse [Zha13]. Para distinguir mejor estos dos objetivos, es cada vez más común definir la novedad de un elemento de manera independiente del usuario. Por lo general, la novedad de un elemento se calcula por la inversa de su popularidad (por ejemplo, medida por la cantidad de calificaciones que ha recibido), es más probable que los elementos con poca popularidad sean nuevos para el usuario [ZKL⁺10].

Estabilidad. La estabilidad en las predicciones y recomendaciones influye en la confianza de los usuarios hacia el RS. Un RS es estable si las predicciones que proporciona no cambian fuertemente en un corto período de tiempo. En [AZ12] se propone una medida de calidad de la estabilidad.

2.6. Resumen

En este capítulo se introdujeron nociones correspondientes a los sistemas de recomendación. Se los definió como herramientas o técnicas capaces de brindar sugerencias al usuario, potencialmente abrumado por la cantidad de opciones disponibles.

En la Sección 2.1 se describieron las distintas funciones que puede desempeñar un RS. Estas funciones son diferentes si se consideran el lado del proveedor de un servicio o el del usuario. A un proveedor de servicio le pueden interesar funciones de un RS como por ejemplo: aumentar el número de artículos vendidos, vender artículos más diversos, aumentar la satisfacción, o fidelidad, del usuario, o entender mejor lo que quiere el usuario. En cambio a un usuario le podría interesar un RS que lo ayude a cumplir sus tareas y objetivos. Entre las opciones de funciones más populares que un RS le puede ofrecer a un usuario se pueden mencionar: encontrar algunos, o todos, los elementos que le interesen, recomendar elementos en secuencia o en paquete, permitirle solo navegar entre las opciones disponibles, admitirle mejorar su perfil, ayudar, o influir, en otros o simplemente expresarse. En consecuencia, un RS debe equilibrar las necesidades de estas dos partes intervinientes y ofrecer un servicio que sea valioso para ambos.

Como una clasificación general, en la Sección 2.2, se mostró que los datos utilizados por los RS se refieren a tres tipos de objetos: elementos, usuarios y transacciones. Estas últimas representan las relaciones entre usuarios y elementos. Los RS son sistemas que recopilan activamente varios tipos de datos para elaborar sus recomendaciones. Estos datos son principalmente sobre los elementos que se sugieren y los usuarios que recibirán estas recomendaciones. Pero, estos datos pueden ser explotados o no, dependiendo de la técnica de recomendación utilizada. Se considera transacción a cada una de las interacciones que ocurren entre el usuario y el sistema. Las transacciones pueden ser implícitas o explícitas. Para el primer caso, el sistema infiere la opinión del usuario de los elementos en función de las acciones que realiza. Para el caso de las explícitas, el RS le permite al usuario que realice una acción particular para emitir su opinión para un determinado elemento.

El usuario podría expresar su opinión a través de la calificación del elemento o de la asignación de etiquetas para evaluarlo.

En la Sección 2.3 se presentan las técnicas más populares para definir un RS. Tradicionalmente, se utilizan dos enfoques principales para generar recomendaciones: las recomendaciones basadas en contenido y las recomendaciones basadas en los métodos de filtrado colaborativo. Los recomendadores basados en contenido son impulsados por la premisa de que las preferencias de los usuarios tienden a persistir a través del tiempo. Estos recomendadores frecuentemente utilizan técnicas de ML para generar un perfil del usuario activo, que normalmente se almacena como una lista de elementos calificados. Para determinar si un elemento nuevo es una recomendación potencialmente buena, los sistemas de recomendación basados en contenido se basan en medidas de similitud entre los nuevos elementos y los elementos calificados ya almacenados como parte del modelo de usuario. Los sistemas de recomendación basados en filtrado colaborativo se basan en la suposición de que las preferencias de los usuarios están correlacionadas. Estos sistemas mantienen un conjunto de perfiles de los usuarios asociados a los elementos que los usuarios han calificado en el pasado. Para un usuario activo determinado, los sistemas de recomendación colaborativos encuentran otros usuarios similares cuyas calificaciones de elementos se relacionan estrechamente con el usuario actual. Nuevos elementos no calificados por el usuario activo pueden ser presentados como sugerencias si los usuarios similares a este los han calificado altamente.

Una combinación de filtrado colaborativo y recomendación basada en contenido, o de cualquier otro tipo de RS, da lugar a los sistemas de recomendación híbridos. Los recomendadores híbridos suelen generar un modelo del usuario activo mediante la supervisión del comportamiento del usuario o mediante el análisis de intereses declarados o por retroalimentación de los usuarios. El modelo generado del usuario normalmente se combina con las necesidades de información del usuario con el fin de buscar sugerencias. Además, el sistema mantiene un grupo de perfiles de otros usuarios, haciendo posible la aplicación de filtrado colaborativo para perfeccionar el conjunto seleccionado de recomendaciones.

Por otra parte, en la Sección 2.4, se hace referencia a las explicaciones. Si bien generar explicaciones no son el objetivo principal de un RS, que un sistema cuente con esta capacidad puede ser de una gran utilidad y puede contribuir significativamente a que los usuarios acepten sus sugerencias. Las explicaciones son piezas importantes de información que pueden ser explotadas, tanto por el usuario como por el proveedor de un servicio,

durante el proceso de comunicación para aumentar su rendimiento. Un RS genera estas piezas de información con diferentes intenciones, entre los cuales se pueden destacar: generar recomendaciones transparentes, válidas, eficaces, pertinentes, persuasivas y/o de manera eficiente. También podrían permitir generar mayor confianza y satisfacción en el usuario del RS, aportándole, además, comprensibilidad y educación.

Finalmente, en la Sección 2.5, se presentan diferentes medidas para evaluar los RS, en particular la calidad de sus resultados. Estas medidas se han vuelto importantes para poder comparar la calidad de las distintas técnicas, métodos y algoritmos utilizados para realizar recomendaciones. Dentro de las métricas más populares y estandarizadas para medir la calidad de las predicciones o recomendaciones se pueden mencionar el *MAE*, la *RMSE*, la *Precisión*, la *Cobertura*, la *Exactitud* y la *F-calificación*. También existen otras métricas para poder evaluar un RS. Algunos ejemplos son la *Diversidad*, la *Novedad*, la *Estabilidad* y la *Serendipia*.

Capítulo 3

Sistemas argumentativos

Argumentación es una forma de razonamiento por la cual, para una dada afirmación, se presta atención explícita a las justificaciones presentadas y a la resolución de los posibles conflictos que podrían existir entre ellas. Mediante este tipo de razonamiento, una afirmación es aceptada o rechazada dependiendo del análisis de los argumentos tanto a su favor como en su contra. La forma en que los argumentos y las justificaciones son consideradas para una afirmación permite definir un mecanismo de razonamiento automático que contempla la posibilidad de contar con información contradictoria, incompleta e incierta [PV02, RS09].

El estudio de argumentación ha sido abordado desde diferentes enfoques: a nivel lógico puede considerarse como una forma de modelar inferencia rebatible, y a nivel dialógico puede verse como una forma de interacción entre agentes inteligentes. Esta forma de razonamiento ha sido ampliamente estudiada en disciplinas como la filosofía [Tou58, Pol87], y desde la década del '70 es posible encontrar estudios en las Ciencias de la Computación que han contribuido en gran medida a la noción de argumento. Recién a mediados de los años '80 se inició el desarrollo del área de argumentación desde un punto de vista computacional, donde los argumentos son explícitamente construidos y comparados como medios para resolver problemas en una computadora. En las últimas décadas, la argumentación ha evolucionado como un atractivo paradigma para conceptualizar el razonamiento de sentido común [CML00, BD07, BH08, AyC14]. Se han creado diferentes aproximaciones para modelar argumentación, tales como sistemas de argumentación abstracta [Dun95, Pra10], sistemas que utilizan lógica clásica [BH01, BH09], o sistemas argumentativos basados en reglas [PS97, GS04, DKT06, AK07]. Por otra parte, este tipo

de razonamiento es particularmente atractivo para la toma de decisiones, y dentro del área de Inteligencia Artificial existe especial interés en abordar este tipo de problemas. En el último tiempo, el campo de aplicación de la argumentación se ha expandido velozmente, en gran parte debido a los avances teóricos, pero también gracias a la demostración exitosa de su uso práctico en un gran número de dominios de aplicación, tales como el razonamiento legal [PS02, Ver03b, dCPLM⁺17], planificación [SGC04, CSCC13, PG18], la ingeniería del conocimiento [CRL00], los sistemas para la toma de decisiones y negociación [BH09, AV12, GGGL16, FTG⁺17, BTG19], los sistemas multi-agente [AMP02, RA06, GGS10, CI15, GTGS18], el e-government [ABM05, CMGE16], el procesamiento de lenguaje natural [CM04b, WvEH16], entre muchos otros.

En lo que resta de este capítulo se introduce una serie de nociones básicas de argumentación. En primer lugar se presenta la estructura conceptual propuesta en [PV02], la cual provee una caracterización de los diferentes elementos presentes en los sistemas argumentativos. Luego, se muestra cómo estos elementos se ven reflejados en los diferentes pasos involucrados en el proceso de razonamiento argumentativo. Seguidamente, se presenta la programación en lógica rebatible [GS04], una aproximación para la formalización de sistemas argumentativos. Se introducen sus principales características, haciendo referencia a cómo instancian los diferentes elementos de la estructura conceptual antes mencionada. Finalmente se describe brevemente, un formalismo basado en argumentación con soporte de base de datos, el cual integra argumentación rebatible con bases de datos que pueden ser actualizadas por otras aplicaciones externas, lo que permitirá la ejecución de procesos de argumentación basados en fuentes de datos externas masivas.

3.1. Una caracterización de los sistemas argumentativos

Los autores en [PV02] proponen una estructura conceptual que describe los principales elementos con los que se puede caracterizar a un sistema argumentativo. Se identifican cinco elementos, algunos de los cuales pueden hallarse presentes de manera implícita, estos son: un *lenguaje de representación*; una definición de la noción de *argumento*; la identificación de *conflictos* entre argumentos; la obtención de las *derrotas* entre argumentos; y, finalmente, la determinación del *estado de aceptabilidad* de los argumentos, la cual puede ser utilizada para definir una noción de consecuencia lógica del sistema.

- *Lenguaje de Representación*: Los sistemas argumentativos son construidos a partir de un lenguaje de representación que es utilizado para modelar la información acerca del dominio sobre el cual se efectuará el razonamiento. Asociada a este lenguaje de representación es definida una noción de consecuencia lógica, la cual es utilizada para definir la noción de argumento. Algunos sistemas utilizan un lenguaje de representación particular (por ejemplo, [GS04]), mientras que otros sistemas dejan la lógica subyacente parcialmente especificada o completamente sin especificar (por ejemplo, [Dun95]). En particular, esta última clase de sistemas puede ser instanciada utilizando diferentes lógicas alternativas, caracterizando así la noción de *marco argumentativo*.
- *Noción de Argumento*: Esta noción corresponde a una prueba en la lógica subyacente. En cuanto al formato para la representación de los argumentos, en la literatura se destacan principalmente tres alternativas. La primera de ellas define a los argumentos como árboles de inferencia basados en premisas. Otra alternativa consiste en la construcción de argumentos a partir de una secuencia de prueba o derivación. Finalmente, algunos sistemas simplemente definen un argumento como un par *premisas-conclusión*, dejando implícito el hecho de que el operador de consecuencia asociado a la lógica subyacente permite obtener una prueba de la conclusión a partir de las premisas. Cabe destacar que algunos formalismos argumentativos se abstraen de la forma en que los argumentos son construidos, dejando su estructura interna completamente sin especificar. Tal es el caso del marco argumentativo propuesto por Dung en [Dun95], el cual trata la noción de argumento como una primitiva, centrándose exclusivamente en las interacciones entre los argumentos.
- *Conflictos entre Argumentos*: El uso de argumentación como mecanismo de razonamiento presupone, de alguna manera, la existencia de conflictos entre argumentos. Esta noción de conflicto, también llamada ataque o contra-argumentación, puede tener origen en diferentes situaciones. La literatura de argumentación usualmente distingue tres tipos de conflictos. El primero de ellos, conocido como *ataque por refutación* (en inglés *rebutting attack*), ocurre cuando dos argumentos poseen conclusiones contradictorias. Otro tipo de conflicto, llamado *ataque a una suposición* (en inglés *assumption attack*) surge cuando un argumento realiza una suposición de no existencia de prueba (es decir, se asume que no es posible obtener una prueba para una determinada fórmula del lenguaje lógico subyacente) y otro argumento

prueba la conclusión que el primero asumió como no demostrable. Existe en la literatura otra forma de conflicto que se encuentra relacionada con el ataque a una suposición: el *ataque a una premisa* (en inglés *undermining attack*). Este tipo de ataque está dirigido sobre una premisa de un argumento para la cual no existe una prueba fehaciente (es decir, una premisa que no forma parte de los hechos de la base de conocimiento). Dados los tipos de conflicto identificados en la literatura, se puede observar que el ataque por refutación es simétrico, mientras que los otros tipos de ataque son asimétricos. Generalmente, los tipos de conflicto arriba mencionados son inferidos a partir de las características del lenguaje de representación. Es por esto que en sistemas abstractos como el propuesto en [Dun95], en los que hay abstracción del lenguaje de representación y de la forma en que los argumentos son construidos, se asume que los conflictos entre argumentos se encuentran establecidos explícitamente de antemano.

- *Derrotas entre Argumentos*: La noción de conflicto o ataque entre argumentos no implica evaluación alguna. Luego, es necesaria la definición de una noción de derrota, la cual conlleva la evaluación de pares de argumentos conflictivos para determinar qué ataques son exitosos. De esta manera, se define una relación binaria de derrota que captura la intuición de “ataca y no es más débil” (en su versión débil) o “ataca y es más fuerte” (en su versión fuerte). En particular, cada uno de los tipos de ataque identificados en el tercer elemento de esta estructura conceptual posee su contrapartida como derrota. En la literatura de argumentación existen diferentes alternativas para la resolución de conflictos y la obtención de derrotas. Una de las más populares en el área de Inteligencia Artificial es la utilización de un criterio de comparación entre argumentos llamado *especificidad*, el cual prefiere argumentos basados en información más específica, valiéndose únicamente de la estructura lógica de los argumentos y abstrayéndose de la información del dominio de aplicación. Sin embargo, en trabajos como [Vre91, Pol95, PS96] los autores establecen que el criterio de especificidad no corresponde a un principio general del razonamiento de sentido común, sino que simplemente es uno de muchos estándares que pueden o no ser utilizados. Más aún, propuestas como [Kon88, Vre91] sugieren que la información acerca del dominio suele ser la herramienta principal para evaluar los argumentos del sistema. Es por esto que sistemas argumentativos como [GS04] se encuentran parametrizados con respecto al criterio de comparación entre argumentos, siendo

este provisto por el usuario y posiblemente hallándose relacionado con el dominio de aplicación.

- *Estado de Aceptabilidad de los Argumentos*: El objetivo de un sistema argumentativo consiste en determinar qué argumentos son finalmente aceptados. Para esto es necesario considerar las derrotas entre los argumentos del sistema. Sin embargo, no resulta suficiente considerar aisladamente la relación de derrota para cada par de argumentos conflictivos, ya que esto permite identificar únicamente la fuerza relativa de dos argumentos individuales. En contraste, para determinar el estado de aceptación de un argumento, es necesario considerar las interacciones de todos los argumentos disponibles en el sistema. Es decir, dado un argumento, deben considerarse sus derrotadores, los derrotadores de sus derrotadores, y así sucesivamente. Por ejemplo, consideremos un sistema argumentativo con los argumentos \mathcal{A} , \mathcal{B} y \mathcal{C} , donde el argumento \mathcal{A} derrota al argumento \mathcal{B} y, por otra parte, el argumento \mathcal{C} derrota al argumento \mathcal{A} . En este caso, el argumento \mathcal{C} provee una defensa para \mathcal{B} ante la derrota de \mathcal{A} . Esta situación es capturada en la literatura bajo la noción de *reinstalación* (en inglés *reinstatement*). Luego, los argumentos aceptados del sistema serán \mathcal{C} y \mathcal{A} . La definición de un mecanismo para el cómputo del estado de aceptabilidad de los argumentos permite obtener las inferencias de un sistema argumentativo. Típicamente, este mecanismo divide a los argumentos en al menos dos categorías: argumentos *aceptados* y argumentos *rechazados*. Algunas veces también se considera una tercera categoría, correspondiente a argumentos en *discusión*; es decir, argumentos cuyo estado de aceptación es *indeciso* dado que no se encuentran aceptados ni rechazados. En la literatura existen diversas propuestas para el cálculo de aceptabilidad de los argumentos de un sistema argumentativo. En particular, estas alternativas pueden definirse siguiendo una aproximación declarativa o una aproximación procedural. La aproximación declarativa simplemente establece ciertas condiciones que un conjunto de argumentos aceptables debe cumplir, sin especificar cómo se computarán tales conjuntos. Por otra parte, las aproximaciones procedurales brindan un procedimiento para determinar si un argumento en particular se encuentra aceptado o no. En general, este tipo de procedimientos simula el juego argumentativo que ocurre en una discusión, y es modelado a través de árboles de argumentos conocidos como *árboles de dialéctica*. De esta manera, se puede considerar que la aproximación declarativa define la semántica de un siste-

ma argumentativo, mientras que la aproximación procedural constituye su teoría de prueba.

A partir de la caracterización propuesta en [PV02] se puede observar cómo los elementos de un sistema argumentativo intervienen en las diferentes etapas del proceso de razonamiento argumentativo. La figura 3.1 provee una esquematización de los cinco pasos del proceso argumentativo:

1. Construcción de argumentos
2. Construcción de interacciones entre argumentos.
3. Comparación de argumentos.
4. Cómputo de aceptabilidad de argumentos.
5. Obtención de conclusiones del sistema.

A continuación se provee una descripción de cada uno de estos pasos.

1. Los argumentos son construidos a partir de una base de conocimiento expresada en el lenguaje de representación subyacente. Dependiendo del lenguaje y de las reglas para la construcción de argumentos, es posible obtener diferentes tipos de argumentos.
2. Los argumentos obtenidos en el paso anterior no pueden ser considerados aisladamente. En efecto, la mayoría de los argumentos de un sistema argumentativo interactúan de alguna manera. Por ejemplo, se puede considerar una relación de conflicto o ataque entre argumento.
3. La idea detrás de este paso consiste en asignar un peso a cada argumento. Luego, mediante la consideración de estos pesos es posible comparar los argumentos. Existen en la literatura diferentes criterios para asignar pesos a los argumentos. Por ejemplo, puede utilizarse el criterio de especificidad mencionado anteriormente. Otra alternativa, utilizada en [AC02], consiste en considerar prioridades explícitas e implícitas.

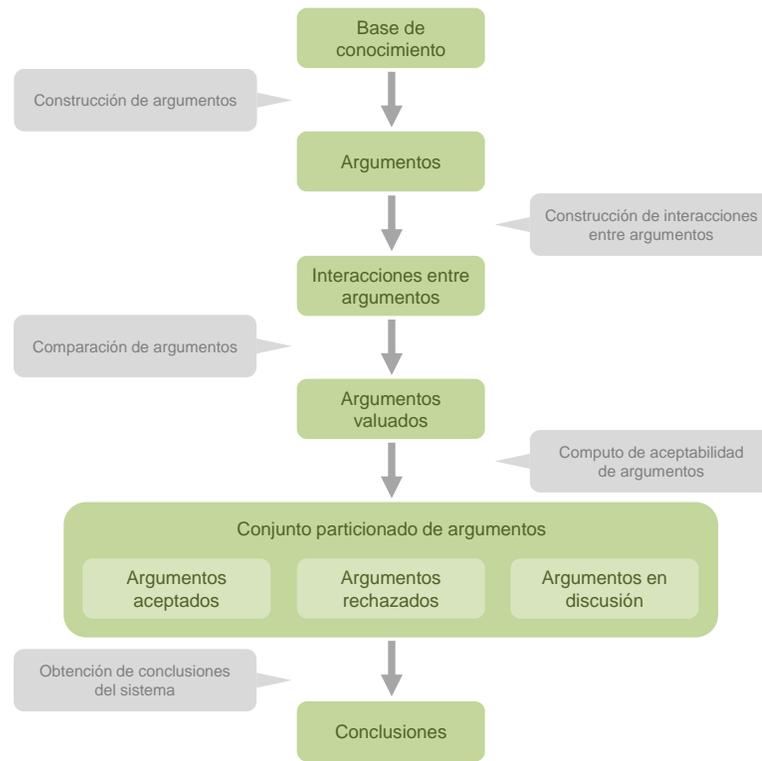


Figura 3.1: Los cinco pasos del proceso de razonamiento argumentativo.

4. El objetivo de un sistema argumentativo es identificar el conjunto de argumentos que deben ser aceptados, los cuales constituyen los “mejores” argumentos del sistema. El cálculo del estado de aceptabilidad de los argumentos contempla las interacciones entre estos, así como también los valores asignados por el criterio de comparación.
5. El estado de aceptabilidad de los argumentos determina las conclusiones del sistema. De esta manera, es posible identificar un conjunto de inferencias a partir de la selección de los argumentos aceptados. Luego, es posible utilizar estas conclusiones para apoyar la toma de decisiones en diferentes dominios de aplicación.

3.2. Formalización de sistemas argumentativos

En esta sección se presentará un *Sistema Argumentativo Basado en Reglas*, el cual representa una aproximación para la formalización de sistemas argumentativos. Como se mencionó anteriormente, otra aproximación posible a la hora de formalizar un sistema

argumentativo, son los sistemas argumentativos abstractos. Estos sistemas se abstraen de la mayoría de los elementos de la estructura conceptual introducida en la Sección 3.1, para concentrarse en el análisis del estado de aceptación de los argumentos del sistema. En contraste, los sistemas argumentativos basados en reglas definen todos los elementos identificados en la estructura conceptual. Es por esto que, usualmente, la literatura clasifica a los sistemas argumentativos basados en reglas como sistemas argumentativos *concretos*, a diferencia de los sistemas argumentativos *abstractos*, también conocidos como *marcos argumentativos*.

Los sistemas argumentativos basados en reglas cuentan con una base de conocimiento que permite almacenar información expresada en un lenguaje lógico de representación. En particular, existe un conjunto de reglas de inferencia con las cuales, a partir de cierta información (antecedente) es posible inferir nueva información (consecuente). Estas reglas son almacenadas en la base de conocimiento en conjunto con otra información en forma de hechos o presuposiciones, representando evidencia que se obtiene del entorno. A partir de esta evidencia es posible utilizar las reglas de inferencia para construir argumentos a favor o en contra de una afirmación. Luego, mediante un análisis exhaustivo, se evalúan los argumentos contruidos para decidir cuáles de ellos son aceptados y, de esta manera, concluir si a partir de la información existente puede garantizarse o no dicha afirmación.

Cabe destacar que los sistemas argumentativos basados en reglas son no monótonos, dado que la incorporación de nueva información al sistema puede generar nuevos argumentos que contradigan a otros ya existentes y, por lo tanto, invalidar afirmaciones que antes se hallaban garantizadas. Es por esto que los sistemas argumentativos basados en reglas permiten representar conocimiento de sentido común y proveen una forma de razonamiento atractiva para aplicaciones de Inteligencia Artificial. En particular, este tipo de sistemas ha resultado de especial interés en sistemas multi-agente, dado que proveen un mecanismo de razonamiento sofisticado para los agentes involucrados en este dominio [RA06, GRTS07].

A continuación se introducirá el formalismo de programación en lógica rebatible propuesto en [GS04], el cual constituye un sistemas argumentativo basado en reglas. A partir de este formalismo se ilustrarán las diferentes características de los sistemas argumentativos concretos. Asimismo, este formalismo resulta de especial interés, ya que será utilizado para los desarrollos presentados de esta tesis.

3.2.1. Programación en lógica rebatible (DeLP)

En esta sección se presentará el formalismo de *Programación en Lógica Rebatible* (DeLP) (por su nombre en inglés *Defeasible Logic Programming*) introducido en [GS04]. Este formalismo es un sistema argumentativo basado en reglas que combina resultados de programación en lógica y argumentación rebatible, y que ha sido aplicado exitosamente en diferentes dominios de aplicación (por ejemplo, [CCS05, RGS07, GCS08, GGS10, GGGL16, FTG⁺17]). DeLP define todos los elementos de la estructura conceptual presentada en la Sección 3.1. En particular, DeLP adopta como lenguaje lógico de representación una extensión de la programación en lógica que incorpora la negación fuerte y permite representar información estricta e información rebatible. Esta información es modelada a través de programas lógicos rebatibles, los cuales constituyen una base de conocimiento. Luego, a partir de estos programas es posible construir argumentos, identificar conflictos entre ellos (mediante el uso de negación fuerte), determinar las derrotas entre ellos (mediante el uso de un criterio de comparación) y utilizar procedimientos de prueba dialécticos para determinar cuáles son los argumentos aceptados del sistema.

Un *programa lógico rebatible* está formado por un conjunto de *hechos y reglas estrictas* y un conjunto de *reglas rebatibles*, permitiendo estas últimas la representación de información tentativa. Toda conclusión del programa deberá estar sustentada por algún *argumento* que pueda construirse utilizando las reglas y los hechos del programa. Cuando se utilicen reglas rebatibles para derivar una conclusión C , esta conclusión será tentativa y podrá ser refutada por información que la contradiga. De esta manera, un argumento que sustenta una conclusión C podrá ser atacado por otros *contra-argumentos* que lo contradigan y que posiblemente resulten ser *derrotadores* para el mismo. Similarmente, estos derrotadores podrán a su vez ser atacados, dando así lugar a nuevos derrotadores. Entonces, para decidir cuándo una conclusión C puede aceptarse a partir de un programa lógico rebatible, se realizará un análisis dialéctico considerando los argumentos a su favor y en su contra. Esto conducirá a la construcción de una estructura arbórea de derrotadores conocida como *árbol de dialéctica*. Finalmente, una conclusión C se hallará *garantizada* si existe un argumento para C que sobreviva a todas las derrotas que recibe en su árbol de dialéctica asociado.

A continuación se introducirán conceptos preliminares de la programación en lógica que son utilizados por DeLP. Seguidamente, se indicará cómo DeLP instancia los elementos de la estructura conceptual presentada en la Sección 3.1. Concretamente, se introducirá

la sintaxis del lenguaje de representación de los programas lógicos rebatibles, se indicará cómo se efectúa la construcción de argumentos, la identificación de conflictos entre argumentos y la resolución de los mismos para obtener las derrotas correspondientes y, finalmente, se presentará un procedimiento de prueba dialéctico que permite inferir las conclusiones garantizadas a partir de un programa DeLP.

Conceptos Preliminares

En esta sección se introducirán conceptos y terminología estándar de la lógica y la programación en lógica que resultan necesarios para la formalización de DeLP.

Definición 3.1 (*Signatura*)

Una signatura es una tupla $\sigma = \langle \mathcal{V}, Func, Pred \rangle$, donde \mathcal{V} , $Func$ y $Pred$ son conjuntos finitos que representan variables, funciones y predicados respectivamente. ■

Dada una signatura σ , una función llamada “*aridad*” le asignará a cada elemento de $Func$ y $Pred$ un número entero positivo. Si $f \in Func$ y $aridad(f) = 0$, entonces f se denominará *constante*. Por otra parte, si $p \in Pred$, y $aridad(p) = 0$, entonces p se denominará *proposición*.

Definición 3.2 (*Alfabeto*)

El alfabeto generado a partir de una signatura σ consiste del conjunto de elementos miembros de la signatura, el símbolo de negación “ \sim ”, y los símbolos de puntuación “(”, “)”, y “,”. ■

Definición 3.3 (*Término*)

Sea $\sigma = \langle \mathcal{V}, Func, Pred \rangle$ una signatura. Un término T , y respectivamente sus componentes $comp(T)$, se definen inductivamente como sigue:

- toda variable $V \in \mathcal{V}$ es un término, y $comp(V) = \{V\}$;
- toda constante $c \in Func$ (es decir, $aridad(c) = 0$) es un término, y $comp(c) = \{c\}$; y
- si $f \in Func$ tal que $aridad(f) = n$ ($n \geq 1$) y t_1, \dots, t_n son términos, entonces $f(t_1, \dots, t_n)$ también es un término.
Luego, $comp(f(t_1, \dots, t_n)) = \{f\} \cup (\bigcup_{1 \leq i \leq n} comp(f_i))$.

■

Definición 3.4 (*Término fijo*)

Sea $\sigma = \langle \mathcal{V}, Func, Pred \rangle$ una signatura y T un término. T será un término fijo si no contiene variables; es decir, si $comp(T) \cap \mathcal{V} = \emptyset$. ■

Definición 3.5 (*Átomo*)

Sea $\sigma = \langle \mathcal{V}, Func, Pred \rangle$ una signatura y sean t_1, \dots, t_n términos. Si $p \in Pred$ tal que $aridad(p) = n$, entonces $p(t_1, \dots, t_n)$ es un átomo. Diremos que un átomo fijo es un átomo $p(t_1, \dots, t_n)$ tal que todos sus términos t_1, \dots, t_n son fijos. ■

Definición 3.6 (*Literal*)

Un literal L es un átomo “ A ” o un átomo negado “ $\sim A$ ”, donde “ \sim ” representa la negación fuerte. Un literal L se dirá negativo si es un átomo negado, y positivo en caso contrario. Un literal fijo es un átomo fijo o un átomo fijo negado. ■

La definición anterior establece que los literales podrán ser átomos negados utilizando el símbolo “ \sim ” de la *negación fuerte* (en inglés *strong negation*)¹. Por una parte, este tipo de negación difiere de la negación utilizada en la lógica clásica ya que, por ejemplo, no permite anidamiento (es decir, “ $\sim\sim A$ ” no es un literal válido). Por otra parte, la negación fuerte también difiere de la negación por falla (o negación *default*) utilizada en la programación en lógica, la cual es usualmente denotada con el símbolo “not”.

Definición 3.7 (*Complemento de un Literal*)

Sea L un literal y A un átomo. El complemento de L con respecto a la negación fuerte, denotado \bar{L} , se define de la siguiente manera:

- si $L = A$, entonces $\bar{L} = \sim A$;
- si $L = \sim A$, entonces $\bar{L} = A$.

■

¹La definición y propiedades de la negación fuerte pueden hallarse en [AP93a, AP93b].

Lenguaje de representación de conocimiento

Un programa lógico rebatible estará compuesto por hechos, reglas estrictas y reglas rebatibles, cuya sintaxis se define a continuación.

Definición 3.8 (*Hecho*)

Un hecho es un literal fijo L , es decir, un átomo fijo o un átomo fijo negado. ■

Definición 3.9 (*Regla Estricta*)

Una regla estricta es un par ordenado, denotado “ $Cabeza \leftarrow Cuerpo$ ”, donde el primer elemento, $Cabeza$, es un literal fijo y el segundo elemento, $Cuerpo$, es un conjunto no vacío, finito de literales fijos. Una regla estricta con cabeza L_0 y cuerpo $\{L_1, \dots, L_n\}$ se escribirá también como: $L_0 \leftarrow L_1, \dots, L_n$ ($n > 0$). ■

Definición 3.10 (*Regla Rebatible*)

Una regla rebatible es un par ordenado, denotado “ $Cabeza \multimap Cuerpo$ ”, donde el primer elemento, $Cabeza$, es un literal fijo y el segundo elemento, $Cuerpo$, es un conjunto finito de literales fijos. Una regla rebatible con cabeza L_0 y cuerpo $\{L_1, \dots, L_n\}$ se escribirá también como: $L_0 \multimap L_1, \dots, L_n$ ($n \geq 0$). ■

Sintácticamente, el símbolo “ \multimap ” es lo único que distingue a una regla rebatible de una regla estricta. El tipo de conexión expresada por estos dos tipos de reglas es diferente. Por una parte, los hechos y reglas estrictas representan conocimiento seguro y libre de excepciones. En particular, dada una regla estricta, siempre que se crea en los literales que conforman el cuerpo se podrá creer con la misma seguridad en el literal correspondiente a la cabeza. Por otra parte, las reglas rebatibles se utilizan para representar conocimiento rebatible, es decir, información tentativa que puede utilizarse en la medida que no exista información que la contradiga. De esta manera, una regla rebatible “ $Cabeza \multimap Cuerpo$ ” expresa que “razones para creer en $Cuerpo$ proveen razones para creer en $Cabeza$ ”, pero tal relación de consecuencia es más débil que la relación estricta, y por lo tanto puede ser objetada. Teniendo en cuenta esto, conocimiento como “*todas las gallinas son aves*” se representará utilizando la regla estricta:

$$ave \leftarrow gallina$$

Mientras que para representar conocimiento tentativo se usa una regla rebatible como la siguiente:

$$vuela \rightarrow ave$$

La anterior regla rebatible expresa que: “razones para creer que es ave provee razones para creer que vuela”.

Por último, otra diferencia entre las reglas estrictas y las reglas rebatibles es que estas últimas pueden tener un conjunto vacío de literales en el cuerpo. Una regla rebatible de este tipo es denotada como “Cabeza \rightarrow ” y recibe el nombre de *presuposición* [Nut88]. De esta manera, una presuposición “ $P \rightarrow$ ” expresa que “existen razones tentativas para creer en P ”. A partir de estos elementos, es posible definir un programa lógico rebatible.

Definición 3.11 (*Programa Lógico Rebatible*)

Un programa lógico rebatible es un conjunto \mathcal{P} de hechos, reglas estrictas y reglas rebatibles. Dado un programa \mathcal{P} , se identificará con Θ al conjunto de hechos, con Ω al conjunto de reglas estrictas, y con Δ al conjunto de reglas rebatibles. Por conveniencia, también se denotará con Π al conjunto $\Theta \cup \Omega$. Por lo tanto, en ocasiones se denotará un programa lógico rebatible \mathcal{P} con el par $\mathcal{P} = (\Pi, \Delta)$, mientras que cuando se quieran distinguir los hechos y reglas estrictas del programa se utilizará la terna $\mathcal{P} = (\Theta, \Omega, \Delta)$. ■

Observación 3.1 *Podría definirse un programa lógico rebatible considerando las presuposiciones separadas en un conjunto diferente. Dado un programa \mathcal{P} , adicionalmente a los ya definidos conjuntos de hechos, reglas estrictas y reglas rebatibles, se identificará con Φ al conjunto de presuposiciones. En la literatura, por conveniencia, puede encontrarse que Δ^+ representa al conjunto $\Delta \cup \Phi$. Por lo tanto es posible encontrar programas lógicos rebatibles presentados como $\mathcal{P} = (\Pi, \Delta^+)$ así como también en otros casos $\mathcal{P} = (\Theta, \Omega, \Delta, \Phi)$.*

La definición anterior establece que los programas lógicos rebatibles utilizan únicamente literales fijos. Por este motivo, salvo que se indique lo contrario, cuando hablemos de un “literal” asumiremos que es un literal fijo. No obstante esto, en algunos ejemplos se utilizarán variables en las reglas estrictas o rebatibles, denotando esquemas de reglas. Dado un esquema de una regla R , se define $Ground(R)$ como el conjunto de todas las instancias de R con literales fijos. Una instancia fija para R es una versión de la regla tal

que las variables de R son reemplazadas por términos fijos, asumiendo que las variables con el mismo nombre corresponden al mismo elemento dentro de la regla. De esta manera, dado un programa lógico rebatible \mathcal{P} se define:

$$Ground(\mathcal{P}) = \bigcup_{R \in \mathcal{P}} Ground(R)$$

En consecuencia, dado un programa lógico rebatible \mathcal{P} con esquemas de reglas, el operador $Ground(P)$ permite obtener todas las instancias fijas de reglas asociadas a \mathcal{P} . Luego, para diferenciar las variables de los demás elementos de un programa lógico rebatible, estas serán denotadas con una letra mayúscula inicial. Para ilustrar estas nociones, consideremos el siguiente ejemplo adaptado de [GS04].

Ejemplo 3.1 Consideremos el programa lógico rebatible $\mathcal{P}_{3.1} = (\Pi_{3.1}, \Delta_{3.1})$ que expresa información acerca de aves, donde

$$\Pi_{3.1} = \left\{ \begin{array}{l} gallina(tina) \\ asustada(tina) \\ kiwi(haka) \\ \sim vuela(haka) \\ ave(X) \leftarrow gallina(X) \\ ave(X) \leftarrow kiwi(X) \end{array} \right\} \quad \Delta_{3.1} = \left\{ \begin{array}{l} vuela(X) \prec ave(X) \\ \sim vuela(X) \prec gallina(X) \\ vuela(X) \prec gallina(X), asustada(X) \end{array} \right\}$$

En este caso, los hechos $gallina(tina)$, $asustada(tina)$, $kiwi(haka)$ y $\sim vuela(haka)$ representan, respectivamente, que “Tina” es una gallina que está asustada y que “Haka” es un kiwi que no vuela. De manera similar, las reglas estrictas expresan que tanto las gallinas como los kiwis son aves. Por otra parte, las reglas rebatibles expresan las siguientes conexiones tentativas: si X es un ave, entonces existen razones para creer que X vuela; si X es una gallina, entonces existen razones para creer que X no vuela; y si X es una gallina que está asustada, entonces existen razones para creer que X vuela. En particular, se pueden obtener las siguientes instancias fijas de reglas de $\mathcal{P}_{3.1}$, entre otras:

$$\begin{array}{l} ave(tina) \leftarrow gallina(tina) \\ ave(haka) \leftarrow kiwi(haka) \\ vuela(haka) \prec ave(haka) \\ \sim vuela(tina) \prec gallina(tina) \end{array}$$

A continuación se introduce la noción de derivación rebatible, la cual define qué literales pueden ser derivados utilizando las reglas de un programa lógico rebatible.

Definición 3.12 (*Derivación Rebatible de un Literal*)

Sea $\mathcal{P} = (\Pi, \Delta)$ un programa lógico rebatible y L un literal. Una derivación rebatible para L a partir de \mathcal{P} , denotado $\mathcal{P} \sim L$, es una secuencia finita de literales fijos $L_1, \dots, L_n = L$ tal que cada literal L_i pertenece a la secuencia porque:

- (a) L_i es un hecho en Π ;
- (b) existe en Π una regla estricta R con cabeza L_i y cuerpo B_1, \dots, B_k , donde todo literal B_j del cuerpo ($1 \leq j \leq k$) es un elemento de la secuencia que precede a L_i ; o
- (c) existe en Δ una regla rebatible R con cabeza L_i y cuerpo B_1, \dots, B_k , donde todo literal B_j del cuerpo ($0 \leq j \leq k$) es un elemento de la secuencia que precede a L_i .

■

La derivación se dice rebatible porque aunque un literal L pueda ser derivado a partir de un programa lógico rebatible, puede existir en el programa información que contradiga a L . Para ilustrar esto consideremos el siguiente ejemplo.

Ejemplo 3.2 *Sea $\mathcal{P}_{3.1}$ el programa lógico rebatible del Ejemplo 3.1. A partir de este programa es posible obtener, entre otras, las siguientes derivaciones rebatibles:*

- *La secuencia $kiwi(haka), ave(haka), vuela(haka)$ es una derivación rebatible para el literal “ $vuela(haka)$ ”.*
- *La secuencia $gallina(tina), ave(tina), vuela(tina)$ es una derivación rebatible para el literal “ $vuela(tina)$ ”.*
- *La secuencia $gallina(tina), \sim vuela(tina)$ es una derivación rebatible para el literal “ $\sim vuela(tina)$ ”.*
- *La secuencia $gallina(tina), asustada(tina), vuela(tina)$ es una derivación rebatible para el literal “ $vuela(tina)$ ”.*

El Ejemplo 3.2 muestra que a partir de un programa lógico rebatible es posible obtener derivaciones rebatibles para un literal y su complemento, como en el caso de los literales “*vuela(tina)*” y “ \sim *vuela(tina)*”. Más aún, es posible obtener más de una derivación rebatible para un literal. Tal es el caso de la segunda y cuarta derivación del Ejemplo 3.2, las cuales constituyen derivaciones rebatibles para el literal “*vuela(tina)*”. Por otra parte, también pueden existir derivaciones que utilizan solo reglas rebatibles o solo reglas estrictas. En particular, este último tipo de derivación resulta de especial interés, y se encuentra caracterizada por la siguiente definición.

Definición 3.13 (*Derivación Estricta de un Literal*)

Sea $\mathcal{P} = (\Pi, \Delta)$ un programa lógico rebatible y L un literal. Una derivación es estricta para L a partir de \mathcal{P} , denotado $\mathcal{P} \vdash L$, si L es un hecho o si todas las reglas utilizadas para obtener la secuencia $L_1, \dots, L_n = L$ son reglas estrictas. ■

Ejemplo 3.3 *Dado el programa lógico rebatible $\mathcal{P}_{3.1}$ del Ejemplo 3.1 se pueden obtener, entre otras, las siguientes derivaciones estrictas:*

- *La secuencia $kiwi(haka), ave(haka)$ es una derivación estricta para el literal “ $ave(haka)$ ”.*
- *La secuencia $gallina(tina)$, que incluye solamente al hecho en cuestión, es una derivación estricta para el literal “ $gallina(tina)$ ”.*

El ejemplo anterior ilustra que, en particular, el conjunto de reglas estrictas utilizado en una derivación rebatible puede ser vacío. Por lo tanto, para todo literal que constituye un hecho del programa existirá una derivación estricta. Como se verá más adelante, las derivaciones estrictas sustentan conclusiones que no podrán ser refutadas, ya que se basan en información segura.

Dado que, como se mencionó anteriormente, a partir de un programa lógico rebatible es posible obtener derivaciones para literales complementarios, a continuación se definen los *conjuntos contradictorios*.

Definición 3.14 (*Conjunto Contradictorio*)

Dado un programa lógico rebatible \mathcal{P} , diremos que un subconjunto C de \mathcal{P} es contradictorio si y solo si a partir de C es posible obtener derivaciones (estrictas o rebatibles) para un literal L y su complemento \bar{L} . ■

Por ejemplo, considerando el programa lógico rebatible $\mathcal{P}_{3.1}$ del Ejemplo 3.1, el conjunto $\Pi_{3.1}$ no es contradictorio, mientras que el conjunto $\Pi_{3.1} \cup \Delta_{3.1}$ es contradictorio ya que, como se observó en el Ejemplo 3.2, permite obtener derivaciones rebatibles para los literales “*vuela(tina)*” y “ \sim *vuela(tina)*”.

Observación 3.2 *El uso de la negación fuerte favorece la expresividad del lenguaje y permite, potencialmente, la representación de información contradictoria. Como las reglas rebatibles permiten expresar información tentativa, dado un programa lógico rebatible $\mathcal{P} = (\Pi, \Delta)$, generalmente ocurrirá que el conjunto $\Pi \cup \Delta$ es contradictorio. Sin embargo, el conjunto Π de hechos y reglas estrictas es utilizado para representar información segura y, por lo tanto, debe poseer cierta coherencia interna. En consecuencia, para todo programa lógico rebatible $\mathcal{P} = (\Pi, \Delta)$ se asumirá que el conjunto Π es no contradictorio.*

La observación anterior establece que los programas lógicos rebatibles permiten expresar información potencialmente contradictoria mediante el uso de reglas rebatibles. Esto es, al añadir conocimiento rebatible al conjunto de hechos y reglas estrictas de un programa, es posible obtener derivaciones rebatibles para un literal y su complemento. Esto sugiere que la noción de derivación rebatible no resulta conveniente como procedimiento de prueba para un literal de un programa lógico rebatible. Por lo tanto, para decidir si un literal L está aceptado a partir de un programa, se realizará un análisis global más profundo considerando toda la información relevante del mismo.

En particular, DeLP define un procedimiento de prueba para decidir qué literales serán aceptados a partir de un programa lógico rebatible. Este procedimiento de prueba está basado en el formalismo de argumentación rebatible y permitirá definir *argumentos* para un literal, hallar los *contra-argumentos* para esos argumentos, y luego determinar las *derrotas* entre argumentos. Finalmente, un análisis dialéctico permitirá determinar cuándo un literal de un programa lógico rebatible está garantizado.

Argumentación rebatible

En DeLP se define la noción de *argumento* para un literal de un programa lógico rebatible, la cual se corresponde con el segundo elemento de la estructura conceptual presentada en la Sección 3.1.

Definición 3.15 (*Argumento*)

Sea $\mathcal{P} = (\Pi, \Delta)$ un programa lógico rebatible y h un literal. Un *argumento* para h es un par $\langle \mathcal{A}, L \rangle$ donde \mathcal{A} es un conjunto de reglas rebatibles de Δ tal que:

1. $\Pi \cup \mathcal{A} \vdash h$;
2. $\Pi \cup \mathcal{A}$ es un conjunto no contradictorio; y
3. \mathcal{A} es minimal: $\nexists \mathcal{A}' \subset \mathcal{A}$ tal que \mathcal{A}' satisface las condiciones (1) y (2).

■

Intuitivamente, un argumento \mathcal{A} para un literal h se caracteriza por un conjunto minimal y no contradictorio de reglas rebatibles que permiten obtener una derivación rebatible para h . Además, el literal h es identificado como la conclusión del argumento.

Ejemplo 3.4 *A partir del programa lógico rebatible $\mathcal{P}_{3.1}$ del Ejemplo 3.1 es posible construir los siguientes argumentos:*

- $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$, con $\mathcal{A} = \{\text{vuela}(\text{tina}) \multimap \text{ave}(\text{tina})\}$.
- $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$, con $\mathcal{B} = \{\sim \text{vuela}(\text{tina}) \multimap \text{gallina}(\text{tina})\}$.
- $\langle \mathcal{C}, \text{vuela}(\text{tina}) \rangle$, con $\mathcal{C} = \{\text{vuela}(\text{tina}) \multimap \text{gallina}(\text{tina}), \text{asustada}(\text{tina})\}$.

Además, es posible construir otros argumentos a partir de una secuencia vacía de reglas como $\langle \emptyset, \text{gallina}(\text{tina}) \rangle$, $\langle \emptyset, \text{ave}(\text{haka}) \rangle$ y $\langle \emptyset, \sim \text{vuela}(\text{haka}) \rangle$. Por otra parte, el Ejemplo 3.2 muestra que existe una derivación rebatible para el literal “vuela(haka)”. Sin embargo, no es posible obtener un argumento para dicho literal ya que la unión del conjunto de reglas rebatibles utilizadas en su derivación ($\{\text{vuela}(\text{haka}) \multimap \text{ave}(\text{haka})\}$) y el conjunto $\Pi_{3.1}$ conduce a un conjunto contradictorio dado que permite derivar los literales “vuela(haka)” y “ $\sim \text{vuela}(\text{haka})$ ”.

El ejemplo anterior permite observar que en DeLP es posible construir argumentos cuyo conjunto de reglas rebatibles es vacío. En particular, la conclusión de este tipo de argumentos es obtenida a partir de una derivación estricta, es decir, una derivación que utiliza solamente hechos y reglas estrictas del programa. De esta manera, DeLP asegura

que siempre existirá un argumento “*estricto*” (es decir, construido únicamente a partir del conjunto Π), para cada literal obtenido a partir de una derivación estricta. En consecuencia, si existe un argumento estricto para un literal h , entonces no existirán argumentos para \bar{h} . Esta situación ocurre en el Ejemplo 3.4, donde existe un argumento estricto para el literal “ $\sim vuela(haka)$ ”, mientras que no existen argumentos para el literal “ $vuela(haka)$ ”.

Teniendo en cuenta que un argumento constituye un conjunto de reglas rebatibles, es posible definir la noción de *sub-argumento*. Básicamente, un sub-argumento es un argumento cuyo conjunto de reglas rebatibles está contenido en otro argumento.

Definición 3.16 (*Sub-argumento*)

Sea \mathcal{P} un programa lógico rebatible y sean $\langle \mathcal{A}, p \rangle$, $\langle \mathcal{B}, q \rangle$ dos argumentos construidos a partir de \mathcal{P} . Diremos que $\langle \mathcal{B}, q \rangle$ es un *sub-argumento* de $\langle \mathcal{A}, p \rangle$ (análogamente, $\langle \mathcal{A}, p \rangle$ es un *super-argumento* de $\langle \mathcal{B}, q \rangle$) si y solo si $\mathcal{B} \subseteq \mathcal{A}$. ■

La definición anterior establece que todo argumento es, en particular, sub-argumento de sí mismo. Además, dado que los argumentos “*estrictos*” poseen un conjunto vacío de reglas rebatibles, estos serán sub-argumentos de cualquier otro argumento construido a partir de un programa DeLP. Si consideramos el Ejemplo 3.4, tenemos que, por ejemplo, los argumentos $\langle \emptyset, gallina(tina) \rangle$ y $\langle \emptyset, \sim vuela(haka) \rangle$ son sub-argumentos de $\langle \mathcal{A}, vuela(tina) \rangle$, $\langle \mathcal{B}, \sim vuela(tina) \rangle$ y $\langle \mathcal{C}, vuela(tina) \rangle$. En contraste, los argumentos $\langle \mathcal{A}, vuela(tina) \rangle$, $\langle \mathcal{B}, \sim vuela(tina) \rangle$ y $\langle \mathcal{C}, vuela(tina) \rangle$ no poseen super-argumentos. Para ilustrar otra situación que refleja la existencia de sub-argumentos, consideremos el siguiente ejemplo adaptado de [GS04].

Ejemplo 3.5 Sea $\mathcal{P}_{3.5} = (\Pi_{3.5}, \Delta_{3.5})$ el siguiente programa lógico rebatible:

$$\Pi_{3.5} = \left\{ \begin{array}{l} \text{buen_precio}(acme) \\ \text{en_fusion}(acme) \end{array} \right\}$$

$$\Delta_{3.5} = \left\{ \begin{array}{l} \text{comprar_acciones}(E) \rightarrow \text{buen_precio}(E) \\ \sim \text{comprar_acciones}(E) \rightarrow \text{buen_precio}(E), \text{empresa_riesgosa}(E) \\ \text{empresa_riesgosa}(E) \rightarrow \text{en_fusion}(E) \end{array} \right\}$$

A partir de $\mathcal{P}_{3.5}$ es posible construir los argumentos $\langle \mathcal{D}, \text{comprar_acciones}(acme) \rangle$, $\langle \mathcal{E}, \text{empresa_riesgosa}(acme) \rangle$ y $\langle \mathcal{F}, \sim \text{comprar_acciones}(acme) \rangle$, donde

$$\begin{aligned} \mathcal{D} &= \left\{ \text{comprar_acciones}(E) \rightarrow \text{buen_precio}(E) \right\} \\ \mathcal{E} &= \left\{ \text{empresa_riesgosa}(acme) \rightarrow \text{en_fusion}(acme) \right\} \\ \mathcal{F} &= \left\{ \begin{array}{l} \sim \text{comprar_acciones}(E) \rightarrow \text{buen_precio}(E), \text{empresa_riesgosa}(E) \\ \text{empresa_riesgosa}(E) \rightarrow \text{en_fusion}(E) \end{array} \right\} \end{aligned}$$

En este caso, se puede observar que el argumento $\langle \mathcal{E}, \text{empresa_riesgosa}(acme) \rangle$ es un sub-argumento de $\langle \mathcal{F}, \sim \text{comprar_acciones}(acme) \rangle$.

Identificación de conflictos: ataques entre argumentos

Como se mencionó anteriormente, el tercer elemento de la estructura conceptual introducida en la Sección 3.1 corresponde a la identificación de conflictos entre los argumentos del sistema. Dada la posibilidad de utilizar la negación fuerte “ \sim ”, es posible representar literales complementarios en un programa DeLP. Más aún, es posible obtener derivaciones para un par de literales complementarios, e incluso construir argumentos para estos literales. A continuación se introducirá la noción de *desacuerdo*, la cual generaliza la noción de literales complementarios para un programa DeLP.

Definición 3.17 (*Literales en Desacuerdo*)

Sea $\mathcal{P} = (\Pi, \Delta)$ un programa lógico rebatible. Diremos que dos literales h_1 y h_2 están en *desacuerdo* si y solo si el conjunto $\Pi \cup \{h_1, h_2\}$ es contradictorio. ■

Claramente, cualquier par de literales complementarios estará en desacuerdo. Por ejemplo, los literales “ p ” y “ $\sim p$ ” están en desacuerdo ya que el conjunto $\{p, \sim p\}$ es contradictorio de por sí, sin necesidad de tomar en cuenta el conjunto Π de hechos y reglas estrictas de un programa lógico rebatible. Sin embargo, como muestra el siguiente ejemplo, dos literales no complementarios también pueden estar en desacuerdo.

Ejemplo 3.6 Sea $\mathcal{P}_{3.10} = (\Pi_{3.10}, \emptyset)$ un programa lógico rebatible, donde $\Pi_{3.10} = \{(\sim h \leftarrow b), (h \leftarrow a)\}^2$. Los literales “a” y “b” están en desacuerdo dado que $\{a, b\} \cup \Pi_{3.10}$ es un conjunto contradictorio que, en particular, permite obtener derivaciones estrictas para los literales complementarios “h” y “ $\sim h$ ”:

A partir de la noción de desacuerdo es posible caracterizar los conflictos entre argumentos. De esta manera, dos argumentos se atacarán cuando sus conclusiones estén en desacuerdo. Más aún, un argumento atacará a otro si sustenta una conclusión que está en desacuerdo con la conclusión de alguno de los sub-argumentos del otro argumento. Esta intuición es capturada por la siguiente definición.

Definición 3.18 (*Contra-argumento o Ataque*)

Sean $\langle \mathcal{A}, p \rangle$ y $\langle \mathcal{B}, q \rangle$ dos argumentos construidos a partir de un programa lógico rebatible \mathcal{P} . El argumento $\langle \mathcal{A}, p \rangle$ *contra-argumenta* o *ataca* al argumento $\langle \mathcal{B}, q \rangle$ en el literal h si y solo si existe un sub-argumento $\langle \mathcal{C}, h \rangle$ de $\langle \mathcal{B}, q \rangle$ tal que los literales p y h están en desacuerdo. El argumento $\langle \mathcal{C}, h \rangle$ se llama sub-argumento de desacuerdo, y el literal h es considerado el punto de contra-argumentación. ■

Ejemplo 3.7 Considerando los argumentos ilustrados en el Ejemplo 3.4, el argumento $\langle \mathcal{B}, \sim \text{vuela}(tina) \rangle$ es un contra-argumento para los argumentos $\langle \mathcal{A}, \text{vuela}(tina) \rangle$ y $\langle \mathcal{C}, \text{vuela}(tina) \rangle$. En particular, los sub-argumentos de desacuerdo son, respectivamente, $\langle \mathcal{A}, \text{vuela}(tina) \rangle$ y $\langle \mathcal{C}, \text{vuela}(tina) \rangle$. Además, el punto de contraargumentación en ambos casos es el literal “vuela(tina)”. Por otra parte, recíprocamente, $\langle \mathcal{A}, \text{vuela}(tina) \rangle$ y $\langle \mathcal{C}, \text{vuela}(tina) \rangle$ son contra-argumentos para $\langle \mathcal{B}, \sim \text{vuela}(tina) \rangle$.

A partir de la Definición 3.18 puede ocurrir que, como se muestra en el Ejemplo 3.7, el sub-argumento $\langle \mathcal{C}, h \rangle$ sea el propio $\langle \mathcal{B}, q \rangle$. En tal caso diremos que $\langle \mathcal{A}, p \rangle$ contra-argumenta o ataca *directamente* a $\langle \mathcal{B}, q \rangle$. Por otra parte, cuando $\langle \mathcal{C}, h \rangle$ es un sub-argumento propio de $\langle \mathcal{B}, q \rangle$, el ataque se produce en un punto interno de $\langle \mathcal{B}, q \rangle$ (el punto de contra-argumentación h), en cuyo caso diremos que $\langle \mathcal{A}, p \rangle$ contra-argumenta o ataca *internamente* a $\langle \mathcal{B}, q \rangle$.

Nótese que los argumentos “*estrictos*” nunca tendrán contra-argumentos. Esto se debe a que, como se mencionó anteriormente, dado un argumento (construido solo a partir

²Para una mayor claridad en la notación, en algunos casos se utilizarán paréntesis para distinguir los elementos dentro de un conjunto.

del conjunto Π) para un literal h no será posible construir argumentos para literales en desacuerdo con h , ya que estos últimos estarían en contradicción con la parte estricta del programa que deriva h .

Resolución de conflictos: derrotas entre argumentos

La noción de contra-argumentación establece cuándo un argumento es atacado por otro, capturando la noción de conflicto entre argumentos a través del uso de la negación fuerte. Sin embargo, siempre que un argumento \mathcal{A} ataca a otro argumento \mathcal{B} , existe un sub-argumento \mathcal{C} de \mathcal{B} que es un contra-argumento para \mathcal{A} . Dada esta simetría, la noción de ataque no permite identificar cuándo un argumento se impone por sobre el otro. Por este motivo es necesario contar con un mecanismo de evaluación sobre los argumentos que establezca cuándo un argumento es mejor que otro, para así determinar cuál de ellos prevalece ante un conflicto. Este mecanismo involucra el uso de un *criterio de comparación entre argumentos* que permite determinar el cuarto elemento de la estructura conceptual presentada en la Sección 3.1: las *derrotas* entre argumentos. DeLP utiliza un criterio de comparación modular³ para determinar cuándo un argumento es tanto o más preferido que otro argumento. Por lo tanto, como la comparación entre argumentos puede definirse de diversas formas, es conveniente abstraerse de la especificación de este criterio, y es posible asumir que se trata de un orden parcial entre argumentos denotado “ \succeq ”. De esta manera, se utiliza $\mathcal{A} \succeq \mathcal{B}$ para denotar que \mathcal{A} es al menos tan preferido como \mathcal{B} . Así, siguiendo la notación usualmente adoptada en la literatura, se escribirá $\mathcal{A} \succ \mathcal{B}$ para expresar que \mathcal{A} es preferido a \mathcal{B} .

Si $\langle \mathcal{A}, p \rangle$ es un contra-argumento para $\langle \mathcal{B}, q \rangle$, es necesario analizar si el ataque de $\langle \mathcal{A}, p \rangle$ sobre $\langle \mathcal{B}, q \rangle$ es lo suficientemente fuerte como para derrotar a $\langle \mathcal{B}, q \rangle$. En consecuencia, es necesario efectuar una comparación entre el argumento atacante y el sub-argumento atacado. Concretamente, si $\langle \mathcal{A}, p \rangle$ ataca a $\langle \mathcal{B}, q \rangle$ en el punto h , siendo $\langle \mathcal{C}, h \rangle$ el subargumento de desacuerdo de $\langle \mathcal{B}, q \rangle$, se debe comparar el argumento atacante $\langle \mathcal{A}, p \rangle$ y el sub-argumento de desacuerdo $\langle \mathcal{C}, h \rangle$.

Definición 3.19 (*Derrota*)

Sean $\langle \mathcal{A}, p \rangle$ y $\langle \mathcal{B}, q \rangle$ dos argumentos construidos a partir de un programa lógico rebati-

³El criterio es especificado por el usuario, permitiendo así emplear aquel que mejor se adapte al dominio de aplicación.

ble \mathcal{P} . El argumento $\langle \mathcal{A}, p \rangle$ es un *derrotador* para el argumento $\langle \mathcal{B}, q \rangle$ si y solo si existe un sub-argumento $\langle \mathcal{C}, h \rangle$ de $\langle \mathcal{B}, q \rangle$ tal que $\langle \mathcal{A}, p \rangle$ ataca a $\langle \mathcal{B}, q \rangle$ en el literal h y vale que:

- $\langle \mathcal{A}, p \rangle$ es mejor que $\langle \mathcal{C}, h \rangle$ de acuerdo al criterio de comparación utilizado (es decir, $\langle \mathcal{A}, p \rangle \succ \langle \mathcal{C}, h \rangle$), denotando que $\langle \mathcal{A}, p \rangle$ es un *derrotador propio* de $\langle \mathcal{B}, q \rangle$; o
- $\langle \mathcal{A}, p \rangle$ no es mejor que $\langle \mathcal{C}, h \rangle$, y $\langle \mathcal{C}, h \rangle$ no es mejor que $\langle \mathcal{A}, p \rangle$ de acuerdo al criterio de comparación utilizado (es decir, $\langle \mathcal{C}, h \rangle \not\succeq \langle \mathcal{A}, p \rangle$ y $\langle \mathcal{A}, p \rangle \not\succeq \langle \mathcal{C}, h \rangle$), denotando que $\langle \mathcal{A}, p \rangle$ es un *derrotador por bloqueo* de $\langle \mathcal{B}, q \rangle$.

■

La definición anterior identifica dos situaciones en las que un ataque entre argumentos resulta exitoso. La primera de ellas, correspondiente a la derrota propia, describe la situación en que el argumento atacante $\langle \mathcal{A}, p \rangle$ es estrictamente preferido al sub-argumento atacado $\langle \mathcal{C}, h \rangle$. En contraste, el segundo caso describe la derrota por bloqueo, la cual ocurre cuando el argumento atacante $\langle \mathcal{A}, p \rangle$ y el sub-argumento atacado $\langle \mathcal{C}, h \rangle$ son igualmente preferidos o resultan incomparables de acuerdo al criterio de comparación adoptado. Si bien DeLP utiliza un criterio de comparación modular, existen criterios concretos para comparar argumentos en este formalismo (ver por ejemplo, [GS04, FECS07]). En particular, en los ejemplos de esta tesis utilizaremos el criterio de comparación *especificidad generalizada* [GS04], el cual se basa en la noción de especificidad definida en [Poo85, SL92]. Resumidamente, este criterio prefiere aquellos argumentos con información más precisa o argumentos más directos. Por ejemplo, el argumento $\langle \{\sim a \rightarrow b, c\}, \sim a \rangle$ es considerado más específico que el argumento $\langle \{a \rightarrow b\}, a \rangle$ porque utiliza más información para derivar su conclusión. De manera similar, el argumento $\langle \{a \rightarrow b\}, a \rangle$ es más específico que el argumento $\langle \{(a \rightarrow d), (d \rightarrow b)\}, \sim a \rangle$ por ser más directo.

Definición 3.20 (*Especificidad Generalizada*)

Sea $\mathcal{P} = (\Theta, \Omega, \Delta)$ un programa lógico rebatible y sea $Literales(\mathcal{P})$ el conjunto de literales que tienen una derivación (estricta o rebatible) a partir de \mathcal{P} . Un argumento $\langle \mathcal{A}, p \rangle$ es *más específico* que otro argumento $\langle \mathcal{B}, q \rangle$, denotado $\langle \mathcal{A}, p \rangle \succ \langle \mathcal{B}, q \rangle$, si se verifican las siguientes condiciones:

1. para todo conjunto $H \subseteq Literales(\mathcal{P})$ vale que:
si $\Omega \cup H \cup \mathcal{A} \sim p$ (es decir, H activa a \mathcal{A}),

y $\Omega \cup H \not\vdash p$,

entonces $\Omega \cup H \cup \mathcal{B} \sim q$ (es decir, H activa a \mathcal{B}); y

2. existe al menos un conjunto $H' \subseteq \text{Literales}(\mathcal{P})$ tal que:

$\Omega \cup H' \cup \mathcal{B} \sim q$ (es decir, H' activa a \mathcal{B}),

$\Omega \cup H' \not\vdash q$, y

$\Omega \cup H' \cup \mathcal{A} \not\vdash p$ (es decir, H' no activa a \mathcal{A}).

■

Dada la definición anterior, los conjuntos H y H' contienen literales, los cuales son tomados como hechos para obtener las derivaciones. La condición (1) establece que para todo conjunto de literales $H \subseteq \text{Lits}(\mathcal{P})$, si H permite derivar p utilizando reglas de $\Omega \cup \mathcal{A}$, donde al menos una regla de \mathcal{A} es utilizada (ya que no existe una derivación estricta de p a partir de $\Omega \cup H$, entonces ese mismo conjunto H permite derivar rebatiblemente a q utilizando las reglas de $\Omega \cup \mathcal{B}$. Luego, la condición (2) busca que exista al menos un conjunto $H' \subseteq \text{Lits}(\mathcal{P})$ que permita derivar rebatiblemente a q pero no a p . Para ilustrar el uso de este criterio de comparación, consideremos el siguiente ejemplo.

Ejemplo 3.8 *Como se observó en el Ejemplo 3.7, existen ataques de los argumentos $\langle \mathcal{A}, \text{vuela}(tina) \rangle$ y $\langle \mathcal{C}, \text{vuela}(tina) \rangle$ hacia el argumento $\langle \mathcal{B}, \sim \text{vuela}(tina) \rangle$ y viceversa, donde los sub-argumentos de desacuerdo son los mismos argumentos atacados. Veamos ahora cómo se resuelven estos ataques mediante el uso de especificidad generalizada como criterio de comparación. En este caso debemos comparar el argumento $\langle \mathcal{B}, \sim \text{vuela}(tina) \rangle$ respectivamente con los argumentos $\langle \mathcal{A}, \text{vuela}(tina) \rangle$ y $\langle \mathcal{C}, \text{vuela}(tina) \rangle$.*

Todo conjunto H_1 que active al argumento

$$\langle \mathcal{B}, \sim \text{vuela}(tina) \rangle = \langle \{ \sim \text{vuela}(tina) \rightarrow \text{gallina}(tina) \}, \sim \text{vuela}(tina) \rangle$$

deberá contener al literal “gallina(tina)”, por lo que H también activará al argumento

$$\langle \mathcal{A}, \text{vuela}(tina) \rangle = \langle \{ \text{vuela}(tina) \rightarrow \text{ave}(tina) \}, \text{vuela}(tina) \rangle$$

dada la existencia de la regla estricta “ave(tina) \leftarrow gallina(tina)” en el programa $\mathcal{P}_{3.1}$.

En contraste, el conjunto $H'_1 = \{\text{ave}(\text{tina})\}$ activa a $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$ pero no a $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$. De esta manera, $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$ es más específico que $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$, conduciendo a una derrota propia de $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$ sobre $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$.

Por otra parte, todo conjunto H_2 que active al argumento

$$\langle \mathcal{C}, \text{vuela}(\text{tina}) \rangle = \langle \{\text{vuela}(\text{tina}) \multimap \text{gallina}(\text{tina}), \text{asustada}(\text{tina})\}, \text{vuela}(\text{tina}) \rangle$$

deberá contener los literales “gallina(tina)” y “asustada(tina)”, por lo que el argumento

$$\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle = \langle \{\sim \text{vuela}(\text{tina}) \multimap \text{gallina}(\text{tina})\}, \sim \text{vuela}(\text{tina}) \rangle$$

también será activado por H_2 .

Luego, el conjunto $H'_2 = \{\text{gallina}(\text{tina})\}$ activa a $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$ pero no a $\langle \mathcal{C}, \text{vuela}(\text{tina}) \rangle$. En consecuencia, $\langle \mathcal{C}, \text{vuela}(\text{tina}) \rangle$ es más específico que $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$, siendo entonces un derrotador propio de $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$.

A continuación se introducirá una representación gráfica para los argumentos de DeLP, los ataques entre argumentos y las correspondientes derrotas. En la notación gráfica propuesta, los argumentos serán representados mediante triángulos. Además, los triángulos dentro de triángulos mayores denotarán sub-argumentos. La información ubicada en el vértice superior de un triángulo corresponde a la conclusión del argumento (sub-argumento) respectivo, mientras que la información situada dentro del triángulo corresponde a los literales utilizados para construir el argumento. El símbolo “ \multimap ” representa la conexión establecida entre estos literales a través de las reglas rebatibles pertenecientes al argumento. En particular, los sub-argumentos correspondientes a argumentos estrictos se denotarán identificando únicamente el literal que concluyen. Por último, los ataques entre argumentos se denotarán mediante flechas punteadas, mientras que las derrotas estarán identificadas mediante flechas sólidas. Esta notación se halla ilustrada por el siguiente ejemplo.

Ejemplo 3.9 Consideremos el programa lógico rebatible $\mathcal{P}_{3.1}$ del Ejemplo 3.1. La Figura 3.2 ilustra todos los argumentos construibles a partir de $\mathcal{P}_{3.1}$ (algunos de los cuales fueron previamente introducidos en el Ejemplo 3.4), así como también los vínculos de ataque y derrota entre ellos (respectivamente introducidos en los ejemplos 3.7 y 3.8).

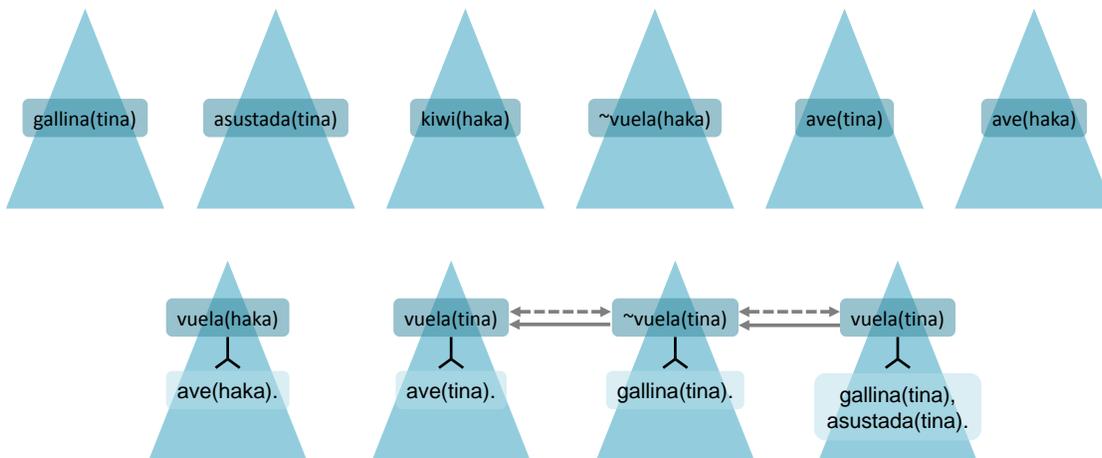


Figura 3.2: Representación gráfica de los argumentos, ataques y derrotas obtenidos a partir del programa lógico rebatible $\mathcal{P}_{3.1}$ del Ejemplo 3.1.

Cálculo de Aceptabilidad: argumentos aceptados y conclusiones garantizadas

Como se explicó al comienzo de este capítulo, el objetivo de un sistema argumentativo está determinado por el quinto elemento de la estructura conceptual presentada en la Sección 3.1: identificación de los argumentos aceptados. Para esto, es necesario considerar las derrotas entre los argumentos del sistema. En el caso de DeLP, además, los argumentos aceptados determinarán qué literales están *garantizados* a partir de un programa lógico rebatible. Dado el conjunto de argumentos que pueden construirse a partir de un programa lógico rebatible \mathcal{P} , la relación de derrota entre argumentos solo puede establecer un orden de preferencia sobre los argumentos en conflicto. Sin embargo, el estado de aceptabilidad de un argumento $\langle \mathcal{A}, h \rangle$ con respecto al programa \mathcal{P} dependerá de la interacción de $\langle \mathcal{A}, h \rangle$ con el resto de los argumentos del sistema. Por ejemplo, si a partir de un programa \mathcal{P} se obtiene un argumento $\langle \mathcal{A}_0, h_0 \rangle$ que no posee derrotadores, entonces un individuo que disponga de \mathcal{P} para razonar podrá “creer” en el literal h_0 . Sin embargo, podría ocurrir que $\langle \mathcal{A}_0, h_0 \rangle$ posea un derrotador $\langle \mathcal{A}_1, h_1 \rangle$, con lo cual existirán razones para invalidar la creencia del individuo en h_0 . Tal es el caso del Ejemplo 3.8, donde la derrota de $\langle \mathcal{B}, \sim vuela(tina) \rangle$ sobre $\langle \mathcal{A}, vuela(tina) \rangle$ invalida la creencia de que Tina vuela. Sin embargo, también puede ocurrir que el argumento $\langle \mathcal{A}_1, h_1 \rangle$ tenga a su vez un derrotador $\langle \mathcal{A}_2, h_2 \rangle$, reinstalando así la creencia del individuo en h_0 . Nuevamente, esto ocurre en el Ejemplo 3.8 mediante la derrota de $\langle \mathcal{C}, vuela(tina) \rangle$ sobre $\langle \mathcal{B}, \sim vuela(tina) \rangle$, reinstalando la creencia de que Tina vuela.

La secuencia de interacciones arriba mencionada puede ir más lejos, existiendo a su vez un derrotador $\langle \mathcal{A}_3, h_3 \rangle$ para $\langle \mathcal{A}_2, h_2 \rangle$ que invalida nuevamente la creencia del individuo en h_0 y así siguiendo. Esto da origen a una secuencia de argumentos $[\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \langle \mathcal{A}_3, h_3 \rangle, \dots]$ conocida como *línea argumentativa*, donde cada elemento en la secuencia es un derrotador de su predecesor.

Definición 3.21 (*Línea Argumentativa*)

Sea \mathcal{P} un programa lógico rebatible y $\langle \mathcal{A}_0, h_0 \rangle$ un argumento construido a partir de \mathcal{P} . Una *línea argumentativa* para $\langle \mathcal{A}_0, h_0 \rangle$ es una secuencia de argumentos de \mathcal{P} , denotada $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \langle \mathcal{A}_3, h_3 \rangle, \dots]$, donde cada argumento $\langle \mathcal{A}_i, h_i \rangle$ perteneciente a la secuencia es un derrotador del argumento predecesor $\langle \mathcal{A}_{i-1}, h_{i-1} \rangle$. ■

Dada una línea argumentativa para el argumento $\langle \mathcal{A}_0, h_0 \rangle$, los argumentos en posiciones impares conforman el conjunto de *argumentos de soporte* (es decir, argumentos que de alguna manera reinstalan la posición de $\langle \mathcal{A}_0, h_0 \rangle$) y se denotan como Λ_A , mientras que los argumentos en posiciones pares componen el conjunto de *argumentos de interferencia* (es decir, argumentos en contra de la posición de $\langle \mathcal{A}_0, h_0 \rangle$) denotado como Λ_I .

Para evitar secuencias de argumentos indeseables que pueden representar cadenas de razonamiento falaz, DeLP impone el requerimiento de que una línea argumentativa debe ser *acceptable*. De esta manera, se define una serie de características que este tipo de líneas argumentativas debe satisfacer. Resumidamente, una línea argumentativa será acceptable si es una secuencia finita, no contiene argumentos o sub-argumentos repetidos, no posee dos derrotas por bloqueo consecutivas, y los conjuntos de argumentos de soporte e interferencia son consistentes con el conocimiento estricto.

El requisito de que una línea argumentativa sea finita está directamente relacionado con el requerimiento de no introducir argumentos repetidos en la línea, con el objetivo de evitar lo que se denomina como *argumentación circular* [GS04]. La Figura 3.3 ilustra dos situaciones que conducen a argumentación circular. La primera de ellas, correspondiente a la Figura 3.3(a), ocurre cuando un argumento es reintroducido para defenderse a sí mismo. El otro caso, ilustrado en la Figura 3.3(b), corresponde a una generalización del caso anterior, en el que un sub-argumento de desacuerdo es reintroducido en la línea para defender al argumento que lo contiene. Claramente, las situaciones ilustradas en la Figura 3.3 conducen a procesos argumentativos falaces y posiblemente infinitos, motivo por el cual debe evitarse que ocurran en una línea argumentativa acceptable.

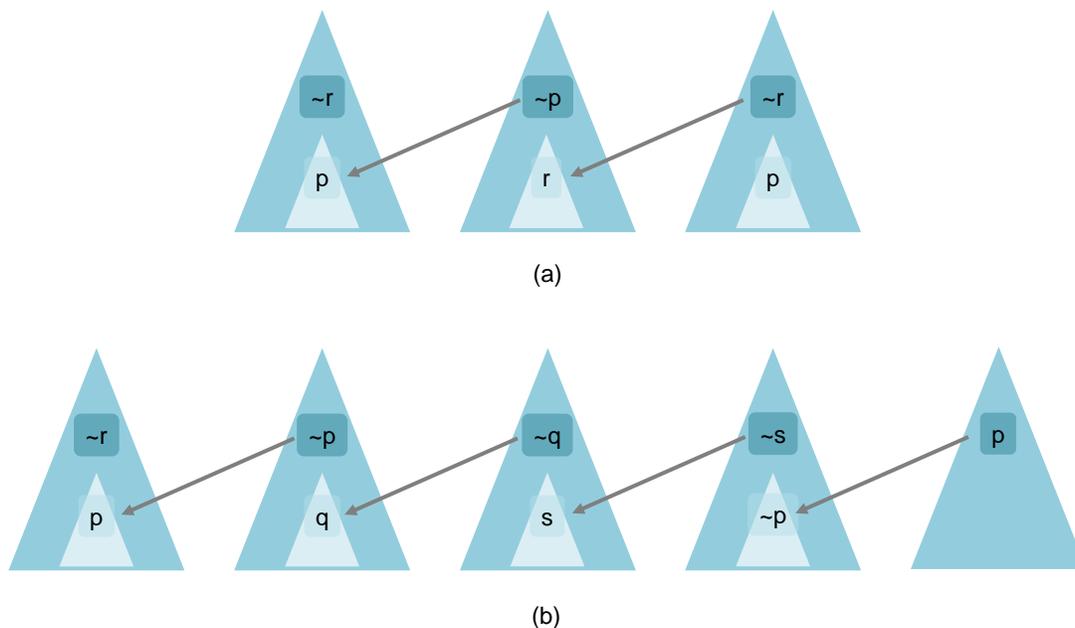


Figura 3.3: Ejemplos de argumentación circular.

El requisito de impedir dos derrotas por bloqueo consecutivas en una línea argumentativa aceptable es razonable por el siguiente motivo. Una derrota por bloqueo de un argumento $\langle \mathcal{A}_1, h_1 \rangle$ sobre un argumento $\langle \mathcal{A}_0, h_0 \rangle$ se desprende del hecho de que los argumentos $\langle \mathcal{A}_1, h_1 \rangle$ y $\langle \mathcal{A}_0, h_0 \rangle$ resultaron equivalentes o incomparables de acuerdo al criterio de comparación utilizado. Es decir, es una derrota que corresponde a un conflicto que no pudo ser resuelto mediante la asignación de pesos a los argumentos. Luego, si existe un argumento $\langle \mathcal{A}_2, h_2 \rangle$ tal que derrota por bloqueo a $\langle \mathcal{A}_1, h_1 \rangle$, se tiene que el criterio de comparación tampoco fue capaz de decidir entre los argumentos $\langle \mathcal{A}_2, h_2 \rangle$ y $\langle \mathcal{A}_1, h_1 \rangle$. De esta manera, si se permitiera incluir dos derrotas por bloqueo seguidas en una línea argumentativa aceptable, se estaría determinando en cierta medida que el argumento $\langle \mathcal{A}_0, h_0 \rangle$ prevalece ante la derrota de $\langle \mathcal{A}_1, h_1 \rangle$, dado que es defendido por $\langle \mathcal{A}_2, h_2 \rangle$. Claramente esta es una situación indeseable, ya que el literal h_0 pertenecería a las inferencias del sistema simplemente por poseer más argumentos que lo apoyan.

Por último, la condición que indica que los argumentos de apoyo y los de interferencia deben ser consistentes con el conocimiento estricto intenta evitar la situación falaz en la que un argumento es defendido por otro argumento que se encuentra en conflicto con él. Este requerimiento establece que al considerar conjuntamente estos argumentos y el conocimiento estricto, el conjunto resultante no debe ser contradictorio. Teniendo

en cuenta todas estas condiciones, la siguiente definición formaliza la noción de línea argumentativa aceptable.

Definición 3.22 (*Línea Argumentativa Aceptable*)

Sea $\mathcal{P} = (\Pi, \Delta)$ un programa lógico rebatible y $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \dots, \langle \mathcal{A}_i, h_i \rangle, \dots, \langle \mathcal{A}_n, h_n \rangle]$ una línea argumentativa para $\langle \mathcal{A}_0, h_0 \rangle$ obtenida a partir de \mathcal{P} . Diremos que Λ es una *línea argumentativa aceptable* si satisface las siguientes condiciones:

1. Λ es una secuencia finita;
2. el conjunto Λ_A de argumentos de soporte de Λ y el conjunto Λ_I de argumentos de interferencia de Λ son tal que $\Pi \cup \{\mathcal{A} \mid \langle \mathcal{A}, p \rangle \in \Lambda_A\}$ y $\Pi \cup \{\mathcal{B} \mid \langle \mathcal{B}, q \rangle \in \Lambda_I\}$ son conjuntos no contradictorios;
3. ningún argumento $\langle \mathcal{A}_k, h_k \rangle$ de Λ es un sub-argumento de un argumento $\langle \mathcal{A}_j, h_j \rangle$ que aparece previamente en Λ ($j < k$); y
4. para todo argumento $\langle \mathcal{A}_i, h_i \rangle$ de Λ tal que $\langle \mathcal{A}_i, h_i \rangle$ es un derrotador por bloqueo de $\langle \mathcal{A}_{i-1}, h_{i-1} \rangle$ y no es un derrotador propio de $\langle \mathcal{A}_{i-1}, h_{i-1} \rangle$, si existe $\langle \mathcal{A}_{i+1}, h_{i+1} \rangle$ en Λ , entonces $\langle \mathcal{A}_{i+1}, h_{i+1} \rangle$ es un derrotador propio de $\langle \mathcal{A}_i, h_i \rangle$.

■

Dada la Definición 3.22, pareciera ser que la noción de línea argumentativa aceptable provee un mecanismo para decidir sobre la garantía de un literal h_0 a partir de un programa \mathcal{P} . Siguiendo esta definición, podríamos establecer lo siguiente: si la línea argumentativa para $\langle \mathcal{A}_0, h_0 \rangle$ tiene un número impar de argumentos, entonces el literal h_0 está garantizado; en caso contrario, si la línea argumentativa posee una cantidad par de argumentos o no existe argumento alguno para h_0 , entonces no lo está. No obstante, como se muestra en el siguiente ejemplo, una línea argumentativa no es suficiente para analizar la interacción entre argumentos, ya que para cada argumento puede existir más de un derrotador.

Ejemplo 3.10 *Sea \mathcal{P} un programa lógico rebatible a partir del cual se construyen los argumentos $\langle \mathcal{A}_0, h_0 \rangle$, $\langle \mathcal{A}_1, h_1 \rangle$, $\langle \mathcal{A}_2, h_2 \rangle$ y $\langle \mathcal{A}_3, h_3 \rangle$. Supongamos que $\langle \mathcal{A}_1, h_1 \rangle$ y $\langle \mathcal{A}_2, h_2 \rangle$ son derrotadores propios de $\langle \mathcal{A}_0, h_0 \rangle$, y que $\langle \mathcal{A}_3, h_3 \rangle$ es un derrotador propio de $\langle \mathcal{A}_2, h_2 \rangle$. A partir de esto, se obtienen las siguientes líneas argumentativas aceptables para $\langle \mathcal{A}_0, h_0 \rangle$:*

- $\Lambda_1 = [\langle \mathcal{A}_0, h_0 \rangle]$
- $\Lambda_2 = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle]$
- $\Lambda_3 = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_2, h_2 \rangle]$
- $\Lambda_4 = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \langle \mathcal{A}_3, h_3 \rangle]$

En este caso, no resulta adecuado considerar aisladamente las líneas argumentativas aceptables para decidir sobre la garantía de h_0 a partir de \mathcal{P} . Esto se debe a que, por ejemplo, la línea argumentativa Λ_4 posee una cantidad impar de argumentos, sugiriendo que h_0 estaría garantizado a partir de \mathcal{P} . En contraste, por ejemplo, Λ_2 sugiere lo contrario, dado que posee una cantidad par de argumentos.

El Ejemplo 3.10 muestra que la noción de línea argumentativa aceptable no es suficiente como mecanismo de prueba para los literales inferidos a partir de un programa lógico rebatible. Dado que para cada argumento puede existir más de un derrotador, esto conduce a una ramificación de líneas argumentativas, dando origen a un árbol de derrotadores llamado *árbol de dialéctica*. Además, nótese que la noción de línea argumentativa aceptable no asegura la exploración completa de los derrotadores para un argumento. Si consideramos el Ejemplo 3.10, la línea argumentativa $[\langle \mathcal{A}_0, h_0 \rangle]$ satisface las condiciones impuestas por la Definición 3.22. De esta manera, para asegurar el análisis exhaustivo de derrotadores, la noción de árbol de dialéctica agrupa todas las líneas argumentativas aceptables maximales.

Definición 3.23 (*Árbol de Dialéctica*)

Sea \mathcal{P} un programa lógico rebatible y $\langle \mathcal{A}_0, h_0 \rangle$ un argumento construido a partir de \mathcal{P} . Un *árbol de dialéctica* para $\langle \mathcal{A}_0, h_0 \rangle$, denotado $\mathcal{T}_{\langle \mathcal{A}_0, h_0 \rangle}$, es un árbol tal que los nodos son argumentos construidos a partir de \mathcal{P} , los arcos representan derrotas entre los nodos, y se verifican las siguientes condiciones:

1. $\langle \mathcal{A}_0, h_0 \rangle$ es la raíz de $\mathcal{T}_{\langle \mathcal{A}_0, h_0 \rangle}$; y
2. si $\Lambda = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \dots, \langle \mathcal{A}_n, h_n \rangle]$ es una línea argumentativa aceptable para $\langle \mathcal{A}_0, h_0 \rangle$ y no existe $\langle \mathcal{A}_m, h_m \rangle$ tal que $\Lambda' = [\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \dots, \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{A}_m, h_m \rangle]$ es una línea argumentativa aceptable para $\langle \mathcal{A}_0, h_0 \rangle$, entonces $\langle \mathcal{A}_n, h_n \rangle$ es una hoja de $\mathcal{T}_{\langle \mathcal{A}_0, h_0 \rangle}$ y Λ es un camino desde la raíz $\langle \mathcal{A}_0, h_0 \rangle$ hasta la hoja $\langle \mathcal{A}_n, h_n \rangle$.



Ejemplo 3.11 Consideremos el programa lógico rebatible $\mathcal{P}_{3.1}$ del Ejemplo 3.1 y el argumento $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$ construido a partir de $\mathcal{P}_{3.1}$ (ver Ejemplo 3.4). De acuerdo al Ejemplo 3.8, el argumento $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$ es un derrotador propio de $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$, y el argumento $\langle \mathcal{C}, \text{vuela}(\text{tina}) \rangle$ es un derrotador propio de $\langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle$, siendo estas las únicas derrotas existentes. Por lo tanto, la única línea argumentativa aceptable (maximal) para $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$ es $\Lambda_{3.15} = [\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle, \langle \mathcal{B}, \sim \text{vuela}(\text{tina}) \rangle, \langle \mathcal{C}, \text{vuela}(\text{tina}) \rangle]$. En particular, $\Lambda_{3.15}$ constituye el árbol de dialéctica $\mathcal{T}_{\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle}$ para el argumento $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$, y se encuentra ilustrado en la Figura 3.4.

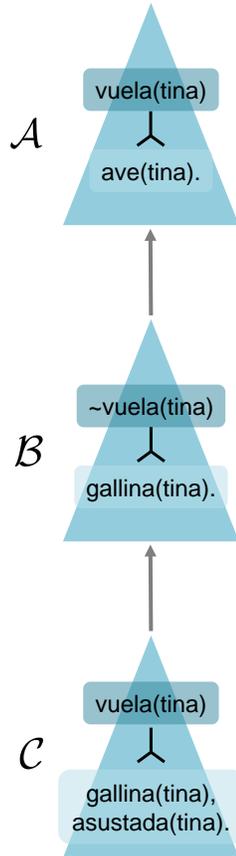


Figura 3.4: Árbol de dialéctica $\mathcal{T}_{\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle}$ para $\langle \mathcal{A}, \text{vuela}(\text{tina}) \rangle$, correspondiente al Ejemplo 3.11.

Como puede observarse en el Ejemplo 3.11, los nodos hoja del árbol de dialéctica corresponden a argumentos no derrotados. En contraste, un nodo interno que tiene como

hijo a un nodo hoja corresponderá a un argumento derrotado. Siguiendo con este análisis desde las hojas hacia la raíz, los nodos en un árbol de dialéctica pueden marcarse como *derrotados* o *no derrotados*, respectivamente denotados como D y U (por sus nombres en inglés, *defeated* y *undefeated*). La siguiente definición determina cómo se efectúa el procedimiento de marcado para un árbol de dialéctica de DeLP.

Definición 3.24 (*Árbol de Dialéctica Marcado*)

Sea \mathcal{P} un programa lógico rebatible y $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ un árbol de dialéctica para un argumento $\langle \mathcal{A}, h \rangle$ construido a partir de \mathcal{P} . El *árbol de dialéctica marcado* para $\langle \mathcal{A}, h \rangle$, denotado $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$, se obtiene marcando cada nodo \mathcal{N} de $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ como D (derrotado) o U (no derrotado) de acuerdo al siguiente criterio:

- Si \mathcal{N} es un nodo hoja de $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$, entonces \mathcal{N} es marcado como U en $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$.
- Si \mathcal{N} es un nodo interno de $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ y posee algún hijo marcado como U en $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$, entonces \mathcal{N} es marcado como D en $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$.
- Si \mathcal{N} es un nodo interno de $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ y todos sus hijos son marcados como D en $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$, entonces \mathcal{N} es marcado como U en $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$.

■

Ejemplo 3.12 *Dado el árbol de dialéctica $\mathcal{T}_{\langle \mathcal{A}, \text{vuela}(tina) \rangle}$ del Ejemplo 3.1 (ilustrado en la Figura 2.5), el proceso de marcado determina que el árbol de dialéctica marcado para $\langle \mathcal{A}, \text{vuela}(tina) \rangle$ es $\mathcal{T}_{\langle \mathcal{A}, \text{vuela}(tina) \rangle}^*$, donde el argumento $\langle \mathcal{C}, \text{vuela}(tina) \rangle$ está marcado como U, el argumento $\langle \mathcal{B}, \sim \text{vuela}(tina) \rangle$ está marcado como D, y el argumento $\langle \mathcal{A}, \text{vuela}(tina) \rangle$ está marcado como U.*

El siguiente ejemplo ilustra un caso más general del proceso de marcado de árboles de dialéctica en DeLP.

Ejemplo 3.13 *Consideremos un programa lógico rebatible a partir del cual se construye un argumento $\langle \mathcal{A}, i \rangle$, cuyo árbol de dialéctica $\mathcal{T}_{\langle \mathcal{A}, i \rangle}$ se ilustra en la Figura 3.5. El árbol de dialéctica marcado $\mathcal{T}_{\langle \mathcal{A}, i \rangle}^*$, correspondiente al argumento $\langle \mathcal{A}, i \rangle$, se ilustra en la Figura 3.7. En este caso, se puede observar que los argumentos correspondientes a nodos hoja del árbol son marcados como U. Luego, aquellos argumentos correspondientes a nodos padre de los*

nodos hoja son marcados como D. Finalmente, y continuando con el proceso establecido por la Definición 3.24, tenemos que el argumento $\langle \mathcal{A}, i \rangle$, correspondiente a la raíz del árbol de dialéctica marcado $\mathcal{T}_{\langle \mathcal{A}, i \rangle}^$, está marcado como D.*

Un árbol de dialéctica marcado $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$ representa el análisis dialéctico que considera todos los argumentos relevantes de un programa lógico rebatible \mathcal{P} a fin de decidir el estado de aceptabilidad de un argumento $\langle \mathcal{A}, h \rangle$. Es decir, el análisis dialéctico toma en cuenta todos los argumentos de \mathcal{P} que influyen, por pertenecer a una línea argumentativa aceptable para $\langle \mathcal{A}, h \rangle$, en la determinación del estado de aceptabilidad del argumento $\langle \mathcal{A}, h \rangle$. Por lo tanto, el estado de un argumento $\langle \mathcal{A}, h \rangle$ será *rechazado* si la raíz de $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$ está marcada como D, o *aceptado* si la raíz de $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$ está marcada como U.

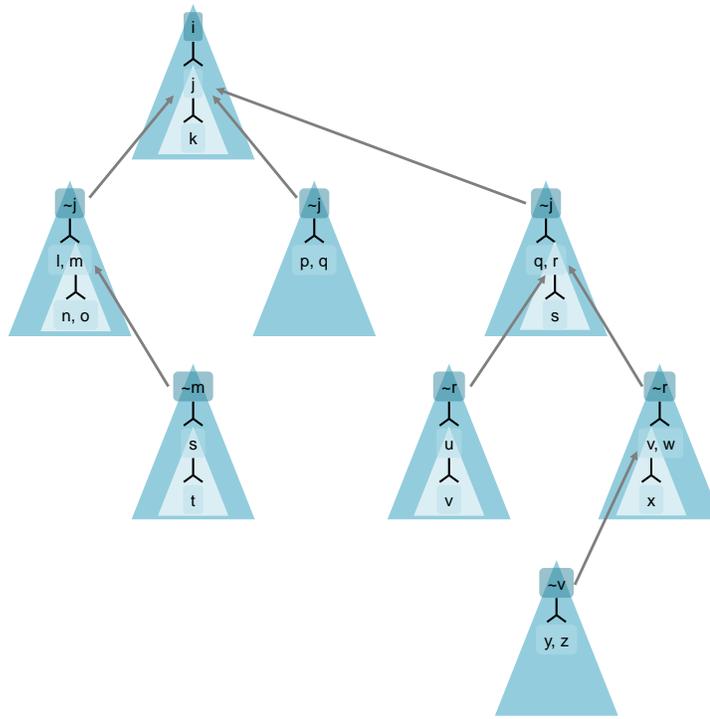


Figura 3.5: Árbol de dialéctica $\mathcal{T}_{\langle \mathcal{A}, i \rangle}$ correspondiente al Ejemplo 3.13.

Definición 3.25 (*Argumento Aceptado*)

Sea \mathcal{P} un programa lógico rebatible y $\langle \mathcal{A}, h \rangle$ un argumento construido a partir de \mathcal{P} . Diremos que $\langle \mathcal{A}, h \rangle$ es un *argumento aceptado* de \mathcal{P} si y solo si la raíz del árbol de dialéctica marcado $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$ está marcada como U. En caso contrario, diremos que $\langle \mathcal{A}, h \rangle$ es un *argumento rechazado* de \mathcal{P} . ■

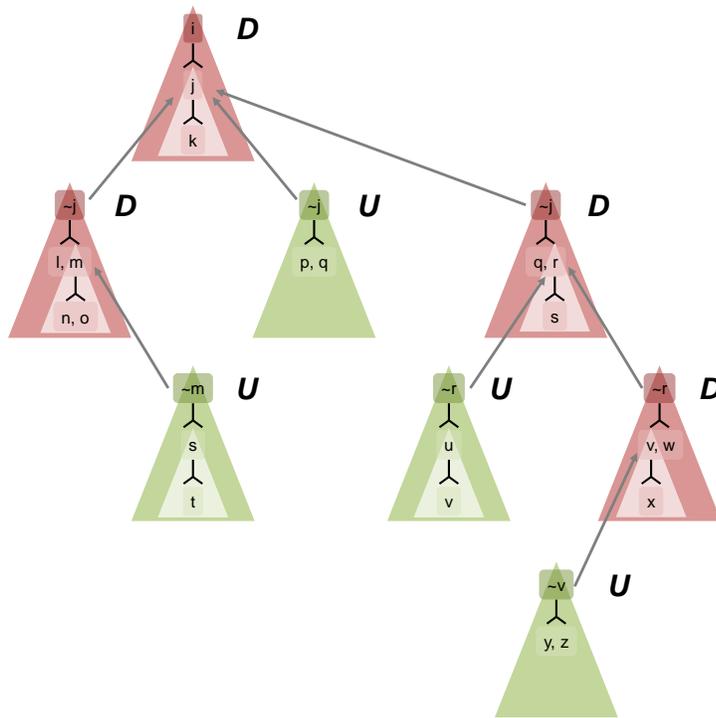


Figura 3.6: Árbol de dialéctica marcado $\mathcal{T}_{\langle \mathcal{A}, i \rangle}^*$ correspondiente al Ejemplo 3.13.

Claramente, si hay un argumento aceptado a partir de un programa lógico rebatible, esto implica que hay una garantía para inferir su conclusión. Siguiendo este razonamiento, DeLP define la noción de *literal garantizado*, correspondiente a aquellos literales que son las conclusiones de los argumentos aceptados. En consecuencia, los literales garantizados a partir de un programa lógico rebatible serán las inferencias obtenidas bajo el procedimiento de prueba dialéctica de DeLP.

Definición 3.26 (*Literal Garantizado*)

Sea \mathcal{P} un programa lógico rebatible y h un literal. Diremos que h es un *literal garantizado* a partir de \mathcal{P} si y solo si existe un argumento $\langle \mathcal{A}, h \rangle$ tal que es un argumento aceptado de \mathcal{P} .

Esto es, sea h un literal que representa una consulta para un programa \mathcal{P} . La respuesta a h será:

- SI: si h está garantizado en \mathcal{P} .
- NO: si \bar{h} está garantizado en \mathcal{P} .

- INDECISO: si h no está garantizado en \mathcal{P} y si \bar{h} no está garantizado en \mathcal{P} .
- DESCONOCIDO: si h no pertenece al lenguaje del programa \mathcal{P} .

■

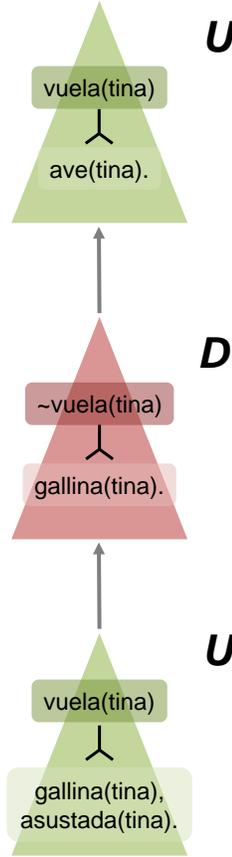


Figura 3.7: Árbol de dialéctica marcado $\mathcal{T}_{\langle \mathcal{A}, vuela(tina) \rangle}^*$ correspondiente al Ejemplo 3.1.

Por ejemplo, consideremos el programa lógico rebatible $\mathcal{P}_{3.1}$ del Ejemplo 3.1 y el argumento $\langle \mathcal{A}, vuela(tina) \rangle$ construido a partir de dicho programa. Como se ilustra en la Figura 3.7, la raíz del árbol de dialéctica marcado $\mathcal{T}_{\langle \mathcal{A}, vuela(tina) \rangle}^*$ está etiquetada como U, implicando que $\langle \mathcal{A}, vuela(tina) \rangle$ es un argumento aceptado de $\mathcal{P}_{3.1}$. En consecuencia, se puede concluir que “ $vuela(tina)$ ” pertenece a las inferencias del programa por ser un literal garantizado a partir de $\mathcal{P}_{3.1}$.

3.3. Argumentación con soporte de base de datos (DBI-DeLP)

Como se mencionó secciones previas, DeLP proporciona un marco adecuado para crear aplicaciones del mundo real que tratan con información incompleta y potencialmente contradictoria, pero este tipo de aplicaciones generalmente produce grandes cantidades de datos que normalmente se almacenan en bases de datos. Claramente, la actualización del conocimientos de manera continua es ineficiente debido a la necesidad de una codificación explícita de las reglas y los hechos en el programa. Sin embargo, estos nuevos datos deben considerarse cuando se crean nuevos argumentos y contra-argumentos, como parte de la evolución del conocimiento del sistema. Esto conduce a una limitación práctica con respecto al volumen de datos en los que se admite la argumentación [CCS05]. Debido a este problema, recientemente ha habido un interés creciente en conectar enfoques de argumentación con repositorios de datos masivos, como las bases de datos relacionales o la Web [Rah08, WBCC09, LBR12, BLSR13, DDE⁺17]

En esta sección se presentará el formalismo de *Bases de Datos Integradas para la Programación en Lógica Rebatible* (DBI-DeLP) (por su nombre en inglés, *Database Integration for Defeasible Logic Programming*) introducido en [DDG⁺13]. Este formalismo permite el razonamiento de sentido común sobre los datos almacenados en bases de datos que pueden ser actualizadas por otras aplicaciones externas, lo que permite la ejecución de procesos de argumentación basados en fuentes externas de datos masivas, otorgando la posibilidad de dar respuestas mucho más precisas y justas a las consultas que reciben.

Para esto, DBI-DeLP utiliza DeLP para manejar el proceso de argumentación, alimentándolo con la información de las fuentes de datos del dominio disponibles. Al igual que con DeLP, en DBI-DeLP se consideran dos tipos de información: aquella que se considera como conocimiento estricto (que debe preservar la coherencia interna, es decir, no puede ser contradictoria) y aquella que es tentativa (que se puede usar si no se puede encontrar nada en su contra). Claramente, la información obtenida mediante el uso de reglas rebatibles es tentativa y no necesita ser coherente con otras obtenidas de esta misma manera. Se puede derivar un literal y su complemento, permitiendo la inclusión, en una base de conocimiento, de información contradictoria.

Dado que la información almacenada en una base de datos es potencialmente contradictoria (tanto en el sentido de datos opuestos como de datos que pueden llevar a

conclusiones complementarias) y dado que no se quiere restringir el contenido en este sentido, el conocimiento obtenido de la base de datos se representa como una *presuposición*, en lugar de representarse como hechos. Como ya se mencionó en la Sección 3.2.1, una *presuposición* se denota como “*Cabeza* \rightarrow ” y se interpreta como “existen razones tentativas para creer en *Cabeza*”.

Básicamente, un programa de DBI-DeLP es un programa DeLP ampliado con la información obtenida de las bases de datos. El conocimiento sobre un determinado dominio obtenido desde diferentes fuentes de datos puede ser contradictorio. Dado que un sistema basado en DBI-DeLP puede usar la información provista por varias entidades a través de sus bases de datos, el uso de hechos para representarlos no es posible porque puede llevar a inconsistencias en el conjunto de conocimiento estricto Π . Debido a eso, DBI-DeLP define las *presuposiciones operativas*, que son *presuposiciones* usadas para representar tal conocimiento.

Definición 3.27 (*Presuposición operativa*)

Una *presuposición operativa* es un literal, es decir, un átomo fijo o un átomo fijo negado, denominado “*Cabeza* \rightarrow ” tal que *Cabeza* representa información almacenada en una base de datos relacional utilizada por el sistema. ■

Un programa DeLP es extendido para incluir información obtenida desde las bases de datos como *presuposiciones operativas*. Un programa DBI-DeLP representa un programa DeLP como se define en la Sección 3.2.1 junto con un conjunto Σ de *presuposiciones operativas*, asociadas con los registros de las bases de datos utilizadas en el sistema. Tales *presuposiciones operativas* se recuperan por demanda por el sistema para resolver una consulta particular y luego se desechan después de obtener la respuesta final. Cabe mencionar que, mientras que el conjunto Π , de hechos y reglas estrictas y el conjunto Δ , de reglas rebatibles, se mantienen fijos en la lógica del programa, en cambio el conjunto Σ de *presuposiciones operativas* cambia cuando las bases de datos que representa son modificadas.

Definición 3.28 (*Programa DBI-DeLP*)

Sea $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$ un conjunto de bases de datos, $\mathcal{P} = (\Pi, \Delta)$ un programa DeLP, \mathcal{X} el conjunto de todos los predicados en las reglas de \mathcal{P} . Un programa DBI-DeLP es una tupla $\mathcal{P}' = (\Pi, \Delta, \Sigma)$, donde Σ es el conjunto de *presuposiciones operativas* para \mathcal{X} en \mathcal{D} . ■

Para ilustrar estos conceptos, se considera un sistema de recomendación de películas basado en argumentación que contiene reglas estrictas tales como:

$$\Pi = \left\{ \begin{array}{l} \text{restringido_menores}(\text{Pelicula}) \leftarrow \text{tiene_violencia}(\text{Pelicula}) \\ \dots \end{array} \right\}$$

y reglas rebatibles como:

$$\Delta = \left\{ \begin{array}{l} \text{tiene_violencia}(\text{Pelicula}) \rightsquigarrow \text{director}(\text{Pelicula}, \text{tarantino}) \\ \dots \end{array} \right\}$$

Dinamicamente, presunciones operativas como:

$$\Sigma = \left\{ \begin{array}{l} \text{genero_pelicula}(\text{piratas_del_caribe}, \text{comedia}) \rightsquigarrow \\ \text{genero_pelicula}(\text{piratas_del_caribe}, \text{accion}) \rightsquigarrow \\ \dots \end{array} \right\}$$

podrían añadirse si se encuentran diferentes géneros para la película *Piratas del Caribe* en un conjunto de datos al intentar resolver una consulta.

En pocas palabras, para obtener datos relevantes para el proceso de argumentación, se utilizan los elementos en el literal que el procedimiento dialéctico está tratando de garantizar para determinar qué registros de la base de datos están relacionados con ese literal, para esto se realizan las consultas SQL pertinentes. Finalmente, se transforman todos los resultados recuperados en presuposiciones operativas de tal manera que estén disponibles para el proceso de argumentación. Esta búsqueda dinámica de información relevante es crucial si se necesitan obtener datos pertinentes (por ejemplo, las películas que un determinado usuario del sistema ha visto) del universo de datos disponibles en el conjunto de datos.

3.4. Resumen

En este capítulo se introdujeron nociones correspondientes a argumentación como proceso de razonamiento argumentativo. En primer lugar se brindó una perspectiva histórica del estudio de argumentación, resaltando la utilidad de los sistemas argumentativos en

diferentes dominios de aplicación. Luego, en la Sección 3.1 se presentó la estructura conceptual introducida en [PV02], la cual caracteriza a los sistemas argumentativos mediante la definición de cinco elementos: un lenguaje lógico subyacente, la definición de la noción de argumento, la identificación de conflictos entre argumentos, la obtención de las derrotas entre argumentos y, finalmente, la determinación del estado de aceptabilidad de los argumentos. A partir de esta caracterización se analizó cómo la definición de estos elementos toma lugar durante los diferentes pasos del proceso argumentativo.

En la Sección 3.2 se presentó una alternativa para la formalización de sistemas argumentativos, los sistemas argumentativos basados en reglas. Como, alternativamente, también es posible formalizar un sistema argumentativo mediante marcos argumentativos abstractos [Dun95], los sistemas argumentativos basados en reglas usualmente son identificados como sistemas argumentativos “*concretos*”. Como caso particular de sistema argumentativo basado en reglas se optó por la Programación en Lógica Rebatible (DeLP) [GS04]. Cabe destacar que, particularmente, se optó por presentar éste sistema dado que será utilizado para los desarrollos de los Capítulos 5 y 6 de esta tesis.

En la Sección 3.2.1 se presentó DeLP, el formalismo propuesto en [GS04]. DeLP es un sistema argumentativo basado en reglas que combina resultados de programación en lógica y argumentación rebatible. Dada su caracterización como un sistema argumentativo basado en reglas, se mostró cómo DeLP instancia los diferentes elementos de la estructura conceptual presentada en la Sección 3.1. En primer lugar, se introdujo la sintaxis del lenguaje de representación a partir del cual se especifican los programas lógicos rebatibles. Luego, a partir de estos programas, se mostró cómo se efectúa la construcción de argumentos a partir de la noción de derivación. Dado el uso de la negación fuerte en la especificación de los programas, es posible obtener argumentos que se hallen en conflicto por sustentar conclusiones contradictorias. Para resolver estos conflictos DeLP utiliza un criterio de comparación modular, el cual asigna pesos a los argumentos involucrados en el conflicto. De esta manera, si un argumento ataca a otro y es más fuerte que el argumento atacado se produce una derrota. A partir de la consideración de todos los argumentos y las relaciones de derrota entre ellos, DeLP define un procedimiento de prueba dialéctico que permite determinar cuáles son los argumentos aceptados del sistema. Por último, esto permite identificar cuáles son las inferencias del sistema (literales garantizados a partir de un programa lógico rebatible), las cuales corresponden a conclusiones sustentadas por argumentos aceptados.

Finalmente, en la Sección 3.3 se introdujo el formalismo DBI-DeLP el cual permite que el formalismo DeLP resulte útil en aquellas aplicaciones del mundo real que manejan grandes volúmenes de información. Este formalismo permite la obtención automática y en tiempo real, a partir de bases de datos relacionales, de la información que almacena el sistema y la convierte en presuposiciones para que puedan ser utilizadas por DeLP.

Capítulo 4

Confiabilidad

La confiabilidad es, sin duda, una característica importante de la vida cotidiana. Sin un fondo de confiabilidad se sufriría una pérdida de eficiencia y dinamismo [GM75]. Quizás peor, sería muy difícil levantarse por la mañana [Luh18], y se sufriría el inevitable colapso de la sociedad [Lag92, Bok99]. En cambio, debido a la presencia de confiabilidad, se puede lograr mejor desempeño en las tareas, un mayor desarrollo personal y más sano [GM75], una comprensión, o al menos una aceptación, de la complejidad de la sociedad [Luh18], y además habilita la capacidad de cooperar [Deu62, Arg91, Sto02, DJE10, PH13]. La confiabilidad aumenta el deseo de las personas de tomar riesgos para un intercambio social productivo [Tyl01, BC08].

En este capítulo se realiza una revisión de diferentes consideraciones para la noción de confiabilidad. Además se exploran distintas alternativas del estado del arte de sistemas de recomendación, en general y en particular de noticias, y sistemas argumentativos considerando la noción de confiabilidad.

4.1. Definiendo confiabilidad

Hay muchos puntos de vista de la confiabilidad [Bar83, Sha87], y hay numerosas razones para esto. Dos de estas razones sobresalen para ser presentadas en particular, ya que tocan los principales problemas que obstaculizan la formalización del concepto.

En primer lugar, el problema de la confiabilidad fue abordado por muchos investigadores, siendo sus propuestas de naturaleza muy diversa. Por tal motivo, existe un gran

número de definiciones de confiabilidad. La segunda razón es más directa: la noción de confiabilidad se puede caracterizar desde distintos puntos de vista, porque hay muchos tipos de confiabilidad [Deu73, Sha87, Co01]. Dicho de otra manera, la confiabilidad se puede clasificar como esperanza, desesperación, confianza, inocencia e impulsividad, por nombrar algunos [Deu73, GM75]. De estos, solo algunos son “positivos” en términos de puntos de vista optimistas de lo que es probable que ocurra.

Con tantos tipos y puntos de vista como hay, también hay muchas disciplinas que estudian el concepto [MDS95, MC01, Wat05, Li07, SW14]. Por lo tanto, se menciona en disciplinas tan diversas como la biología evolutiva [Bat00], la sociología [Luh00, Luh18], la psicología social [Deu62, EK09], la economía [HAC90, Das00, MDM15], la historia [Gam00b, Pag00] y la filosofía [Her88, Lag92].

La confiabilidad es un fenómeno común. De hecho, se ha argumentado que el ser humano, ni siquiera sería capaz de enfrentar las complejidades del mundo sin recurrir a la confiabilidad, porque es con confiabilidad que se puede razonar sensatamente sobre las posibilidades de la vida cotidiana [Luh18]. Por ejemplo, una persona sale de la casa todas las mañanas confiando en que podrá regresar y que no terminará en el hospital debido a un accidente que confía que no sucederá. A pesar de su importancia, ha habido una falta de investigación detallada sobre el tema [GM75, Luh00, Luh18]. Además, el trabajo que se ha llevado a cabo presenta sus propios problemas, sobre todo debido a la ausencia de una definición de confiabilidad sólida y aceptada.

Como han señalado varios autores, la confiabilidad es un concepto que es complejo y al mismo tiempo difícil de precisar y, como resultado, hay varias definiciones diferentes en la literatura. Se pueden elegir algunos ejemplos específicos. Una definición de confiabilidad establece que:

- La confiabilidad ocurre cuando un individuo percibe un camino ambiguo, el resultado puede ser bueno o malo, y la ocurrencia del resultado bueno o malo es contingente en las acciones de otra persona; finalmente, el mal resultado es más perjudicial que lo que el buen resultado es de beneficioso. Si el individuo elige seguir ese camino, puede decirse que hizo una elección de confiabilidad, si no, desconfía [Deu62].

Esta definición es aceptable en la medida en que se muestre la estructura básica de una elección confiable. Sin embargo, hay desacuerdo en cuanto a la idea de que los beneficios

deberían ser menores que el daño causado. Existe otra definición similar [GM75], pero señala que “la pérdida o el dolor asociado al incumplimiento de la confiabilidad a veces se consideran mayores que la recompensa o el placer derivado de la confiabilidad cumplida”. Pero también existen otras definiciones, una de ellas sugiere que:

- La confiabilidad es una apuesta sobre las futuras acciones contingentes de los demás [Szt99].

Mientras que otra, basándose en un rango de definiciones existentes, define la confiabilidad como:

- La confiabilidad es la medida en que una parte está dispuesta a depender de algo o de alguien en una situación dada con un sentimiento de seguridad relativo, aun cuando las consecuencias negativas sean posibles [MC96].

También existe otra definición que estipula:

- La confiabilidad es la probabilidad subjetiva por la cual un individuo, A, espera que otro individuo, B, realice una acción dada de la cual depende su bienestar [Gam00a].

La vaguedad relativa de estas definiciones es útil porque hace al concepto de confiabilidad más general. Incluye explícita e implícitamente aspectos de una amplia noción de confiabilidad que depende de la entidad o parte confiable, la confiabilidad de la entidad o parte confiable, la utilidad en el sentido de que la utilidad positiva resultará de un resultado positivo y la utilidad negativa resultará de un resultado negativo, y finalmente una cierta actitud de riesgo en el sentido de que la parte que confía está dispuesta a aceptar el riesgo situacional resultante de los elementos anteriores. El riesgo surge, por ejemplo, cuando el valor en juego en una transacción es alto y la probabilidad de falla no es despreciable.

Si bien estas definiciones difieren, hay claramente algunos elementos comunes. Existe un grado de incertidumbre asociado con la confiabilidad, ya sea expresada como una probabilidad subjetiva, como una apuesta (que, por supuesto, puede expresarse como una probabilidad subjetiva [Jay03]), o como un “sentimiento de seguridad”. Se termina aceptando la idea de que:

- La confiabilidad implica cierto grado de incertidumbre en cuanto al resultado.

- La confiabilidad implica optimismo y esperanza en cuanto al resultado.

La confiabilidad está, por lo tanto, fuertemente ligada a tener esperanza, convicción o fe en algo, ya sea la persona en quien se confía, el medio ambiente o lo que sea de lo cual el resultado deseado depende. Se llega al concepto de confiabilidad como la elección de ponerse en manos de otros, ya que el comportamiento del otro determina lo que se obtiene de una situación. A continuación se presentan algunas características de la confiabilidad, en términos de consideraciones sociales, biológicas y tecnológicas.

Consideraciones sociales

En la sociedad, la confiabilidad es un hecho de la vida cotidiana [Deu73, Bai86, Yam90, Luh00, Luh18]. De hecho, sin confiabilidad, como sugiere la cita de apertura de este capítulo, las sociedades dejarían de existir [Lag92]. Hay muchos ejemplos en dónde la confiabilidad desempeña un papel explícito en las sociedades. Los individuos para poder realizar ciertas tareas u obtener ciertos servicios, como pueden ser visitar a un médico, ir al mecánico o necesitar de algún talento o habilidad especial con la que no cuentan, deben recurrir a otros individuos. Cuando se opta por uno y no por otro que ofrece lo mismo, podemos decir que se está tomando una decisión de confiabilidad por un determinado individuo en particular.

También, ocurre que a veces los individuos se organizan y trabajan en conjunto, generalmente se unen en grupos para promover algún objetivo particular [Sha81]. En este caso se puede decir que los individuos han tomado una decisión de confiabilidad hacia el grupo y “confían” en él para poder lograr su objetivo. La confiabilidad fomenta el florecimiento de grupos [DJE10]. Si bien se ha encontrado que los grupos se comportan de manera agresiva hacia los miembros de otros grupos (y del mismo grupo también) con los que compiten [Bro88], se puede considerar la cooperación entre grupos. Si se toma al grupo como una entidad individual, también podría tomar decisiones de confiabilidad hacia otros grupos, de esta manera podría existir la cooperación entre distintos grupos para poder plantear diferentes metas, posiblemente más complejas, como puede ser construir un edificio, formar un partido político o crear algún tipo de institución con un determinado fin. La confiabilidad impulsa la creación de organizaciones [ZMP98, DF01, MPZ03], y hasta de naciones [KK97, LC10].

Se puede ver que todas las decisiones de confiabilidad de cada individuo pueden desempeñar un papel importante en la formación, el comportamiento y la estructura de la sociedad en general. Que los individuos de la sociedad se levanten por la mañana es un signo de la confiabilidad que se tiene en la sociedad y en el entorno [Luh18].

Consideraciones biológicas

La conducta de confiabilidad no es un fenómeno exclusivamente humano, también los demás miembros del mundo biológico demuestran confiabilidad. Esto radica en gran medida en la reciprocidad retrasada en el mundo animal [Har91]. Los animales se ayudan unos a otros con la esperanza de que, en el futuro, cuando necesiten ayuda, reciban ayuda. Un buen ejemplo es el vampiro murciélago. Cuando esos murciélagos han tenido una buena noche y un exceso de sangre, alimentan a los que no [Har91]. Invariablemente, aquellos que se comportan de esta manera son alimentados por aquellos a quienes alimentan cuando ellos mismos tienen una mala noche de caza. Claramente este es un ejemplo de confiabilidad, el murciélago que alimenta a los demás muestra una forma primitiva de confiabilidad en aquellos a quienes alimenta, porque confía en que ellos tendrán un comportamiento similar en el futuro. Las situaciones que implican confiabilidad constituyen una subclase de aquellas que implican riesgo. Son situaciones en las que el riesgo que uno toma depende del rendimiento de otro actor. Por lo tanto, el murciélago muestra confiabilidad. Por supuesto, los murciélagos que continuamente colaboran recíprocamente como grupo obtienen una ventaja evolutiva sobre sus vecinos que no tienen este comportamiento, ya que aquellos que sí colaboran serán alimentados cuando necesiten alimentarse, aumentando así sus oportunidades de supervivencia (y las de su grupo) a largo plazo [Har91].

Hay otros ejemplos de reciprocidad retrasada, como los chimpancés que se ayudan unos a otros en las peleas [Tri85]. Existen argumentos para pensar que la confiabilidad es más probable que esté presente aquí, ya que la capacidad intelectual de los primates es mayor que la de muchos otros animales [Har91]. Gran parte del razonamiento detrás de la reciprocidad retrasada puede abordarse en términos de “supervivencia de los genes más aptos”. Los animales ayudan a quienes están relacionados consigo mismos porque así perpetúan las posibilidades de supervivencia de sus genes.

Consideraciones tecnológicas y/o artificiales

En el mundo moderno, las computadoras se están volviendo omnipresentes. Se puede decir que donde existe una sociedad, también lo hace la confiabilidad. Las sociedades artificiales ya existen, aunque de forma limitada. Un ejemplo de tal sociedad artificial es la red telefónica. Consiste en muchos nodos, cada uno de ellos de cierta inteligencia, cada uno de los cuales decide en qué camino enrutar el tráfico (llamadas telefónicas, faxes, etc.). El solo hecho de que estos nodos funcionen entre sí significa que la confiabilidad está presente.

Otro ejemplo es Internet. La tecnología en seguridad actual brinda cierta capacidad para generar un cierto nivel de confiabilidad en las comunicaciones. Por ejemplo, los algoritmos criptográficos para la privacidad y las firmas digitales, los protocolos de autenticación para probar la autenticidad y los métodos de control de acceso para administrar la autorización. Los nodos de red maliciosos a menudo incurren en problemas con la privacidad de la red y de los datos al distribuir claves públicas falsificadas. Para abordar este problema en [WSWW17] se propone un mecanismo que combina la distribución de claves con la administración de confiabilidad. Sin embargo, por lo general, estos métodos no pueden administrar el concepto más general de confiabilidad. Los algoritmos criptográficos, por ejemplo, no pueden decir si un código firmado digitalmente ha sido creado por programadores competentes y un certificado de clave pública firmado tampoco dice si el propietario es un espía industrial. Entonces podría decirse que, tal vez en estos ejemplos, se trata de una confiabilidad limitada. La confiabilidad es algo poco entendido en tales redes artificiales [WL92].

4.2. Confiabilidad y sistemas de recomendación

Los sistemas de recomendación se usan para estimar con precisión el grado al que un usuario en particular le gustará un elemento en particular. Existe una gran variedad de estos algoritmos [RV97, AT05, JZFF10, RRSK11, MS17]. Los métodos más utilizados para hacer recomendaciones, como se mostró en el Capítulo 2, son los basados en el contenido o los métodos de filtrado colaborativo. Los métodos basados en contenido sugieren elementos similares a los que el usuario indicó anteriormente que le gustaban. Por lo tanto, estos métodos tienden a tener su alcance de recomendaciones limitado a la vecindad inmediata

del historial de compras o el registro de calificación de los elementos del usuario. Por ejemplo, si un cliente de un servicio de alquiler de películas hasta el momento solo ha pedido películas románticas, el sistema solo podrá recomendar elementos relacionados y no explorará otros intereses del usuario. Los sistemas de recomendación también pueden mejorarse significativamente mediante el filtrado colaborativo. Esta técnica se basa en la detección de usuarios vecinos, es decir, usuarios cuyos gustos son similares a los del usuario que está en busca de recomendaciones. Las calificaciones de los vecinos son utilizadas para proveer al usuario de recomendaciones.

Como ya se vio en el Capítulo 2, los RS clásicos presentan diferentes limitaciones y debilidades, algunas de las cuales pueden aliviarse mediante técnicas que incorporan la noción de confiabilidad. Estas técnicas de recomendación avanzadas basadas en la confiabilidad se adhieren más al paradigma de filtrado colaborativo, en el sentido de que una recomendación para un elemento se basa en las calificaciones de otros usuarios para ese elemento, en lugar de en un análisis del contenido del elemento.

A pesar de las mejoras significativas en los enfoques de recomendación, aún quedan algunos problemas importantes. En [MA09, RRSK11, BOHG13] se discuten algunas de las debilidades de los sistemas de filtrado colaborativo. Por ejemplo, los usuarios suelen calificar o experimentar solo una pequeña fracción de los elementos disponibles, lo que hace que la matriz de calificaciones sea muy escasa (ya que un sistema de recomendación a menudo trata con millones de elementos). Por ejemplo, un conjunto de datos particular de *Epinions*¹ contiene más de 1.500.000 revisiones que recibieron alrededor de 25.000.000 de calificaciones de más de 160.000 usuarios diferentes [GKRT04, VCDCT08]. Debido a esta dispersión de datos, un algoritmo de filtrado colaborativo experimenta muchas dificultades al tratar de identificar buenos vecinos en el sistema. En consecuencia, la calidad de las recomendaciones generadas podría sufrir por esto. Además, también es un gran desafío generar buenas recomendaciones para los usuarios que son nuevos en el sistema (es decir, los usuarios con el problema de arranque en frío), ya que no han calificado un número significativo de elementos y, por lo tanto, no se pueden vincular correctamente con usuarios similares.

En [SS⁺01a, SS01b, MKL09] ha quedado demostrado que los usuarios prefieren sistemas más transparentes y que las personas tienden a confiar más en las recomendaciones de las personas en las que confían (“amigos”) que en los sistemas de recomendación en

¹<http://www.epinions.com>

línea que generan recomendaciones basadas en personas anónimas similares a ellos. En la vida real, una persona que quiere evitar un mal negocio, si se trata de la compra de algún elemento, o simplemente si no quiere perder el tiempo viendo una película que no será de su agrado, puede preguntarle a un amigo (es decir, a alguien en quien confía) qué piensa acerca de un determinado elemento i . Si este amigo no tiene una opinión sobre i , puede preguntarle a un amigo suyo, y así sucesivamente hasta que alguien con una opinión sobre i (es decir, un recomendador) haya sido encontrado. Los sistemas de recomendación basados en confiabilidad intentan simular este comportamiento. Una vez que se encuentra una ruta hacia una recomendación, el sistema puede combinar el juicio de esa recomendación con la información de confiabilidad disponible (a través de la propagación y agregación de confiabilidad) para obtener una recomendación personalizada. De esta manera, a partir de las relaciones de confiabilidad se va construyendo una especie de red de confiabilidad, la que permite llegar a más usuarios y a más elementos. En [HRW08] se identifican algunas de estas relaciones, diferenciando a los “amigos declarados” de los “amigos reales” en *Twitter*², al contar la cantidad de tweets que van dirigidos de un usuario a otro. En [DHX⁺16, SKLB17] se presentan RS que ofrecen sugerencias basadas en este tipo de relaciones (“amistad”). En [GZT14, YKSN18] otorgan diferentes pesos a la confiabilidad entre usuarios para generar recomendaciones de películas.

En la configuración de filtrado colaborativo de la Figura 4.1, los usuarios a y b estarán vinculados entre sí porque han otorgado calificaciones similares a ciertos elementos (entre los cuales se encuentra i_1), y análogamente, b y c pueden vincularse entre sí. En consecuencia, se puede hacer una predicción del interés de a en i_2 . Pero en este escenario no hay un vínculo entre a e i_3 o, en otras palabras, no hay manera de averiguar si i_3 sería una buena recomendación para el usuario a . Esta situación podría cambiar cuando se hayan establecido las relaciones de confiabilidad entre los usuarios del sistema de recomendación. Las líneas continuas en la Figura 4.1 indican relaciones de confiabilidad entre el usuario a y el usuario b , y entre b y el usuario c . Mientras que en un escenario sin una red de confiabilidad, un sistema de filtrado colaborativo no puede generar una predicción sobre i_3 para el usuario a , esto podría resolverse mediante la utilización de la confiabilidad: si a expresa un cierto nivel de confiabilidad en b , y b en c , por propagación se puede obtener una indicación de la confiabilidad de a en c . Si el resultado indicara que el usuario a debe confiar altamente en c , entonces i_3 podría convertirse en una buena recomendación para

²<https://twitter.com/>

a , y será altamente calificado entre los otros elementos recomendados.

Sin embargo, esto no quiere decir que la confiabilidad sea transitiva [GKRT04, SNP13, WCFHV17]. Debido a su naturaleza propagativa, la información de confiabilidad se puede pasar de un usuario a otro. La propiedad propagativa de la confiabilidad es mencionada en [JGK03], donde se utiliza en un sistema de recomendación para permitir la propagación de confiabilidad de acuerdo con condiciones explícitas. La propagación de la confiabilidad a través de distintos usuarios tiene un comportamiento similar a la propagación de la información “de boca en boca” [SM95, ARH00, FAM15].

Este sencillo ejemplo ilustra que aumentar un sistema de recomendación al incluir relaciones de confiabilidad puede ayudar a resolver el problema de escasez. Además, un sistema que incorpora la noción de confiabilidad, también alivia el problema del arranque en frío: se ha demostrado que al emitir unas pocas declaraciones de confiabilidad, en comparación con la misma cantidad de información de calificaciones, el sistema puede generar recomendaciones más precisas [MA04, GZYS15b, GZYS15a, YLLL16]. Finalmente, el funcionamiento de un sistema que incorpora la noción de confiabilidad (por ejemplo, el concepto de propagación de confiabilidad) es intuitivamente más comprensible para los usuarios que los clásicos enfoques de “caja negra”.

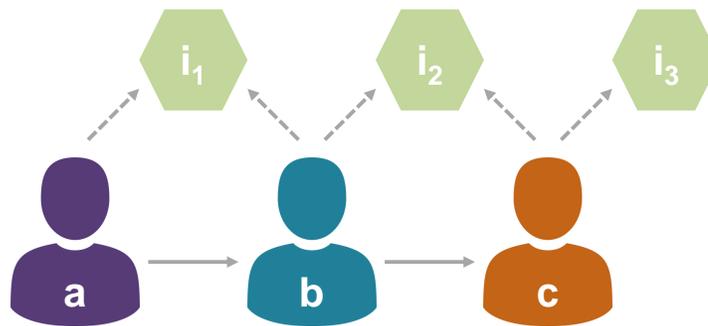


Figura 4.1: Relaciones de confiabilidad en un sistema de recomendación.

Como era de esperarse, ya se han hecho algunos intentos en esta dirección, por ejemplo [Gol05, PPK05, OS05, PM06, HS08, LHC08, MA09]. Los sistemas de recomendación mejorados con la noción de confiabilidad se pueden dividir aproximadamente en dos clases, de acuerdo con la forma en que se obtienen los valores de confiabilidad. El primer grupo utiliza información proveniente de una red de confiabilidad generada por la entrada directa de los usuarios, es decir, emitiendo explícitamente declaraciones de confiabilidad.

Se pueden encontrar ejemplos en [Gol06, HS08, MA09]. Dicha estrategia permite utilizar la propagación y agregación de confiabilidad en la red para inferir los valores de confiabilidad finales que se necesitan en el algoritmo de recomendación. Por otro lado, el segundo grupo no requiere que el usuario estime la confiabilidad en sus conocidos. En cambio, los valores de confiabilidad se calculan automáticamente, por ejemplo, en función del historial de un usuario de hacer recomendaciones confiables [OS05, LHC08] o en función de las reglas de transitividad para la similitud entre usuarios [PPK05]. En la literatura del comportamiento (*behavioral*), el concepto de confiabilidad está mejor definido. Por ejemplo, en [MDS95] se propone un marco en el que se determinan la capacidad, la benevolencia, la integridad y la propensión a confiar como sus factores clave, o en [McA95] se distingue entre la confiabilidad basada en la cognición y la basada en el afecto. Sin embargo, en el área de investigación de recomendaciones, la confiabilidad se utiliza a menudo como un término general para una amplia gama de relaciones entre los usuarios, especialmente cuando se trata del cálculo automático de los valores de confiabilidad. En estos casos, la confiabilidad se utiliza para denotar una variedad de conceptos, que van desde la similitud percibida de los gustos, sobre la reputación, hasta la evaluación de la competencia de un usuario.

4.3. Confiabilidad y argumentación

En los últimos años, diferentes modelos de confiabilidad han sido propuestos en la literatura. Las técnicas utilizadas pueden clasificarse ampliamente en técnicas estadísticas y de aprendizaje automático, técnicas basadas en heurísticas y técnicas basadas en comportamiento. Las técnicas estadísticas y de aprendizaje automático se centran en proporcionar un modelo matemático sólido para la gestión de la confiabilidad. Las técnicas basadas en heurística se centran en la definición de un modelo práctico para implementar sistemas de confiabilidad robustos. Los modelos basados en el comportamiento se centran en el comportamiento del usuario en la comunidad. Los sistemas bayesianos [MMH02, JI02] y los modelos de creencias [Jøs01, YS02, JHP06] son algunos ejemplos de técnicas puramente estadísticas. Normalmente, en los sistemas bayesianos, las calificaciones binarias (honestas o deshonestas) se utilizan para evaluar la confiabilidad mediante la actualización estadística de las funciones de densidad de probabilidad beta. En un modelo de creencia, la creencia de un consumidor con respecto a la verdad de una calificación también se incluye en el

cálculo de la confiabilidad. Las técnicas para combinar creencias varían. Por ejemplo, la teoría de Dempster-Shafer se emplea en [YS02], mientras que la lógica subjetiva se usa en [Jøs01, JHP06]. Las soluciones basadas en ML suelen utilizar técnicas como las redes neuronales artificiales y los modelos ocultos de Markov (HMM, del inglés *Hidden Markov Model*) para calcular y predecir la confiabilidad. Por ejemplo, en [SPX04] utilizan HMM para evaluar la confiabilidad del recomendador, y en [ESN09] proponen un modelo de confianza discreto basado en HMM. Debido a que tanto las soluciones estadísticas como las de ML son muy complejas, los investigadores han avanzado hacia soluciones basadas en heurísticas. Este tipo de soluciones [XL04, HJS06], pretenden definir un sistema de gestión de confiabilidad, práctico, robusto y fácil de entender. En [MAB09] se presenta una solución híbrida que define heurísticas claves y un modelo estadístico (HMM) para la evaluación de la reputación. En [AEG⁺10] se presenta un modelo basado en comportamiento, donde la confiabilidad se evalúa en función del comportamiento de comunicación de los miembros en una red social. La confiabilidad de comportamiento se evalúa según dos tipos de confiabilidad: confiabilidad de conversación y confiabilidad de propagación. La confiabilidad en la conversación especifica cuánto tiempo y con qué frecuencia se comunican dos miembros entre sí. Una comunicación más larga y frecuente indica más confiabilidad entre las dos partes. La confiabilidad de propagación se refiere a la propagación de información. La propagación de la información obtenida de un miembro a varios otros miembros indica que se está depositando un alto grado de confiabilidad en la información e, implícitamente, su fuente.

Gran parte del trabajo sobre la confiabilidad en informática se ha concentrado en tratar con escenarios específicos en los que la confiabilidad debe establecerse o manejarse de determinada manera. Se han realizado estudios sobre el desarrollo de la confiabilidad en el comercio electrónico mediante el uso de sistemas de reputación y estudios sobre el rendimiento de dichos sistemas [SS05, OBR⁺08]. Otro área de preocupación es la confiabilidad de la información y sus fuentes en la web, como la proporcionada por los sistemas de recomendación. Por ejemplo, en [VCVHV12], los autores relacionan la confiabilidad de la información con la longitud de las rutas que conectan a los usuarios. Los autores evaluaron varias longitudes de ruta incorporando estrategias de agregación para generar predicciones más precisas. También se han investigado mecanismos para determinar en qué fuentes confiar cuando se enfrentan con múltiples datos en conflicto [DBES09]. Esta idea fue extendida para calificar a las personas que brindan información, al observar la historia de los argumentos que han proporcionado [BV11]. Otros modelos se han aplicado en áreas como

la de las redes sociales [WBS08, RD16], el comercio electrónico [SL11, SL12, FZŞMT14] y los sistemas P2P [CDE08, YMS11, CLSQ15].

Computar la confiabilidad es un problema de razonamiento bajo incertidumbre, que requiere la predicción y anticipación, por parte de un agente (el evaluador), del comportamiento futuro de otro agente (el objetivo). Por lo tanto, el enfoque de argumentación se puede utilizar para mejorar el desempeño de los mecanismos de confiabilidad. Si bien la literatura sobre confiabilidad es abundante, el trabajo previo sobre argumentación y confiabilidad es mucho más escaso. Dada la reconocida capacidad de la argumentación para apoyar el razonamiento bajo incertidumbre [Amg03, RA06], existen algunas propuestas que han considerado el uso de argumentos para computar la confiabilidad. Una de éstas, propone un enfoque basado en argumentación para la evaluación de confiabilidad que es bipolar (separando los argumentos para la confiabilidad y para la desconfianza) y cualitativo (ya que los argumentos pueden apoyar varios grados de confiabilidad/desconfianza) [Pra07]. Otra utiliza argumentación lógica donde los argumentos apoyan las mediciones de confiabilidad, por ejemplo mediciones cualitativas tales como “muy confiable” o “poco confiable” [PTS⁺11]. Un modelo de confiabilidad que aplica lógica temporal rebatible se presenta en [KB12]. Otra propuesta presenta un marco formal para modelar cómo las diferentes dimensiones de la confiabilidad percibida de la fuente interactúan para determinar la aceptabilidad de un mensaje, y cómo las desviaciones de tal expectativa producen una retroalimentación específica sobre la confiabilidad de la fuente [VBGVDT13]. En [BV14] se propone un modelo de confiabilidad difuso basado en argumentación para un sistema de recomendación multi-agente. En [GTGS18] se propone un marco argumentativo con ataques recursivos para definir un modelo de confiabilidad para un sistema multi-agente, colaborativo y abierto.

La investigación en modelos de confiabilidad se ha centrado principalmente en calcular el grado de confiabilidad que un agente tendrá en otro agente si no existe una relación de confiabilidad explícita entre ellos [GKRT04, BK06, JE10]. En la mayoría de estos enfoques, los valores de confiabilidad se representan dentro del rango $[-1, 1]$. Por otro lado, las caracterizaciones formales de la noción de confiabilidad se han investigado utilizando lógicas y semánticas de confiabilidad [CNS03, HF06, Mul11], mientras que la argumentación se ha utilizado para razonar sobre la confiabilidad en [PMS10, VBGVDT11, PAL⁺14, SPL⁺16, MPB16]. El núcleo de este enfoque es calcular de manera eficiente un modelo de reputación basado en la experiencia directa de los

agentes y los informes proporcionados por otros.

Utilizando argumentación es posible obtener metodologías para razonar sobre la confiabilidad. La argumentación puede proporcionar un mecanismo para manejar muchos de los aspectos que se necesitan capturar acerca de la confiabilidad, tal como se analizará con mayor detalle más adelante en este capítulo.

4.4. Confiabilidad y noticias en la web

Los servicios de noticias web son especialmente atractivos porque, de forma diferente a los periódicos impresos, las noticias en línea se pueden entregar de manera rápida y personalizada. Proveedores de noticias de múltiples fuentes en la web, como *Google News*³, *Yahoo! News*⁴ y *MSNBC*⁵ integran noticias de varias fuentes y las entregan según factores como la popularidad de las noticias, las fuentes, la frescura, la ubicación geográfica del lector y los intereses del lector. También hay que considerar a las redes sociales como *Twitter* o *Facebook* que funcionan como distribuidoras de noticias, al punto que han llegado a convertirse en la principal fuente de información para muchas personas [HB18]. El desarrollo de un algoritmo para la selección y clasificación de noticias en la web es muy difícil porque requiere la combinación de muchos aspectos, a veces complicados. Como consecuencia, el problema de clasificar y recomendar noticias web ha atraído mucha atención de investigación en los últimos años. Hay varias razones por las que las medidas de la autoridad de página como el PageRank [BP98b] no se pueden aplicar directamente en el momento de la clasificación de noticias web. A diferencia de lo que sucede con las páginas web, los periódicos de Internet rara vez utilizan enlaces. Además, las noticias de última hora suelen tener prioridad sobre las noticias anteriores porque los lectores prefieren ver información sobre los eventos de noticias tan pronto como tienen lugar. Sin embargo, las noticias nuevas generalmente tienen muy pocos enlaces entrantes, lo que impide la aplicación de algoritmos de análisis de enlaces para favorecer las noticias nuevas sobre las obsoletas. Sin embargo, un modelo de clasificación que otorga alta prioridad a las nuevas noticias tendrá algunas deficiencias. Los informes sobre noticias recientes tienden a ser incompletos y muchas historias presentadas como noticias de última hora se revisan

³<https://news.google.com>

⁴<https://news.yahoo.com/>

⁵<https://www.msn.com>

cuando se dispone de información adicional. Otro enfoque para clasificar las noticias podría basarse en la popularidad de las noticias, estimado al monitorear el número de lectores que acceden a una noticia o por un sistema de votación de sus historias favoritas. Sin embargo, la popularidad de las noticias puede no reflejar el valor real que una noticia tiene para los lectores individuales.

Generalmente, el lector debe decidir si vale la pena leer una noticia y si los hechos descritos en ella son creíbles. Es posible que la información que se proporciona en las noticias no siempre sea completamente verificable y, por lo tanto, otro factor importante que puede ayudar a seleccionar noticias es la confiabilidad o la credibilidad. A pesar del volumen de investigación sobre este tema, no existe una definición o medida de confiabilidad acordada en los medios de comunicación [Fis16]. El nivel de confiabilidad que un lector tiene en una noticia no está necesariamente asociado con las medidas de autoridad o popularidad de las noticias, e incluso puede tener una correlación negativa con la novedad de las mismas. Un servicio de recomendación de noticias que utiliza un sistema de administración de confiabilidad puede ayudar al lector a tomar la decisión al seleccionar noticias de fuentes confiables o en base a las opiniones de otros lectores confiables. Por lo tanto, la confiabilidad es un aspecto importante en la implementación de los sistemas de recomendación en general, y los recomendadores de noticias en la web en particular. Normalmente, la noción de confiabilidad se define en términos de dos componentes: intenciones de confiabilidad y creencias de confiabilidad. Por ejemplo, un usuario del sistema puede confiar en las intenciones de un proveedor o las intenciones de un servicio proveedor de información. Por otro lado, un usuario puede confiar en las creencias de otros usuarios.

Como se mencionó anteriormente, en los últimos años se han definido distintos modelos de confiabilidad [RD16], algunos de los cuales se han aplicado en el área de recomendación de noticias en la web. Por ejemplo, en [NSKA06] se propone un método para calificar la confiabilidad de las noticias utilizando algoritmos que comparan el contenido de diferentes fuentes de noticias. PolyNews [PKS06] es un marco de servicios de noticias que intenta mitigar el efecto del sesgo de los medios mediante la creación de múltiples puntos de vista clasificados. NewsTrust [new] fue un servicio creado para evaluar noticias donde los usuarios podían clasificar noticias, redactores de noticias y fuentes de noticias. Un sistema de recomendación de múltiples capas para publicaciones basadas en confiabilidad se propone en [Hes06]. Otro sistema, PersoNews [BKB⁺06] utiliza un clasificador Naïve bayesiano para proporcionar un lector de contenido RSS (del inglés *Really Simple Syndication*) per-

sonalizado. Un enfoque basado en argumentos para generar recomendaciones de agentes se propone en [BA10]. Aunque este trabajo se relaciona con la propuesta de utilizar reglas rebatibles para generar recomendaciones presentada en el Capítulo 5, no aborda el problema de la recomendación de noticias y se enfoca en entornos de inteligencia ambiental, sin intentar modelar la noción de confiabilidad. En [WLL15] a partir de la identificación de factores sociales contextuales que influyen en las relaciones de confiabilidad, se propone un enfoque probabilístico para la inferencia de confiabilidad en las redes sociales. Debido a la proliferación de noticias falsas (en inglés, *fake news*) que se dio principalmente en las redes sociales, se han definido propuestas basadas en la confiabilidad para hacerle frente a este problema [GYBO18].

4.5. Resumen

En este capítulo se hizo referencia a la noción de confiabilidad. En la Sección 4.1 se mostraron diferentes definiciones de la noción de confiabilidad, argumentando el porqué aún no se ha llegado a un acuerdo en cómo definirla en el área de sistemas de recomendación. Además se analizaron distintas consideraciones, tanto sociales, biológicas como tecnológicas que contemplan a la confiabilidad, a modo de justificación del porqué es una noción fundamental para contemplar, en este caso particular, al momento de diseñar un sistema de recomendación.

En la Sección 4.2 y la Sección 4.3 se analizaron algunas de las propuestas del estado del arte que consideran a la confiabilidad en sistemas de recomendación y en argumentación respectivamente. En la Sección 4.4 se hizo hincapié puntualmente en los distintos avances de sistemas de recomendación de noticias basados en confiabilidad.

Capítulo 5

Modelo de confiabilidad para la recomendación de noticias

Los RS tienen el objetivo de sugerir a cada usuario los elementos que podrían interesarle. En particular, los RS basadas en filtrado colaborativo (CF) se basan en las opiniones expresadas por los demás usuarios. De hecho, CF intenta encontrar automáticamente usuarios similares al usuario activo y recomienda los elementos que le gustan a estos usuarios similares. Esta simple intuición es eficaz para generar recomendaciones y es ampliamente usada. Sin embargo, los RS basados en CF sufren algunas debilidades inherentes que son intrínsecas en el proceso de encontrar usuarios similares. De hecho, el proceso de comparar dos usuarios con el objetivo de calcular su similitud implica comparar las calificaciones que proporcionaron a los distintos elementos. Para ser comparables, es necesario que los dos usuarios califiquen al menos algunos elementos en común. Sin embargo, en un dominio típico, por ejemplo en el dominio de películas, música, libros, noticias, etc. el número de elementos es muy grande (en el orden de millones), mientras que el número de elementos calificados por cada usuario es en general pequeño (en el orden de decenas o menos). Esto significa que es muy poco probable que dos usuarios aleatorios hayan calificado algún elemento en común y, por lo tanto, sean comparables.

Para superar esta debilidad, se propone explotar la información de confiabilidad expresada explícitamente por los usuarios. Los usuarios pueden indicar si confían o no en otros usuarios. En el contexto de RS, se relaciona con lo valiosas y relevantes que serán las calificaciones proporcionadas por un determinado usuario. Esta información adicional (declaraciones de confiabilidad) se puede organizar en una red de confianza. En este

enfoque, la idea no es buscar usuarios similares como hace CF, sino buscar usuarios confiables mediante la explotación de la propagación de la confiabilidad a través de la red de confiabilidad. Los elementos apreciados por estos usuarios se recomiendan al usuario activo.

En este capítulo, partiendo del trabajo realizado en [BSCM11, BCM13], se hace un análisis de distintas propiedades deseables de la confiabilidad. Además, se propone un modelo de representación y propagación de confiabilidad. Finalmente se estudia su idoneidad mediante un conjunto de ejemplos representativos y un análisis teórico de las principales características de la confiabilidad.

A diferencia de los modelos mencionados en el Capítulo 4, se presentará un modelo de representación de confiabilidad utilizando DeLP como herramienta. Un ingrediente clave del modelo propuesto es la propagación de confiabilidad basada en mecanismos de inferencia. La confiabilidad se modela a través de una serie de postulados básicos, que se pueden extender o relajar fácilmente. Los postulados se traducen en reglas rebatibles DeLP, que permiten inferir conclusiones tentativas con respecto a la confiabilidad de alguna entidad del sistema.

Con el fin de brindar una justificación de la aptitud de DeLP como herramienta para representar confiabilidad, se escogió realizar un recomendador de noticias basado en confiabilidad. Se propone un modelo capaz de lidiar, no solo con la representación, sino también con la propagación de confiabilidad.

5.1. Caracterización del modelo

El modelo propuesto para la recomendación de noticias basado en confiabilidad toma como punto de partida un conjunto de postulados definido previamente en [SMCS09]. El sistema trata con tres entidades diferentes: noticias, fuentes, tópicos y lectores.

- **Noticia.** Una noticia o un artículo es una comunicación escrita de un evento de noticias preparado por una agencia de noticias específica (fuente). Cuando una noticia está disponible en la web, podemos identificar algunos de sus atributos como lo son el título, la fuente, la hora, la descripción, la categoría y el enlace al contenido de las noticias. Otra información relacionada con la noticia, como el autor, también se puede obtener en ciertas situaciones.

- **Fuente.** La fuente de una noticia o artículo de noticias es la agencia a cargo de suministrar el informe que utilizarán los medios de comunicación. Las noticias también pueden ser publicadas por redes sociales, páginas web o blogs.
- **Tópico.** El tópico o tema de una noticia hace referencia al contenido de la misma o de lo que se habla en ella. Los tópicos pueden definirse con diferente granularidad, describiendo categorías más genéricas o más específicas en las cuales clasificar a las noticias.
- **Lector.** Un lector es un usuario del servicio de noticias. El sistema mantiene un grupo de lectores. Los lectores también pueden proporcionar declaraciones de confiabilidad sobre las distintas entidades del sistema, como lo son las noticias, las fuentes y también otros lectores.

Se han identificado relaciones fundamentales entre estas entidades, necesarias para modelar el concepto de confiabilidad, que se denominan declaraciones de confiabilidad. Una declaración de confiabilidad puede ser positiva o negativa (desconfianza) y, es una afirmación explícita del hecho de que un lector confía (desconfía) en una noticia, una fuente u otro lector. Estas declaraciones permiten inferir relaciones de confiabilidad implícitas, que son útiles para proporcionar recomendaciones al lector basadas en la confiabilidad.

Se propone el siguiente conjunto de postulados para modelar la noción de confiabilidad entre los lectores, las noticias y las fuentes de noticias de una manera intuitiva.

- **Postulado 1.** Una noticia proveniente de una fuente confiable normalmente será de confianza.
- **Postulado 2.** Una noticia proveniente de una fuente no confiable normalmente no será de confianza.
- **Postulado 3.** Una noticia en la que confíe un lector de confianza normalmente será de confianza.
- **Postulado 4.** Una noticia en la que no confíe un lector de confianza normalmente no será de confianza.
- **Postulado 5.** Una fuente en la que un lector de confianza confíe normalmente será de confianza.

- **Postulado 6.** Una fuente en la que un lector de confianza no confíe normalmente no será de confianza.

Una situación interesante surgirá cuando se puedan llegar a dos conclusiones conflictivas. Por ejemplo, un lector de confianza desconfía de una noticia, pero la noticia fue publicada por una fuente confiable. Suponiendo que, en general, se prefiere basar la opinión en información proporcionada por fuentes confiables, se pueden agregar los siguientes dos postulados:

- **Postulado 7.** Una noticia proveniente de una fuente confiable generalmente será de confianza, incluso si un lector confiable no confía en dicha noticia.
- **Postulado 8.** Una noticia proveniente de una fuente no confiable generalmente no será de confianza, incluso si un lector confiable confía en dicha noticia.

Otra consideración importante es que la confiabilidad depende del contexto, lo que significa que se puede optar por confiar o desconfiar de ciertas fuentes o lectores según el tema de la noticia en cuestión. Por ejemplo, se puede confiar en una noticia que informe sobre un accidente en una determinada ciudad si la noticia proviene de una fuente de noticias local a esa ciudad, pero es posible que no se confíe en la misma fuente si también informa sobre otros temas. Del mismo modo, se puede confiar en los lectores que tienen experiencia reconocida en ciertas áreas pero no en otras. Esto se puede expresar de la siguiente manera:

- **Postulado 9.** Una noticia sobre un tema en particular generalmente será de confianza si la fuente de la noticia es confiable para ese tema.
- **Postulado 10.** Una noticia sobre un tema en particular generalmente no será confianza si la fuente de la noticia no es confiable para ese tema.
- **Postulado 11.** Una noticia sobre un tema en particular, en la que confía otro lector, generalmente será de confianza si el lector es confiable para ese tema.
- **Postulado 12.** Una noticia sobre un tema en particular generalmente no será de confianza si la fuente de la noticia no es confiable para ese tema.

Se podrían agregar postulados adicionales. Por ejemplo, si algún lector tiene una muy buena reputación para verificar los hechos, se podría preferir confiar en la opinión de este, incluso si está en conflicto con las opiniones de otros lectores. Otros postulados que podrían agregarse a la lista pueden incluir referencias a la fecha y hora de la noticia. Por ejemplo, se confiará en una noticia más reciente en lugar de una desactualizada. La lista de postulados podría extenderse indefinidamente, incluidas las referencias al autor de las noticias, el país de origen de la fuente, las categorías de noticias, etc. Algunos postulados podrían personalizarse, ya que los lectores pueden estar en desacuerdo sobre la conclusión que se debe adoptar en función de ciertos hechos. En aras de la simplicidad, se tomarán los postulados 1 a 12 como los postulados centrales del modelo de administración de confiabilidad.

5.1.1. Usando DeLP para modelar confiabilidad en noticias

Como se ha mencionado en el Capítulo 3, la programación lógica rebatible (DeLP, del inglés *Defeasible Logic Programming*) [GS04] es un formalismo de argumentación rebatible de propósito general basado en la programación lógica, destinado a modelar conocimiento inconsistente y potencialmente contradictorio. El formalismo proporciona un lenguaje de representación del conocimiento que brinda la posibilidad de representar información tentativa de manera declarativa, y un mecanismo de razonamiento que considera todas las formas en que se podría soportar una conclusión y decide cuál tiene el mejor soporte. Una característica interesante de DeLP es que las conclusiones obtenidas a través de este formalismo pueden explicarse fácilmente mediante el proceso de razonamiento argumentativo. Además, DeLP se ocupa del razonamiento falaz, resolviendo situaciones que son problemáticas para muchos formalismos de argumentación, como se muestra en [GS04]. Los 12 postulados definidos anteriormente pueden traducirse en las siguientes reglas DeLP:

$$\begin{aligned}
 \text{trust_report}(V, R) &\prec \text{report_source}(R, S), & (R_1) \\
 &\text{trust_source}(V, S). \\
 \sim\text{trust_report}(V, R) &\prec \text{report_source}(R, S), & (R_2) \\
 &\sim\text{trust_source}(V, S).
 \end{aligned}$$

$$\begin{aligned}
trust_report(V, R) &\leftarrow trust_viewer(V, V_1), & (R_3) \\
&trust_report(V_1, R). \\
\sim trust_report(V, R) &\leftarrow trust_viewer(V, V_1), & (R_4) \\
&\sim trust_report(V_1, R). \\
trust_source(V, S) &\leftarrow trust_viewer(V, V_1), & (R_5) \\
&trust_source(V_1, S). \\
\sim trust_source(V, S) &\leftarrow trust_viewer(V, V_1), & (R_6) \\
&\sim trust_source(V_1, S). \\
trust_report(V, R) &\leftarrow report_source(R, S), & (R_7) \\
&trust_source(V, S), \\
&trust_viewer(V, V_1), \\
&\sim trust_report(V_1, R). \\
\sim trust_report(V, R) &\leftarrow report_source(R, S), & (R_8) \\
&\sim trust_source(V, S), \\
&trust_viewer(V, V_1), \\
&trust_report(V_1, R). \\
trust_report(V, R) &\leftarrow is_about(R, T), & (R_9) \\
&report_source(R, S), \\
&trust_source_topic(V, S, T). \\
\sim trust_report(V, R) &\leftarrow is_about(R, T), & (R_{10}) \\
&report_source(R, S), \\
&\sim trust_source_topic(V, S, T). \\
trust_report(V, R) &\leftarrow is_about(R, T), & (R_{11}) \\
&trust_viewer_topic(V, V_1, T), \\
&trust_report(V_1, R). \\
\sim trust_report(V, R) &\leftarrow is_about(R, T), & (R_{12}) \\
&trust_viewer_topic(V, V_1, T), \\
&\sim trust_report(V_1, R).
\end{aligned}$$

Para un lector en particular v , y en función de las reglas y los hechos de DeLP correspondientes, las noticias se clasificarán en cuatro conjuntos:

- **Noticias Confiables.** Aquellas noticias r_i para las cuales existe al menos un argumento garantizado que soporta $trust_report(v, r_i)$.
- **Noticias No Confiables.** Aquellas noticias r_i para las cuales existe al menos un argumento garantizado que soporta $\sim trust_report(v, r_i)$.

- **Indeciso.** Aquellas noticias r_i para las cuales no existe ningún argumento garantizado para $trust_report(v, r_i)$ o $\sim trust_report(v, r_i)$.
- **Desconocido.** Aquellas noticias r_i para las cuales no existe información para $trust_report(v, r_i)$ o $\sim trust_report(v, r_i)$.

Esta clasificación permitirá que el lector se centre en las noticias que se consideran confiables, y que se le advierta sobre las que no lo son.

5.2. Aptitud de DeLP para modelar confiabilidad

Como se ha señalado en estudios anteriores, no existen criterios generales para emitir una valoración objetiva de la confiabilidad [Ben11]. Aunque esto impide llevar a cabo una evaluación empírica del modelo definido es posible analizar su aptitud por medio de un conjunto de ejemplos representativos y un análisis teórico de sus características principales. Por lo tanto, el objetivo de esta sección es proporcionar evidencia para justificar la aptitud de DeLP como un mecanismo para el modelado de la noción de confiabilidad. Esto se logra mediante la presentación de una selección de un conjunto de ejemplos que ilustran la operación del enfoque propuesto y mediante la realización de un análisis teórico de algunas de sus propiedades.

5.2.1. Algunos ejemplos seleccionados

Supongamos que Ana es uno de los usuarios del sistema. El sistema cuenta con el modelo de representación de confiabilidad antes descrito, y Ana tiene ya expresado sus valores de confiabilidad sobre otros usuarios, diversas fuentes de noticias y sobre algunas noticias. Consideremos el ejemplo que llamaremos “google_hits_one_billion”, que se refiere a una noticia informando que *Google* recibió 1000 millones de visitantes únicos en mayo de 2011. Supongamos que después de ser traducida en hechos DeLP, la información de confiabilidad del sistema está representada por los siguientes hechos:

```
report_source(google_hits_one_billion, slashdot)
report_source(facebook_hits_one_billion, etc_news)
report_source(microsoft_hits_one_billion, msn_news)
```

```

trust_source(ana, slashdot)
~trust_source(cristian, etc_news)
~trust_report(marcela, google_hits_one_billion)
trust_report(marcela, facebook_hits_one_billion)
~trust_report(marcela, microsoft_hits_one_billion)
trust_report(cristian, microsoft_hits_one_billion)
trust_viewer(ana, emanuel)
trust_viewer(ana, cristian)
trust_viewer(emanuel, marcela)

```

Basado en esto y los postulados descritos anteriormente, el sistema es capaz de clasificar el valor de confiabilidad de una noticia para un determinado usuario como confiable, no confiable, indeciso o desconocido y utilizar esta información en el momento de la presentación de sugerencias al usuario. En este caso, supongamos que el sistema tiene que clasificar las noticias “google_hits_one_billion”, “facebook_hits_one_billion” y “microsoft_hits_one_billion”.

Para decidir cómo clasificar cada noticia, el sistema internamente trata de encontrar un argumento garantizado asociado con los valores de confianza acerca de cada uno de ellos. En la Figura 5.1 se puede ver que hay un argumento garantizado soportando la declaración *trust_report(ana, google_hits_one_billion)*, por lo que la noticia debe ser confiable para Ana. En la Figura 5.2 se puede ver la existencia de un argumento garantizado para *~trust_report(ana, facebook_hits_one_billion)*, por lo que Ana no debería confiar en esta noticia. Por último, la Figura 5.3 muestra que no se puede confiar en la noticia, porque no hay un argumento garantizado para *trust_report(ana, microsoft_hits_one_billion)*. Además, el sistema no puede concluir que Ana debería desconfiar de esta noticia porque no es posible encontrar un argumento garantizado para *~trust_report(ana, microsoft_hits_one_billion)*. Por lo tanto, el valor de confiabilidad de esta noticia para Ana se dice que es indeciso.

Mediante estos ejemplos se puede ver cómo la confiabilidad es propagada entre los usuarios y cuáles son los posibles escenarios para el estado epistémico de un usuario en relación con una noticia específica.

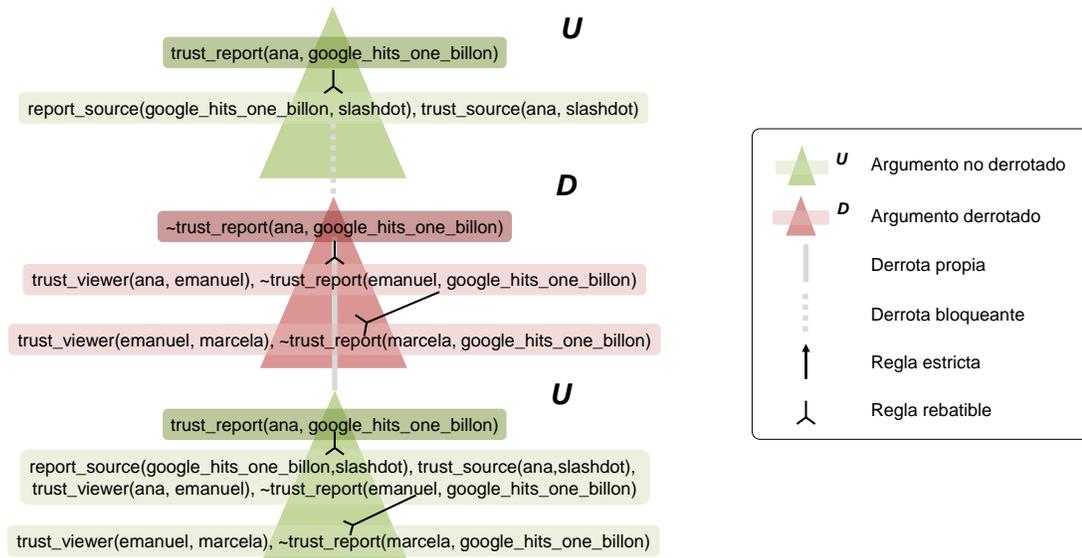


Figura 5.1: Árbol de dialéctica DeLP mostrando razones para confiar en la noticia “google_hits_one_billion”.

5.2.2. Propiedades de la confiabilidad y su caracterización usando DeLP

Un modelo adecuado de la noción de confiabilidad debe exhibir algunas propiedades deseables. En la actual sección se analizan estas propiedades, demostrando que DeLP es un marco natural para caracterizarlas.

La confiabilidad es subjetiva

Al igual que muchos aspectos de la Web, las noticias se están convirtiendo en una actividad colaborativa. Sin embargo, el juicio de la credibilidad de las noticias es idiosincrásico. Por lo tanto, los modelos que tratan con la confiabilidad como una noción objetiva son poco realistas. En este caso, la opinión que proviene de la “sabiduría de las multitudes” puede no ser tan útil como la opinión personal del lector, o la opinión de otro lector de confianza.

La naturaleza subjetiva de la confiabilidad conduce a la personalización del cálculo de la confiabilidad, donde los sesgos y preferencias de la confiabilidad tienen un impacto directo en el valor de la confiabilidad calculada. Varios modelos de confiabilidad consideran la personalización de la confiabilidad [SS05, WV07, WBS09, DHWW14].

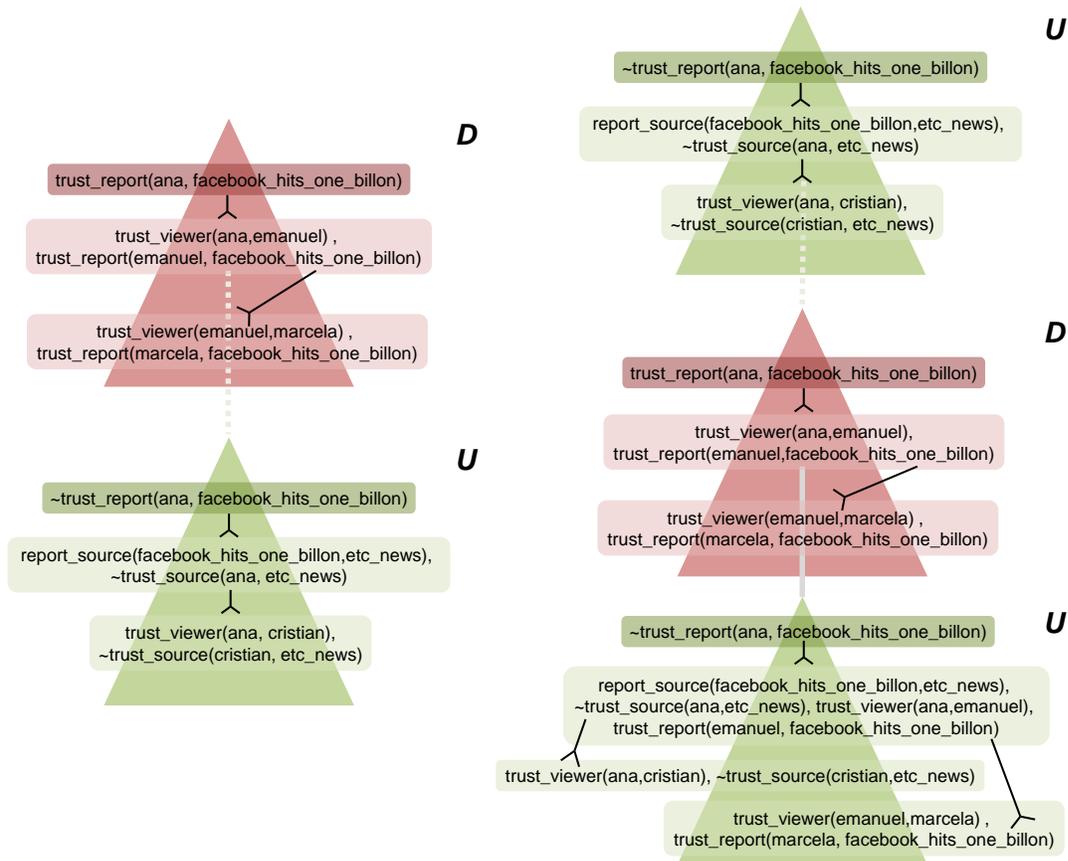


Figura 5.2: Árbol de dialéctica DeLP mostrando razones para no confiar en la noticia “facebook_hits_one_billion”.

En [Jac02], se destaca la importancia de la confiabilidad personalizada en las relaciones sociales y organizativas. En informática, la confiabilidad personalizada se ha estudiado en [ZM00, Sab02, YSS04, ELV13, MS16, SXL⁺17], entre otros.

En el ejemplo mencionado, se pueden identificar tres clases diferentes de estado epistémico asociado con los usuarios Ana, Cristian y Marcela con respecto a la confiabilidad o credibilidad de la noticia “microsoft_hits_one_billion”:

- La respuesta para $\text{trust_report}(\text{cristian}, \text{microsoft_hits_one_billion})$ es *sí*.
- La respuesta para $\text{trust_report}(\text{marcela}, \text{microsoft_hits_one_billion})$ es *no*.
- La respuesta para $\text{trust_report}(\text{ana}, \text{microsoft_hits_one_billion})$ es *indeciso*.

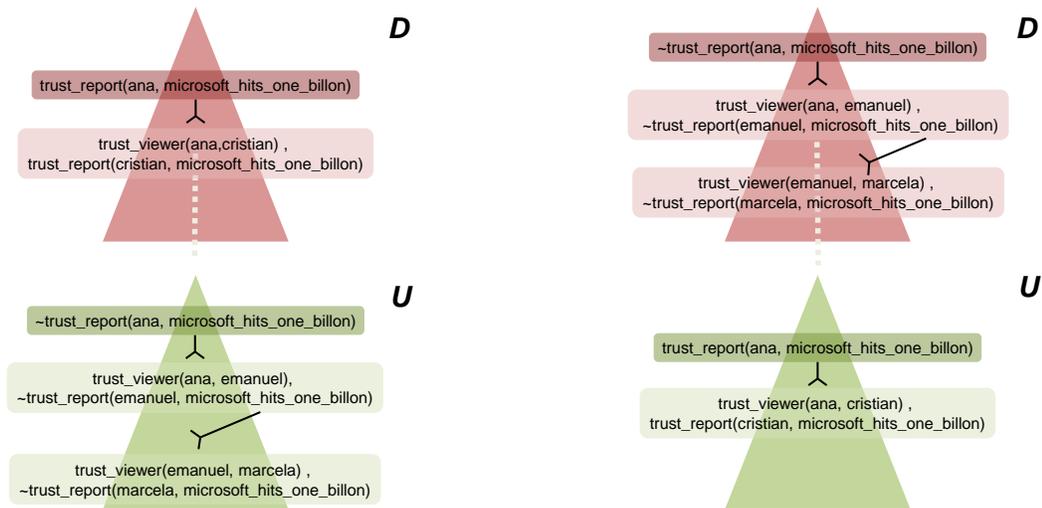


Figura 5.3: Árboles de dialéctica DeLP que no muestran razones para confiar o no confiar en “microsoft_hits_one_billion”.

La confiabilidad es dinámica y rebatible

La confiabilidad puede cambiar con nuevas experiencias (interacciones u observaciones) [SBL⁺04]. Las nuevas experiencias son más importantes que las antiguas, ya que las viejas experiencias pueden volverse obsoletas o irrelevantes con el tiempo. Esta propiedad de la confiabilidad está ampliamente modelada en la informática [THL13, ADMF⁺15, WSWW17, RVN18, GAH19]. Se utilizan diversas técnicas para modelar este dinamismo, por ejemplo, el envejecimiento de las interacciones antiguas en [KSGM03, WRIJ05], dando más peso a las interacciones recientes en [SHZK05, ZF07] y utilizando solo la interacción más reciente en [JF03]. Otra forma de reflejar este dinamismo, en la redes sociales, es a través del “perdón”, luego de haber realizado una ofensa [LAM18]. En algunos modelos, como PeerTrust [XL04], los usuarios pueden elegir la ventana temporal para el envejecimiento dinámico de las interacciones/experiencias antiguas según sus necesidades. En otros, como PowerTrust [ZH07], el cálculo de confiabilidad se realiza periódicamente para garantizar que los valores de confianza calculados estén actualizados.

En general, las agencias de noticias están sujetas a limitaciones de tiempo, lo que resulta en la publicación de reportes de noticias con información incompleta o inexacta. Por lo tanto, la confiabilidad en dichas noticias podría ser revocada por la existencia de otras más recientes. Mientras tanto, los reportes de noticias en los que un lector confía

podrían ser anulados por otras noticias que posean más autoridad (por ejemplo, procedente de alguna otra fuente de confianza). DeLP puede tratar efectivamente con la naturaleza rebatible de la confiabilidad. En el ejemplo, se mostró que:

- La respuesta para $trust_report(ana, facebook_hits_one_billion)$ es *no*.
- La respuesta para $trust_report(emanuel, facebook_hits_one_billion)$ es *sí*.

Sin embargo, si se agregan los hechos:

$$trust_source(ana, etc_news).$$

$$\sim trust_source(emanuel, etc_news).$$

Las respuestas mencionadas anteriormente cambian:

- La respuesta para $trust_report(ana, facebook_hits_one_billion)$ es *sí*.
- La respuesta para $trust_report(emanuel, facebook_hits_one_billion)$ es *no*.

Aunque la noción de tiempo no es modelada explícitamente, el sistema captura el dinamismo de la confiabilidad mediante la generación de conclusiones actualizadas que podrían ser diferentes de las obtenidas antes de agregar nuevos hechos.

La confiabilidad no es siempre simétrica

Muchas veces la confiabilidad no es simétrica. Un usuario puede confiar en otro usuario más de lo que este último confía en él [Vil05, DGS11, SKS⁺16, GZYS16, DRP17]. Es más, una relación de confiabilidad entre el segundo y el primer usuario podría no existir. Sin embargo, cuando ambas partes son confiables, convergerán en una alta confiabilidad mutua después de interacciones repetidas. Por el contrario, si uno de los usuarios no actúa de manera confiable, el otro usuario se verá obligado a penalizarlo, lo que dará lugar a una baja confiabilidad mutua. La ausencia de simetría puede considerarse un caso especial de personalización. La ausencia de simetría ocurre debido a las diferencias en las percepciones, opiniones, creencias y expectativas de las personas. La naturaleza no simétrica de la confiabilidad se ha identificado en varias jerarquías dentro de las organizaciones [YK00].

Como se puede observar en el ejemplo, el sistema puede entonces modelar una relación de confiabilidad no simétrica.

- La respuesta para $trust_viewer(emanuel, marcela)$ es *sí*.
- La respuesta para $trust_viewer(marcela, emanuel)$ es *indeciso*.

La confiabilidad no es siempre transitiva

La confiabilidad, como ya se menciono anteriormente, es propagativa. Esto no quiere decir, sin embargo, que la confiabilidad se propaga de forma ilimitada y por lo tanto, en general, no es transitiva [CH96, YS00, GKRT04, SNP13, WCFHV17]. Debido a su naturaleza propagativa, la información de confiabilidad se puede pasar de un usuario a otro. La propagación es la propiedad de la confiabilidad más estudiada. Varios modelos de confiabilidad han utilizado esta propiedad [SFR00, MMH02, Sab02, YSS04, JE10, AWBT18]. De manera similar, la literatura basada en la topología FOAF (del inglés *Friend Of A Friend*) se basa en la naturaleza propagativa de la confiabilidad [GPH03]. La transitividad implica propagación, pero lo contrario no es cierto. Desafortunadamente, la naturaleza propagativa de la confiabilidad a veces se confunde con la naturaleza transitiva de la confiabilidad en la literatura.

En el ejemplo se puede ver que a pesar de la existencia de hechos tales como:

$trust_viewer(ana, emanuel)$.

$trust_viewer(emanuel, marcela)$.

$trust_report(marcela, facebook_hits_one_billion)$.

El sistema concluye que:

- La respuesta para $trust_report(ana, facebook_hits_one_billion)$ es *no*.

La confiabilidad es dependiente del contexto

Por lo general la confiabilidad tiene cierto alcance en un contexto específico [THL13, SNP13, MWZW18]. Entonces, es importante que los modelos de confiabilidad permitan diferenciar el valor de confiabilidad para cada tópico o contexto diferente [TA12, PSM13, JTG19]. Por ejemplo, en la vida real una persona puede confiar en otra como albañil, pero por lo general no confía en esa misma persona como mecánico. Es decir si necesita construir una casa, esta persona es confiable en el contexto de la construcción, pero no

será confiable en el contexto de arreglar un automóvil. Esto es diferente del contexto de confiabilidad, que normalmente se refiere al entorno en el que existe la relación de confiabilidad, por ejemplo, aplicación de la ley, seguros y control social [TLU06, AHT10]. La naturaleza específica del contexto de la confiabilidad se discute en las ciencias sociales y psicológicas [RSBC98].

Haciendo referencia al ejemplo seleccionado, y recordando que un usuario puede elegir confiar o no confiar en ciertas fuentes o usuarios dependiendo del tema de la noticia en cuestión, supongamos que el ejemplo se amplía con los siguientes hechos:

```
report_source(vicodin_relieves_moderate_to_severe_pain, medical_news).
trust_viewer_topic(cristian, house, medicine).
is_about(vicodin_relieves_moderate_to_severe_pain, medicine).
trust_report(house, vicodin_relieves_moderate_to_severe_pain).
```

Los hechos anteriores indican que Cristian va a confiar en House cuando este exprese su confiabilidad con respecto a noticias sobre medicina. En este caso, ya que House confía en la noticia médica “vicodin_relieves_moderate_to_severe_pain”, la respuesta obtenida para *trust_report(cristian, vicodin_relieves_moderate_to_severe_pain)* va a ser *sí*.

Por otra parte, se supone que House también ha indicado que él no confía en la noticia “barça_is_the_best_football_team_of_all_time”. Esta noticia aparece en “sports_news”, una fuente de noticias confiable para Cristian. Esto está representado por los siguientes hechos:

```
~trust_report(house, barça_is_the_best_football_team_of_all_time).
report_source(barça_is_the_best_football_team_of_all_time, sports_news).
trust_source(cristian, sports_news).
```

En este caso, a pesar de que para House esta noticia no es confiable, la respuesta a *trust_report(cristian, barça_is_the_best_football_team_of_all_time)* será *sí*.

5.2.3. Propiedades de DeLP y su aptitud para modelar confiabilidad

El modelo de recomendación propuesto está basado en el marco argumentativo DeLP. Las lógicas argumentativas y las lógicas no monótonas en general no presentan las propiedades habituales de las lógicas clásicas, pero se prefieren a estas lógicas por su poder

expresivo y flexibilidad. Para caracterizar las lógicas no monótonas Dov Gabbay [Gab85] fue pionero en la comparación de las teorías no monótonas con respecto a un conjunto de propiedades deseables. El autor centra la atención en las propiedades de la relación de inferencia inducida por la teoría formal, es decir, la relación entre las conclusiones y el conjunto de premisas que las apoyan. Este enfoque también fue seguido primero por Kraus *et al.* [KLM90], y más tarde por Makinson [Mak94], quienes estudiaron el conjunto de propiedades básicas que toda teoría no monótona debe tener. La *inclusión*, la *idempotencia* y la *monotonidad cautelosa* (en inglés *cautious monotonicity*) fueron identificadas como el conjunto de propiedades básicas que las teorías no monótonas deben verificar. Los sistemas que verifican las propiedades de corte (en inglés *cut*) y monotonicidad cautelosa se dice que son *acumulativos*. Pero tratar de caracterizar a las teorías de la argumentación con estas propiedades no condujo a resultados alentadores [PV02, SC08]. Desafortunadamente, aún no hay consenso en la comunidad de argumentación sobre un determinado conjunto de propiedades que la inferencia argumentativa debe cumplir.

Para analizar formalmente el modelo propuesto se optó por considerar dos propiedades simples pero importantes. En primer lugar se consideró la propiedad de *inclusión*, que es claramente deseable en el contexto de cualquier relación de inferencia razonable, para aceptar esas premisas sobre las que se razonan. En lo que sigue se analiza la propiedad de inclusión en el contexto del sistema de recomendación de noticias descrito.

Proposición 5.1 *Considere un sistema de recomendación utilizando un programa DeLP $\mathcal{P} = (\Pi, \Delta)$ para derivar sus conclusiones. Sea \vdash el operador de inferencia de DeLP, que permite derivar una conclusión garantizada a partir de un programa DeLP. Sea $C(\Pi) = \{\phi \mid \Pi \vdash \phi\}$ una consecuencia que denota el conjunto de conclusiones garantizadas que se pueden obtener a partir del conjunto de hechos Π utilizando el programa de \mathcal{P} . Entonces $\Pi \subseteq C(\Pi)$.*

Demostración: Sea ϕ un hecho en Π . Así ϕ es apoyada por un argumento vacío. Los argumentos vacíos no pueden ser atacados ni derrotados por un argumento, y luego ϕ es una conclusión garantizada del programa \mathcal{P} , es decir, $\phi \in C(\Pi)$. \square

Esto puede parecer un resultado trivial, pero algunas teorías argumentativas que se han utilizado para implementar sistemas similares al presentado (por ejemplo, el sistema de ASPIC [CA07]) no satisfacen esta propiedad [SC08].

Otra propiedad que es razonable para cualquier mecanismo de inferencia es la *consistencia*. La consistencia es una propiedad básica para el sistema de recomendación de noticias, en el sentido de que no es razonable que para un determinado usuario v el sistema clasifique simultáneamente una dada noticia como confiable y no confiable. En el sistema recomendador propuesto esto significa que no es posible llegar a la conclusión $trust_report(v, r_i)$ y $\sim trust_report(v, r_i)$ para un usuario v y una noticia r_i . Con el fin de demostrar que este es el caso, vamos a demostrar que el conjunto de argumentos que pueden ser garantizados desde un programa DeLP dado no son contradictorios entre sí. Para mostrar esta propiedad primero se deben demostrar algunos resultados auxiliares.

Proposición 5.2 *Sea $\mathcal{P} = (\Pi, \Delta)$ un programa y $\langle \mathcal{A}, Q \rangle$, $\langle \mathcal{B}, R \rangle$ dos argumentos contruidos a partir de \mathcal{P} tal que $\langle \mathcal{A}, Q \rangle$ es un derrotador de $\langle \mathcal{B}, R \rangle$ y $\langle \mathcal{A}, Q \rangle$ es garantizado con respecto a \mathcal{P} . Sea $\langle \mathcal{A}_s, Q_s \rangle$ un argumento de soporte de $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$ tal que $\langle \mathcal{A}_s, Q_s \rangle$ es un *Nodo U*. Si $\langle \mathcal{A}_s, Q_s \rangle$ aparece en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$ este debe ser también etiquetado como un *Nodo U* en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$.*

Demostración: Para demostrar esta propiedad se usara inducción estructural sobre el subárbol con raíz en $\langle \mathcal{A}_s, Q_s \rangle$. Esta demostración es una adaptación de la prueba presentada en [Cap03].

Caso base Vamos a considerar el hecho de que $\langle \mathcal{A}_s, Q_s \rangle$ no tiene derrotadores en $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$.

Si \mathcal{A}_s pertenece a $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$, entonces debe ser marcado como *Nodo U*. Dado esto no hay nuevos derrotadores que se puedan construir.

Paso inductivo Supongamos que $\langle \mathcal{A}_s, Q_s \rangle$ tiene k derrotadores en $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$, marcados como $\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_k$, y $\langle \mathcal{A}_s, Q_s \rangle$ está presente en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$. Dado que \mathcal{A}_s es un *Nodo U* en $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$ cada derrotador de \mathcal{B}_i debe ser un *Nodo D* en $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$.

A continuación se demostrará que si \mathcal{B}_i está presente en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$, entonces debe ser etiquetado como un *Nodo D* en este árbol. Teniendo en cuenta que \mathcal{B}_i es un *Nodo D* en $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$ tiene que existir un argumento de soporte no derrotado \mathcal{C} en el subárbol con raíz en \mathcal{B}_i .

Si utilizamos la hipótesis inductiva en este subárbol se deduce que si \mathcal{C} está presente en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$ debe ser etiquetado como un *Nodo U* y luego \mathcal{B}_i debe ser un *Nodo D* en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$. Queda por comprobar lo que sucede cuando \mathcal{C} no puede ser presentado como un derrotador de \mathcal{B}_i en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$. Esto puede derivarse de una de estas razones:

1. \mathcal{C} es contradictorio con un argumento de interferencia en la línea de la argumentación en la que se debe agregar \mathcal{C} .
2. \mathcal{C} es un sub-argumento de un argumento anterior de esta línea de argumentación.

El primer escenario no es posible, dado que en este caso \mathcal{C} no podría aparecer en $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$, ya que sería contradictorio con un argumento de apoyo en la línea de argumentación. En el segundo caso, \mathcal{C} debe ser un sub-argumento de \mathcal{B} (de lo contrario no podría pertenecer a $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$). Sin embargo, ya que \mathcal{C} es un derrotador de \mathcal{B}_i , el conjunto $\mathcal{C} \cup \mathcal{B}_i \cup \Psi$ permite la derivación de literales complementarios. Entonces $\mathcal{B} \cup \mathcal{B}_i \cup \Psi$ también permite la derivación de literales complementarios ya que $\mathcal{C} \subseteq \mathcal{B}$. En esta situación, no es posible que \mathcal{B}_i esté en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$, dado que contradice un argumento de apoyo en su línea argumentativa. Por lo tanto, si \mathcal{B}_i está en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$ debe ser etiquetado como un **Nodo U** en este árbol.

□

El siguiente corolario se deriva de la proposición anterior.

Corolario 5.1 *Sea \mathcal{P} un programa y $\langle \mathcal{A}, Q \rangle, \langle \mathcal{B}, R \rangle$ dos argumentos contruidos a partir de \mathcal{P} . Si $\langle \mathcal{A}, Q \rangle$ es un derrotador de $\langle \mathcal{B}, R \rangle$ y $\langle \mathcal{A}, Q \rangle$ es un argumento garantizado en \mathcal{P} entonces $\langle \mathcal{A}, Q \rangle$ debe ser etiquetado como un **Nodo U** en el árbol de dialéctica con raíz en $\langle \mathcal{B}, R \rangle$.*

Demostración: Por hipótesis, el argumento $\langle \mathcal{A}, Q \rangle$ es un argumento de soporte no derrotado en $\mathcal{T}_{\langle \mathcal{A}, Q \rangle}$, que también está presente en $\mathcal{T}_{\langle \mathcal{B}, R \rangle}$. De la proposición 5.2 podemos inferir que $\langle \mathcal{A}, Q \rangle$ debe ser un **Nodo U** en el árbol de dialéctica para $\langle \mathcal{B}, R \rangle$. □

Finalmente, el siguiente lema prueba la consistencia en las recomendaciones.

Lema 5.1 *Sea un usuario v , una noticia r_i , y un programa $\mathcal{P} = (\Pi, \Delta)$ que contiene reglas y hechos DeLP. Entonces no es posible concluir para r_i $trust_report(v, r_i)$ y $\sim trust_report(v, r_i)$.*

Demostración: Si se supone por contradicción que existe v y r_i tal que $trust_report(v, r_i)$ y $\sim trust_report(v, r_i)$ se pueden derivar en el sistema, entonces

existe un argumento garantizado \mathcal{A} que soporta $trust_report(v, r_i)$ y un argumento garantizado \mathcal{B} que soporta $\sim trust_report(v, r_i)$. Sea $Warr(\mathcal{P})$ el conjunto de argumentos garantizados en \mathcal{P} . Ambos, \mathcal{A} y \mathcal{B} deberían estar en $Warr(\mathcal{P})$. En este caso \mathcal{A} ataca a \mathcal{B} . Se puede asumir sin pérdida de generalidad que \mathcal{A} derrota a \mathcal{B} . Ya que \mathcal{A} y \mathcal{B} están garantizados podemos asumir que existe un árbol de dialéctica con raíz en \mathcal{A} , marcado $\mathcal{T}_{\langle \mathcal{A}, trust_report(v, r_i) \rangle}$, que garantiza $trust_report(v, r_i)$ y un árbol de dialéctica $\mathcal{T}_{\langle \mathcal{B}, \sim trust_report(v, r_i) \rangle}$ que garantiza $\sim trust_report(v, r_i)$. El argumento $\langle \mathcal{A}, trust_report(v, r_i) \rangle$ derrota a $\langle \mathcal{B}, \sim trust_report(v, r_i) \rangle$ y así $\langle \mathcal{A}, trust_report(v, r_i) \rangle$ debe ser marcado como Nodo D en el árbol de dialéctica $\mathcal{T}_{\langle \mathcal{B}, \sim trust_report(v, r_i) \rangle}$, ya que de otra manera $\langle \mathcal{B}, \sim trust_report(v, r_i) \rangle$ podría no ser un argumento garantizado. Sin embargo, aplicando el Corolario 5.1 el argumento $\langle \mathcal{A}, trust_report(v, r_i) \rangle$ debe ser etiquetado como Nodo U en el árbol de dialéctica para $\langle \mathcal{B}, \sim trust_report(v, r_i) \rangle$ (ya que $\langle \mathcal{A}, trust_report(v, r_i) \rangle$ está garantizado y $\langle \mathcal{B}, \sim trust_report(v, r_i) \rangle$ derrota a $\langle \mathcal{A}, trust_report(v, r_i) \rangle$). Esta contradicción surge por asumir la existencia de un par de argumentos $\langle \mathcal{A}, trust_report(v, r_i) \rangle, \langle \mathcal{B}, \sim trust_report(v, r_i) \rangle$ bajo las condiciones previamente definidas. \square

En resumen, las teorías de argumentación siguen esperando la definición de un conjunto de propiedades que las caractericen y que puedan utilizarse como punto de referencia para estas teorías. Mientras tanto, se está de acuerdo en que la inclusión y la consistencia son dos propiedades básicas para cualquier teoría argumentativa, y deben ser consideradas como un punto de partida en cualquier análisis formal. En particular, se sostiene que son propiedades deseables para cualquier sistema utilizado para modelar la noción de confiabilidad.

5.3. Validación por simulación

Con el objetivo de validar el sistema de recomendación de noticias y con ésto el modelo de administración y propagación de confiabilidad propuesto, se han realizado una serie de simulaciones utilizando datos sintéticos. La Figura 5.4 describe la arquitectura del sistema que se ha implementado para llevar a cabo las simulaciones.

En cada una de las simulaciones realizadas, se ha generado una cantidad diferente de usuarios, noticias, fuentes de noticias y tópicos sintéticos para analizar cómo se comporta

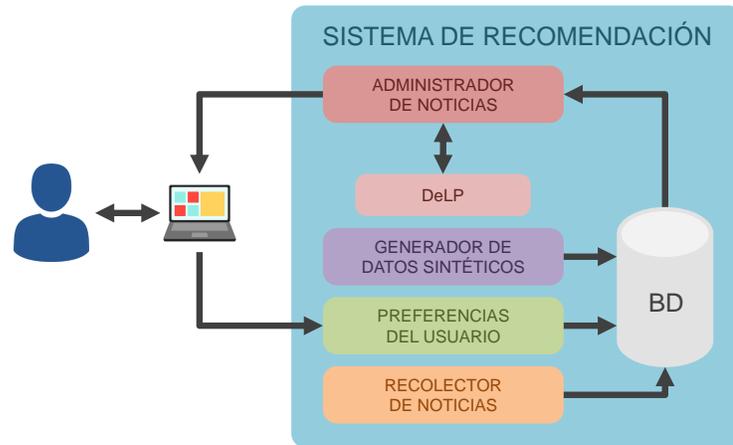


Figura 5.4: Arquitectura del sistema implementada para realizar las simulaciones.

el sistema. Otro parámetro necesario, para cada una de las simulaciones, fue determinar el grado de interacción entre los elementos del sistema.

Se utiliza el término “interacción” para hacer referencia a las aserciones de confianza o desconfianza expresadas por el usuario hacia las diferentes entidades del sistema, como noticias, fuentes de noticias y/o también otros usuarios. En otras palabras, son relaciones de confiabilidad explicitadas por el usuario. La Figura 5.5 resume los parámetros de configuración que fueron utilizados en cada una de las simulaciones junto a sus tiempos de ejecución¹.

Simulación	#Usuarios	#Noticias	#Fuentes	#Tópicos	Interacción	Tiempo
Sim 1	5	4	2	2	5%	00:00:04
Sim 2	5	4	2	2	25%	00:00:06
Sim 3	5	4	2	2	40%	00:03:57
Sim 4	10	9	3	3	5%	00:00:19
Sim 5	10	9	3	3	25%	00:50:23
Sim 6	15	12	4	4	5%	00:18:44
Sim 7	15	12	4	4	25%	01:32:16
Sim 8	20	50	5	5	5%	00:23:05
Sim 9	20	50	5	5	10%	01:04:08
Sim 10	20	50	5	5	50%	05:29:47

Figura 5.5: Configuración de las simulaciones y sus tiempos de ejecución.

¹La maquina utilizada para realizar las simulaciones disponía de un procesador AMD Phenom II con 6 núcleos y 8 GB de RAM.

Es importante destacar cómo estas simulaciones permiten validar el marco propuesto desde otro ángulo y mediante una metodología diferente. La Figura 5.6 presenta las relaciones de confiabilidad iniciales entre 20 usuarios y 50 noticias. Después de ejecutar la simulación 9 el sistema converge a las relaciones de confiabilidad que se muestran en la Figura 5.7. El resto de las matrices mostrando el antes y después de la ejecución de cada simulación pueden verse en el Apéndice A.

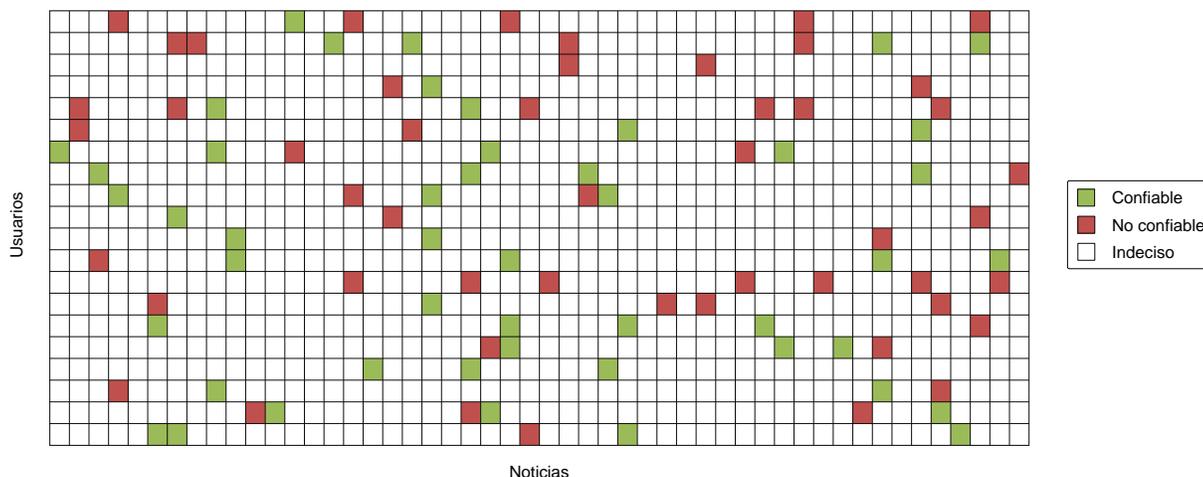


Figura 5.6: Relaciones de confiabilidad explícitas para la simulación 9.

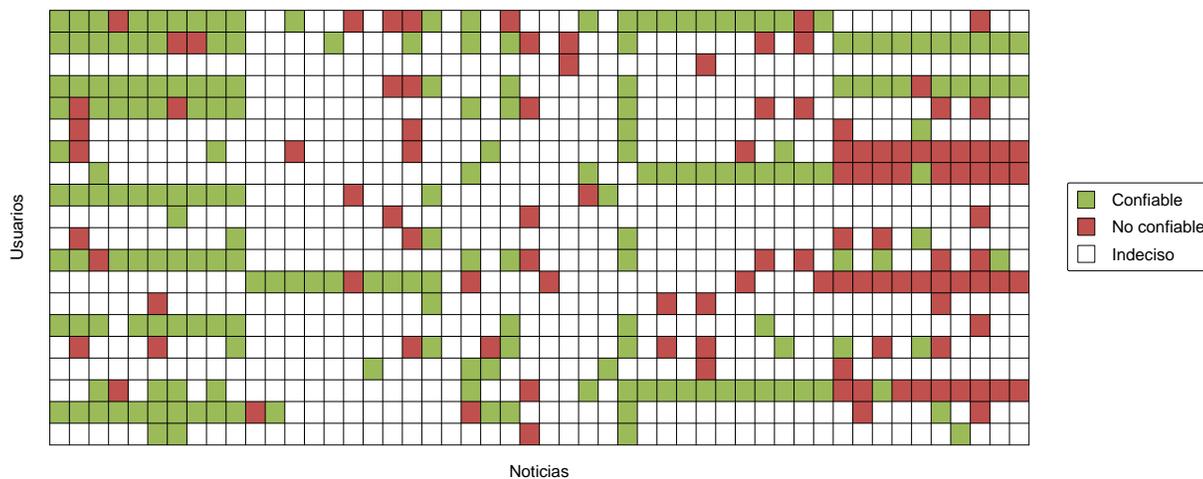


Figura 5.7: Relaciones de confiabilidad explícitas e inferidas para la simulación 9.

Estas cifras proporcionan una buena evidencia del poder de inferencia del marco. Al reducir significativamente el número de entradas de “indecisos” se puede ganar confianza

en las recomendaciones que se pueden presentar al usuario. Esto puede ser particularmente útil para contrarrestar el “arranque en frío”, uno de los problemas más comunes de los RS clásicos. Esto se debe a que, posiblemente, con un número menor de aserciones de confiabilidad provistas por el usuario (interacciones), el sistema tiene la capacidad de generar recomendaciones útiles. En la Figura 5.8 se muestra el porcentaje de relaciones de confiabilidad inferidas por el sistema durante las simulaciones.

Simulación	Entidades				Interacción	Relaciones inferidas	Indicis	
	#Usuarios	#Noticias	#Fuentes	#Tópicos			Antes	Después
Sim 1	5	4	2	2	5%	10%	95%	85%
Sim 2	5	4	2	2	25%	30%	75%	45%
Sim 3	5	4	2	2	40%	30%	60%	30%
Sim 4	10	9	3	3	5%	14%	95%	81%
Sim 5	10	9	3	3	25%	28%	75%	47%
Sim 6	15	12	4	4	5%	13%	95%	82%
Sim 7	15	12	4	4	25%	19%	75%	56%
Sim 8	20	50	5	5	5%	12%	95%	83%
Sim 9	20	50	5	5	10%	22%	90%	68%
Sim 10	20	50	5	5	50%	11%	50%	39%

Figura 5.8: Información inferida por el sistema.

Al analizar las Figuras 5.6 y 5.7 se pueden validar visualmente dos importantes propiedades discutidas anteriormente: *inclusión*, es decir, el estado de la credibilidad explicitada por el usuario con respecto a una noticia no puede ser revocado, y *consistencia*, es decir, el sistema no puede concluir que una entidad puede ser confiable y no confiable simultáneamente para el mismo usuario.

5.4. Conclusiones

En este capítulo se definió un modelo para administrar relaciones de confiabilidad en un RS. En la Sección 5.1 se presentó un marco teórico que puede aplicarse para implementar un sistema de recomendación basado en confiabilidad para las noticias en la web.

En la Sección 5.2 se determinó la aptitud de DeLP para modelar la noción de confiabilidad. Para esto, se realizó el análisis de un conjunto de propiedades deseables de la confiabilidad, como ser que es subjetiva, dinámica, rebatible, dependiente del contexto y

no siempre simétrica o transitiva. También se demostró que todas estas propiedades pueden modelarse de forma natural utilizando el marco propuesto. Se realizó la prueba formal de dos teoremas fundamentales que caracterizan el comportamiento de cualquier sistema de recomendación basado en el modelo propuesto de propagación de confiabilidad. El primer teorema establece que una premisa de confiabilidad explícita no puede ser revocada por el sistema. Más formalmente, establece que el conjunto de premisas de confiabilidad se incluyen en el conjunto de conclusiones de confiabilidad. El segundo teorema establece la consistencia del sistema, lo que excluye la posibilidad de concluir que un usuario confía y desconfía simultáneamente de la misma entidad.

Finalmente, en la Sección 5.3 se llevaron a cabo una serie de simulaciones. Estas simulaciones permitieron validar el modelo propuesto mostrando que cumple con las propiedades de *inclusión* y *consistencia*. A su vez, también dejan en evidencia el poder de inferencia del modelo.

Capítulo 6

Sistemas de recomendación mixtos basados en argumentación

Los sistemas de recomendación (RS), como ya se mencionó a largo del Capítulo 2, son mecanismos de soporte que ayudan a los usuarios en su proceso de toma de decisiones mientras interactúan con volúmenes de información grandes o complejos. La mayoría de los RS están destinados a ayudar a los usuarios a lidiar con el problema de la sobrecarga de información facilitando el acceso a los elementos relevantes [BOHG13] y se basan en técnicas de aprendizaje automático y en algoritmos de IR. Los sistemas resultantes suelen proporcionar sugerencias basadas en evidencias cuantitativas, es decir, medidas de similitud entre elementos o usuarios. El proceso de inferencia que condujo a estas sugerencias es comúnmente opaco para el usuario, es decir, como si se tratara de una “caja negra” que produce estas sugerencias.

En el Capítulo 2 fueron presentados los RS basados en contenido, de filtrado colaborativo y los híbridos. Aunque los RS híbridos son sustancialmente más eficaces que los enfoques básicos, como los basados en contenido y el filtrado colaborativo, los sistemas existentes son todavía limitados, ya que no pueden realizar inferencia cualitativa sobre las sugerencias que ofrecen. Una serie de formalismos bien conocidos, como las lógicas descriptivas (en inglés, *Description Logics*) [BCM⁺03], podrían integrarse naturalmente a las tecnologías de recomendación existentes para proporcionar capacidades de inferencia. Sin embargo, la mayoría de estos formalismos son incapaces de lidiar con la naturaleza rebatible de las preferencias del usuario en entornos complejos. Las decisiones acerca de las preferencias del usuario se basan principalmente en heurísticas, las cuales se apoyan en las

calificaciones anteriores del usuario o la recopilación de información de otros usuarios con intereses similares. Además, los enfoques cuantitativos adoptados por la mayoría de los sistemas de recomendación existentes no tienen un modelo subyacente que facilite su interpretabilidad. Esto hace que sea difícil proporcionar una explicación clara a los usuarios de los factores y procedimientos que llevaron al sistema a ofrecer ciertas recomendaciones. Otro problema que enfrentan los RS es que modelar los criterios de preferencia de los usuarios no es una tarea fácil, ya que generalmente implica un conocimiento incompleto y potencialmente inconsistente sobre el dominio de búsqueda. En este contexto, los marcos de argumentación rebatible constituyen una alternativa interesante, ya que han madurado por más de una década para convertirse en un escenario útil para formalizar el razonamiento cualitativo de sentido común de una manera computacionalmente atractiva.

La programación lógica rebatible, DeLP, como ya se mencionó en el Capítulo 3, es una formalización particular de argumentación rebatible basada en la programación lógica, que brinda la posibilidad de representar información en forma de reglas estrictas y rebatibles de una manera declarativa. Una característica importante del enfoque de DeLP es que, al realizar un razonamiento dialéctico rebatible, puede manejar información que en principio es contradictoria. El proceso de decidir si una conclusión es soportada, o está *garantizada*, comienza por analizar si existe un argumento aceptado que respalda esa conclusión, es decir, un argumento por el cual todo posible argumento de ataque ha sido derrotado. La noción de ataque y derrota ya fueron presentadas previamente en la Sección 3.2.1, pero pueden describirse intuitivamente como la consideración de argumentos en conflicto (*ataque*) con el argumento de soporte. El ataque se convierte en una derrota cuando el argumento de ataque es mejor, en un sentido específico, que el argumento de soporte. Por lo tanto, el uso de argumentación permitirá al sistema presentar sugerencias razonadas, que el usuario podrá conocer y aceptar más a fondo solo si la herramienta de recomendación puede presentar un caso convincente. Investigaciones recientes han demostrado que la argumentación se puede integrar en un número creciente de aplicaciones del mundo real, como planificación (en inglés, *planning*) [SGC04, PG18], sistemas multiagente [PSJ98, AMP02, PM03, LL07, CI15, GTGS18], razonamiento legal [PS02, Ver03b, dCPLM⁺17], ingeniería del conocimiento [CRL00], sistemas para la toma de decisiones y negociación [APM00, RRJ⁺03, BH09, AV12, FTG⁺17, BTG19], análisis de reportes de noticias [Hun01], agrupación en clústeres [GC04], búsqueda web inteligente [CM04a, CMS04], sistemas de soporte de argumentación [Ver03a], sistemas de mediación y argumentación colaborativa soportada por computadora [LNA⁺97, GK99,

RW01], el procesamiento de lenguaje natural [CM04b, WvEH16], entre otros.

Partiendo del trabajo realizado en [BBD⁺12, BBD⁺14], en este capítulo se presenta un acercamiento hacia la integración de variantes típicas de los sistemas de recomendación, con un marco de argumentación rebatible. Para mejorar el soporte al usuario por medio de argumentación, los criterios de preferencia del usuario se modelan mediante hechos, reglas estrictas y rebatibles codificadas como parte de un programa DeLP. Estos criterios de preferencia se combinan con información de fondo adicional, referente a algún dominio de recomendación en particular, y son utilizados por el marco argumentativo para priorizar posibles sugerencias, mejorando así los resultados finales proporcionados al usuario activo.

El resto del capítulo está estructurado de la siguiente manera. Primero, se plantearán modelos, de una manera intuitiva, para implementar sistemas de recomendación basados en argumentación, para los dominios de películas y música. Luego se mostrará el rol que cumple la argumentación en un sistema de recomendación, haciendo hincapié en que la argumentación permite ofrecer explicaciones como conjuntos coherentes de razones a favor o en contra de una recomendación. Finalmente se evaluará empíricamente uno de los sistemas planteados, particularmente el RS de películas, mostrando los resultados obtenidos de comparar distintas versiones del mismo recomendador, como así también de compararlo con otros RS del estado del arte.

6.1. RS basados en argumentación

Como se ha señalado anteriormente [CMG09], los enfoques de recomendación basados en criterios puramente cuantitativos, a pesar de ser muy eficaces, a menudo fallan en dar a los usuarios las razones detrás de las recomendaciones, lo que afecta la confianza de los usuarios en los resultados. El grado de confianza que un usuario coloca en un recomendador es crucial para la aceptación de las recomendaciones [BTC17]. Los usuarios desean conocer el motivo de las recomendaciones, y podrían no aceptar predicciones eficaces si no hay confianza en ellas [BGOZ18]. En los enfoques mixtos, como el que se propone en este capítulo, las explicaciones se complementan naturalmente con las recomendaciones. Por lo tanto, el usuario recibirá tanto una recomendación como una razón que la justifica. Esto tiene una doble ventaja: el usuario tendrá mayor confiabilidad en el resultado presentado, y el usuario podrá brindar retroalimentación explícita al sistema, la que podrá guiar la generación de futuras recomendaciones.

El mecanismo para la generación de recomendaciones en un sistema de recomendación basado en argumentación es muy diferente del proceso adoptado por la mayoría de los recomendadores existentes. Sin embargo, comparten el mismo espíritu: intentan establecer la similitud entre los objetos o usuarios, y luego usan esta similitud para generar recomendaciones. La principal diferencia es que mientras que los enfoques puramente cuantitativos calculan esta similitud como una medida numérica, en el enfoque basado en argumentación propuesto, la similitud se determina mediante reglas que establecen qué características tienen que ser compartidas entre los objetos o usuarios para que se consideren similares.

A continuación se muestra cómo los dos enfoques, cuantitativo y cualitativo, pueden modelarse utilizando reglas DeLP para implementar un sistema de recomendación de carácter mixto. Este enfoque mixto de recomendación permite generar recomendaciones razonadas sobre grandes conjuntos de datos del mundo real.

Se han planteado sistemas de recomendación para dos dominios diferentes, como lo son la música y las películas. En la Figura 6.1 se puede apreciar la arquitectura básica general de los sistemas de recomendación definidos.

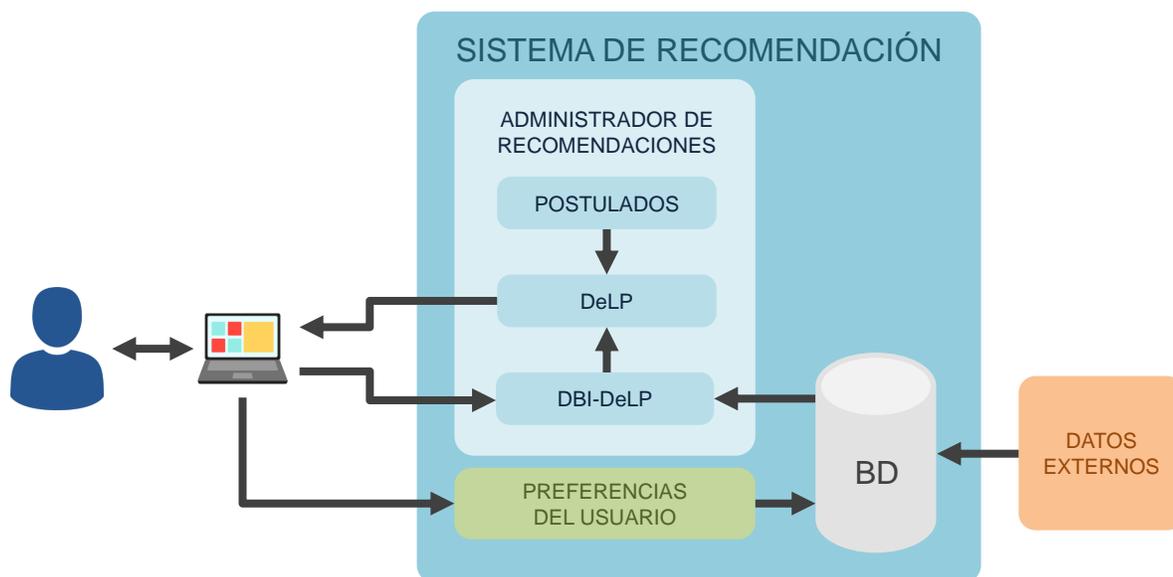


Figura 6.1: Arquitectura de los RS basados en DeLP.

En general, los RS propuestos con esta arquitectura tendrán el siguiente comportamiento para generar sugerencias. El usuario mediante la interacción con el sistema incorporará sus preferencias explícitamente (por ejemplo mediante la calificación de algún

elemento), o implícitamente (por ejemplo, al ser monitoreado por el sistema) y se almacenaran en la base de datos del RS. El usuario interactúa con el sistema generando consultas para poder recibir sugerencias de elementos. El tipo de elemento que se recomiende dependerá del dominio del RS. Cuando se realiza alguna consulta al RS, para determinar si recomendar un elemento o no al usuario, será recibida por el administrador de recomendaciones. Particularmente será DBI-DeLP el sistema a cargo de presentar dicha consulta a la base de datos y recuperar información relevante, junto con las preferencias del usuario. Cabe mencionar que la información almacenada en la base de datos relacionada con el dominio del RS puede ser actualizada o aumentada a partir de datos externos. Una vez hecho esto, todo pasa a manos de DeLP. Es decir, una vez que se haya obtenido toda la información relevante y las preferencias del usuario de la base de datos, todo se traduce a hechos DeLP. Finalmente esto es combinado con los postulados definidos para el RS en cuestión, los cuales son representados mediante reglas DeLP. Entonces con toda esta información (los postulados, las preferencias del usuario, la información relacionada con la consulta y la consulta propiamente dicha) DeLP genera una respuesta para la consulta. En base a esa respuesta se recomendará o no el elemento al usuario.

6.1.1. RS de películas basado en argumentación

El enfoque de recomendación propuesto se basa en un conjunto de postulados intuitivos que se han desarrollado para describir las condiciones en que una película debe ser recomendada para un usuario determinado. El sistema trata con cuatro entidades diferentes: películas, géneros, actores y usuarios.

- **Película.** Una película, o filme, es una obra cinematográfica que cuenta una historia, que puede ser tomada de un libro, una obra de teatro, otra película, hechos reales, un videojuego, un cómic, una novela gráfica o una idea original. Una película puede ser representada por un diverso conjunto de atributos, entre los más descriptivos o representativos, que pueden encontrarse en los distintos conjuntos de datos, se pueden mencionar los géneros en los cuales fue clasificada, los actores, directores y escritores que participaron en ella, así como también su título, duración, año de estreno, etc.
- **Género.** Un género cinematográfico es el tema general de una película que sirve para su clasificación. Los géneros cinematográficos se clasifican según los elementos

comunes de las películas, originalmente según sus aspectos formales como el ritmo, el estilo y sobre todo, el sentimiento que busquen provocar en el espectador. Alternativamente, los géneros cinematográficos se definen por su ambientación o por su formato.

- **Actor.** Un actor, o actriz, es aquella persona que representa a alguno de los distintos personajes involucrados en la historia de una película.
- **Usuario.** Un usuario es cualquier persona que utilice el RS. El sistema mantiene un listado de todas las calificaciones de películas realizadas por el usuario.

Soportando recomendaciones: aspectos cuantitativos y cualitativos

Se toma el siguiente conjunto de postulados definidos de una manera intuitiva, para determinar en base a qué aspectos, se realizarán las recomendaciones de películas para los usuarios del sistema.

■ Postulado 1

- $P_{1,1}$: A un usuario normalmente le gustará una película si el promedio de calificaciones de la película se encuentra por encima del promedio general de calificaciones de todas las películas.
- $P_{1,2}$: A un usuario normalmente no le gustará una película si el promedio de calificaciones de la película se encuentra por debajo del promedio general de calificaciones de todas las películas.

■ Postulado 2

- $P_{2,1}$: A un usuario normalmente le gustará una película si el género de la película es uno de los géneros favoritos del usuario.
- $P_{2,2}$: A un usuario normalmente no le gustará una película si el género de la película es uno de los géneros que más le disgusta al usuario.

■ Postulado 3

- $P_{3,1}$: A un usuario normalmente le gustará una película si un actor de la película es uno de los actores favoritos del usuario.

- $P_{3.2}$: A un usuario normalmente no le gustará una película si un actor de la película es uno de los actores que más le disgusta al usuario.

■ Postulado 4

- $P_{4.1}$: A un usuario normalmente le gustará una película si le gustó otra película que tiene al menos un actor en común y comparten el mismo género.
- $P_{4.2}$: A un usuario normalmente no le gustará una película si no le gustó otra película que tiene al menos un actor en común y comparten el mismo género.

El *Postulado 1*, representa el hecho de que una película, por lo general será una buena película, si el promedio de las calificaciones que ha recibido por parte de los distintos usuarios, está por encima de la calificación promedio de todas las películas presentes en el sistema. Este postulado se encuentra basado en aspectos cuantitativos, y permite realizar recomendaciones no personalizadas, es decir, no importa el usuario para el cual se quiera generar la recomendación. Este tipo de recomendaciones son particularmente útiles para casos de usuarios con pocas o ninguna calificación en el sistema. De esta manera, el sistema ofrece una recomendación que para un usuario general podría ser de utilidad, en lugar de ofrecer una recomendación al azar.

Los *Postulados 2, 3 y 4*, permitirán realizar recomendaciones basándose en información que típicamente se podría tener de una película, como lo son el género y los actores que han participado en ella. A diferencia del *Postulado 1*, estos postulados corresponden a un enfoque basado en aspectos cualitativos, haciendo referencia a información que representa al elemento a recomendar. Particularmente al *Postulado 4* se lo puede considerar como una combinación de los *Postulados 2 y 3*, dando lugar a recomendaciones más específicas y refinadas.

Con la definición de estos postulados básicos e intuitivos, se logra un modelo para realizar un sistema de recomendación de películas. En este caso, como el RS resultante se basará en los postulados definidos, los cuales combinan aspectos cuantitativos y cualitativos, se obtendrá un RS de carácter mixto.

Implementación en DeLP

Los postulados definidos en la sección anterior, se pueden traducir fácilmente en reglas DeLP. La aplicabilidad de las reglas definidas dependerá de la información en el conjunto

de datos seleccionado. A continuación se puede ver la traducción de los postulados a reglas DeLP.

$$\begin{aligned}
good_movie(Movie) &\multimap avg_rating(Movie, Rating), & (P_{1.1}) \\
&Rating > AVG_Rating. \\
\sim good_movie(Movie) &\multimap avg_rating(Movie, Rating), & (P_{1.2}) \\
&Rating \leq AVG_Rating. \\
likes_by_genre(Movie, User) &\multimap top_genre(User, Genre), & (P_{2.1}) \\
&genre(Movie, Genre). \\
\sim likes_by_genre(Movie, User) &\multimap bottom_genre(User, Genre), & (P_{2.2}) \\
&genre(Movie, Genre). \\
likes_by_actor(Movie, User) &\multimap top_actor(User, Actor), & (P_{3.1}) \\
&leads_in(Movie, Actor). \\
\sim likes_by_actor(Movie, User) &\multimap bottom_actor(User, Actor), & (P_{3.2}) \\
&leads_in(Movie, Actor). \\
likes_by_actor_genre(Movie, User) &\multimap top_actor(User, Actor), & (P_{4.1}) \\
&leads_in(Movie, Actor), \\
&top_genre(User, Genre), \\
&genre(Movie, Genre). \\
\sim likes_by_actor_genre(Movie, User) &\multimap bottom_actor(User, Actor), & (P_{4.2}) \\
&leads_in(Movie, Actor), \\
&bottom_genre(User, Genre), \\
&genre(Movie, Genre).
\end{aligned}$$

Hay que tener en cuenta que *AVG.Rating* representa el valor promedio de todas las calificaciones que recibieron las películas del conjunto de datos, es decir, este valor deberá ser calculado una vez que se haya optado por un conjunto de datos en particular. Además, en busca de la simplicidad, se considerará que ciertos literales ya se encuentran incorporados. Estos literales son calculados recuperando información del conjunto de datos seleccionado. Este proceso es realizado por medio del formalismo DBI-DeLP, presentado en la Sección 3.3, permitiendo usar los literales previamente incorporados, para probar los literales bajo análisis. Por ejemplo, el sistema utiliza el literal *genre(Movie, Genre)* para recuperar información de la base de datos sobre los géneros de una determinada película. Además, los postulados utilizan diferentes predicados auxiliares para averiguar los actores y géneros favoritos/menos preferidos para un determinado usuario. A continuación se

muestra cómo estos predicados son calculados para el análisis que se presentará en este capítulo, aunque es importante notar que existen diversas maneras de calcularlos.

- *avg_rating*: se calcula como el promedio de todas las calificaciones que recibió una determinada película.
- *top_actor*: se definieron dos formas de determinar los actores favoritos. La primera de ellas es la de obtener los tres actores que aparecen con mayor frecuencia en el conjunto de las películas vistas por el usuario. Una forma alternativa es considerar solo aquellas películas que el usuario ha calificado con un valor de cuatro o más.
- *bottom_actor*: se obtiene como los tres actores que aparecen con mayor frecuencia en el conjunto de películas que el usuario ha calificado con un valor de dos o menos.
- *top_genre*: se define como los tres géneros que aparecen con mayor frecuencia en el conjunto de películas que el usuario ha calificado con un valor de cuatro o más.
- *bottom_genre*: se obtiene como los tres géneros que aparecen con mayor frecuencia en el conjunto de películas que el usuario ha calificado con un valor de dos o menos.

Además, para utilizar los postulados es necesario codificar un grupo auxiliar de reglas de la forma:

$$\begin{aligned} \textit{recommend}(\textit{Movie}, \textit{User}) &\rightarrow \textit{likes_by_postulate}(\textit{Movie}, \textit{User}). \\ \sim\textit{recommend}(\textit{Movie}, \textit{User}) &\rightarrow \sim\textit{likes_by_postulate}(\textit{Movie}, \textit{User}). \end{aligned}$$

Por lo tanto, una instanciación particular de las reglas podrá ser, por ejemplo:

$$\begin{aligned} \textit{recommend}(\textit{Movie}, \textit{User}) &\rightarrow \textit{likes_by_actor}(\textit{Movie}, \textit{User}). \\ \sim\textit{recommend}(\textit{Movie}, \textit{User}) &\rightarrow \sim\textit{likes_by_actor}(\textit{Movie}, \textit{User}). \end{aligned}$$

Como se mostrará posteriormente, la definición del sistema se basa en el conjunto de postulados descritos anteriormente (*Postulados 1 al 4*), que como puede verse corresponde a un enfoque basado en contenido. Sin embargo, esta es una característica particular de esta versión del recomendador, y no representa una restricción del marco subyacente. Es posible ampliar fácilmente el recomendador incorporando postulados adicionales que representan aspectos de filtrado colaborativo. Por ejemplo, podría incorporarse un conjunto de postulados adicionales para extender los postulados básicos, haciendo uso de la relación entre usuarios.

■ Postulado 5

- $P_{5.1}$: A un usuario normalmente le gustará una película si esta película le gusta a otro usuario similar.
- $P_{5.2}$: A un usuario normalmente no le gustará una película si esta película no le gusta a otro usuario similar.

■ Postulado 6

- $P_{6.1}$: A un usuario normalmente le gustará una película si esta película le gusta a un grupo de usuarios similares.
- $P_{6.2}$: A un usuario normalmente no le gustará una película si esta película no le gusta a un grupo de usuarios similares.

Este conjunto adicional de postulados describe el uso de aspectos colaborativos para realizar una recomendación, explotando la relación entre usuarios similares (vecinos). Combinando este conjunto extra de postulados con el conjunto presentado anteriormente, se termina de definir un enfoque de recomendación híbrido, es decir, las recomendaciones estarían basadas tanto en el contenido de los elementos a recomendar, como en aspectos colaborativos.

6.1.2. RS de música basado en argumentación

El RS de música propuesto se ha desarrollado en base a la definición de algunos postulados intuitivos que establecen las condiciones que deben cumplirse para que una determinada canción sea recomendada a un usuario en particular. El sistema administra cinco entidades diferentes: canciones, artistas, géneros musicales, etiquetas y usuarios.

- **Canción.** Una canción es una composición musical para la voz humana, con letra y comúnmente acompañada por instrumentos musicales. Una canción puede ser representada mediante un gran conjunto de atributos, entre los más descriptivos o representativos que pueden encontrarse en los distintos conjuntos de datos, se pueden mencionar los géneros en los cuales fue clasificada, los artistas que la interpretan o la compusieron, el álbum del cual forma parte, así como también su título, duración, fecha de publicación, etc.

- **Artista.** Un artista es aquella persona que interpreta una canción. También puede tratarse de un grupo de personas si se trata de una banda o conjunto musical.
- **Género.** Un género musical es una categoría que recibe la composición musical de acuerdo a diferentes criterios, tales como su función (por ejemplo música de danza, religiosa, de cine, etc.), su instrumentación (por ejemplo música vocal, instrumental, electrónica, etc.), el contexto social en que es producida o el contenido de su texto.
- **Etiqueta.** Una etiqueta es una palabra o frase que se asocia a alguna de las otras entidades del sistema (canciones, artistas, usuarios). Las etiquetas cumplen la función de ayudar a poder describir a las entidades para así poder compararlas y asociarlas entre ellas.
- **Usuario.** Un usuario es cualquier persona que utilice el RS. El sistema mantiene un listado de todas las calificaciones de canciones realizadas por el usuario además de las etiquetas que haya definido y/o asociado a otras entidades.

Soportando recomendaciones: aspectos cuantitativos y cualitativos

A continuación se define el siguiente conjunto de postulados de una manera intuitiva, para determinar en base a qué aspectos, se realizarán las recomendaciones de canciones a los usuarios del sistema.

▪ Postulado 1

- $M_{1,1}$: A un usuario normalmente le gustará una canción si la canción es escuchada a menudo por el resto de usuarios del sistema.
- $M_{1,2}$: A un usuario normalmente no le gustará una canción si la canción no es frecuentemente escuchada por el resto de usuarios del sistema.

▪ Postulado 2

- $M_{2,1}$: A un usuario normalmente le gustará una canción si le gusta otra canción del mismo artista.
- $M_{2,2}$: A un usuario normalmente no le gustará una canción si no le gusta otra canción del mismo artista.

■ Postulado 3

- $M_{3.1}$: A un usuario normalmente le gustará una canción si le gusta otra canción interpretada por un artista relacionado con esta canción.
- $M_{3.2}$: A un usuario normalmente no le gustará una canción si no le gusta otra canción interpretada por un artista relacionado con esta canción.

■ Postulado 4

- $M_{4.1}$: A un usuario normalmente le gustará una canción si le gusta otra canción del mismo género musical.
- $M_{4.2}$: A un usuario normalmente no le gustará una canción si no le gusta otra canción del mismo género musical.

■ Postulado 5

- $M_{5.1}$: A un usuario normalmente le gustará una canción si la canción le gusta a un usuario similar.
- $M_{5.2}$: A un usuario normalmente no le gustará una canción si la canción no le gusta a un usuario similar.

■ Postulado 6

- $M_{6.1}$: A un usuario normalmente le gustará una canción si la canción le gusta a un grupo de usuarios similares.
- $M_{6.2}$: A un usuario normalmente no le gustará una canción si la canción no le gusta a un grupo de usuarios similares.

El *Postulado 1*, representa el hecho de que una canción, por lo general será una buena canción para recomendar, si la cantidad de reproducciones que tuvo está por encima de la cantidad de reproducciones promedio de todas las canciones del conjunto de datos, es decir, la canción fue más escuchada que el promedio. Este postulado se encuentra basado en aspectos puramente cuantitativos, y permite ofrecer recomendaciones no personalizadas, por lo que, no importa el usuario para el cual se quiera generar la recomendación. Este tipo de recomendaciones resultan particularmente útiles para los casos en los que el usuario

ha tenido poca interacción, o ninguna, con el sistema. De esta manera, el sistema ofrece una recomendación que podría ser de utilidad para un usuario general, en lugar de ofrecer una recomendación de modo aleatorio.

Los *Postulados 2, 3 y 4*, permitirán realizar recomendaciones basándose en información que típicamente se podría tener de una canción, como lo son el género y los artistas que la compusieron y/o interpretan. A diferencia del *Postulado 1*, estos postulados corresponden a un enfoque basado en aspectos cualitativos, haciendo referencia a información que representa al elemento a recomendar.

Los *Postulados 5 y 6*, hacen uso de aspectos colaborativos para realizar una recomendación, explotando la relación entre usuarios similares (vecinos). Para el caso del *Postulado 5* teniendo en cuenta solo a un usuario, y en el caso del *Postulado 6* a un grupo de usuarios.

Finalmente, combinando los seis postulados, se logra un modelo para realizar un sistema de recomendación de música. El RS resultante tendrá un enfoque híbrido, ya que para realizar una recomendación se estará basando tanto en aspectos del filtrado colaborativo como también en el contenido de los elementos a recomendar.

Implementación en DeLP

Los postulados definidos en la sección anterior, se pueden traducir fácilmente en reglas DeLP. La aplicabilidad de las reglas dependerá de la información en el conjunto de datos elegido. A continuación se puede ver la traducción de los postulados a reglas DeLP.

$$\text{likes_by_listen_ratio}(Track, User) \leftarrow \text{listen_ratio}(Track, Ratio), \quad (M_{1.1})$$

$$Ratio > AVG_Ratio.$$

$$\sim \text{likes_by_listen_ratio}(Track, User) \leftarrow \text{listen_ratio}(Track, Ratio), \quad (M_{1.2})$$

$$Ratio \leq AVG_Ratio.$$

$$\text{likes_by_artist}(Track, User) \leftarrow \text{has_listened}(User, Track_1, Artist, l), \quad (M_{2.1})$$

$$\text{artist_track}(Artist, Track),$$

$$\text{artist_track}(Artist, Track_1).$$

$$\sim \text{likes_by_artist}(Track, User) \leftarrow \text{has_listened}(User, Track_1, Artist, d), \quad (M_{2.2})$$

$$\text{artist_track}(Artist, Track),$$

$$\text{artist_track}(Artist, Track_1).$$

$$\text{likes_by_related_artist}(Track, User) \multimap \text{has_listened}(User, Track_1, Artist, l), \quad (M_{3.1})$$

$$\text{same_tags}(Artist, Track).$$

$$\sim\text{likes_by_related_artist}(Track, User) \multimap \text{has_listened}(User, Track_1, Artist, d), \quad (M_{3.2})$$

$$\text{same_tags}(Artist, Track).$$

$$\text{likes_by_musical_genre}(Track, User) \multimap \text{has_listened}(User, Track_1, -, l), \quad (M_{4.1})$$

$$\text{same_genre}(Track, Track_1).$$

$$\sim\text{likes_by_musical_genre}(Track, User) \multimap \text{has_listened}(User, Track_1, -, d), \quad (M_{4.2})$$

$$\text{same_genre}(Track, Track_1).$$

$$\text{likes_by_similar_user}(Track, User) \multimap \text{similar_to}(User, User_1), \quad (M_{5.1})$$

$$\text{has_listened}(User_1, Track, -, l).$$

$$\sim\text{likes_by_similar_user}(Track, User) \multimap \text{similar_to}(User, User_1), \quad (M_{5.2})$$

$$\text{has_listened}(User_1, Track, -, d).$$

$$\text{likes_by_other_users}(Track, User) \multimap \text{belongs_to}(\text{Cluster}, User), \quad (M_{6.1})$$

$$\text{likes_track}(\text{Cluster}, Track).$$

$$\sim\text{likes_by_other_users}(Track, User) \multimap \text{belongs_to}(\text{Cluster}, User), \quad (M_{6.2})$$

$$\sim\text{likes_track}(\text{Cluster}, Track).$$

Hay que tener en cuenta que *AVG_Ratio* representa el valor promedio de veces que fueron escuchadas las canciones del conjunto de datos. También es necesario aclarar que, para representar las calificaciones del usuario se optó por una calificación binaria, es decir, un usuario solamente podrá decir si una canción le gustó o no le gustó. Es por esto que como último parámetro del predicado *has_listened* se puede observar una “*l*” para hacer referencia a que al usuario le gustó la canción (en inglés, *liked*) o una “*d*” para hacer referencia a que al usuario no le gustó la canción (en inglés, *disliked*). Esta forma de calificación fue elegida ya que, por ejemplo, es la forma en la que *Last.fm*¹ le permite a sus usuarios calificar las canciones.

En busca de la simplicidad, se considerará que ciertos literales ya se encuentran incorporados. Estos literales son calculados recuperando información del conjunto de datos seleccionado. Este proceso es realizado por medio del formalismo DBI-DeLP, presentado en la Sección 3.3, permitiendo usar los literales previamente incorporados para probar los literales bajo análisis. Por ejemplo, el sistema utiliza el literal *artist_track(Artist, Track)* para recuperar información de la base de datos sobre el artista de una determinada canción. Además, los postulados utilizan diferentes predicados auxiliares para averiguar si las

¹<https://www.last.fm/>

canciones, artistas, usuarios y grupos de usuarios (en inglés *clusters*) están relacionados. A continuación se muestra cuál es la función de éstos predicados.

- *listen_ratio*: se calcula como el cociente entre la cantidad de reproducciones que tiene una determinada canción y la cantidad total de reproducciones de todas las canciones del sistema.
- *same_tags*: este predicado tiene la función de determinar si una canción está relacionada con un artista. Para esto utiliza las etiquetas que tienen asociadas las canciones y los artistas. Si la canción y el artista comparten un número significativo de etiquetas se puede decir que están relacionados.
- *same_genre*: la función de este predicado es determinar si una canción comparte el mismo género con otra canción. Se espera que cada canción tenga asociado al menos un género musical para poder realizar la comparación, pero en caso de no tenerlo podría determinarse que comparten el mismo género si ambas canciones tienen asociadas un número significativo de etiquetas iguales.
- *similar_to*: la similitud entre los usuarios está basada en las canciones escuchadas y las etiquetas utilizadas.
- *belongs_to*: un usuario pertenecerá a un grupo de usuarios (cluster) en función de las canciones escuchadas y las etiquetas utilizadas.
- *likes_track*: se puede decir que a un grupo de usuarios (cluster) le gusta una canción si a la mayoría de los usuarios del grupo le gustó la canción.

Además, para utilizar los postulados se necesita codificar un grupo auxiliar de reglas en la forma:

$$\begin{aligned} \text{recommend}(\text{Track}, \text{User}) &\leftarrow \text{likes_by_postulate}(\text{Track}, \text{User}). \\ \sim\text{recommend}(\text{Track}, \text{User}) &\leftarrow \sim\text{likes_by_postulate}(\text{Track}, \text{User}). \end{aligned}$$

Por lo tanto, una instanciación particular de la regla podrá ser, por ejemplo:

$$\begin{aligned} \text{recommend}(\text{Track}, \text{User}) &\leftarrow \text{likes_by_artist}(\text{Track}, \text{User}). \\ \sim\text{recommend}(\text{Track}, \text{User}) &\leftarrow \sim\text{likes_by_artist}(\text{Track}, \text{User}). \end{aligned}$$

6.1.3. Decidiendo qué es importante: aspectos relevantes en la recomendación final

Sin importar cuál sea el dominio de un RS basado en argumentación, la definición de modelos similares a los anteriormente mencionados (RS de películas o música), se basará en un conjunto dado de postulados que posibilitará obtener argumentos a favor o en contra de las potenciales recomendaciones. Una cuestión derivada de este enfoque es la forma de decidir si finalmente recomendar un elemento (en los casos anteriores películas o canciones) o no. Por este motivo se ha introducido un criterio de preferencia de argumentos que combina dos criterios diferentes que le ayudarán al RS a resolver esta situación.

- **Prioridad entre reglas.** Se utiliza un orden de preferencia predefinido entre reglas para determinar qué argumento prevalece. Para hacer esto, se debe definir una relación de orden parcial entre las reglas. Luego, al comparar dos argumentos, se consideran las dos reglas que tienen conclusiones conflictivas, y el argumento ganador es el que contiene la regla de mayor prioridad.
- **Especificidad generalizada.** Se prefieren aquellos argumentos que se basan en más información o aquellos que apoyan sus conclusiones más directamente [SGCS03, GS04].

Dados estos criterios, es necesario describir cómo se combinan para establecer qué argumentos prevalecen en una situación de ataque. Para los RS planteados se otorgó mayor prioridad al criterio de *Prioridad entre reglas* sobre el de *Especificidad generalizada*. Posteriormente, se mostrará cómo este orden entre los dos criterios permitirá establecer una relación de preferencia entre los diferentes aspectos involucrados en una recomendación.

Se utilizará el criterio de prioridad entre reglas para determinar cuál prevalece sobre las demás. Esto da la oportunidad de encontrar fácilmente la mejor combinación de los postulados por pruebas empíricas.

El segundo criterio de preferencia seleccionado es la especificidad generalizada, se utiliza solo por simplicidad, ya que va a resolver los conflictos en aquellos casos en los que no se quiera establecer prioridades explícitas. En particular, esto es útil cuando se utiliza una regla para refinar las condiciones establecidas en otra. Por ejemplo, de acuerdo con este criterio, teniendo en cuenta las reglas:

$$\begin{aligned}
\textit{recommend}(\textit{Item}, \textit{User}) &\multimap \textit{likes_by_postulate}(\textit{Item}, \textit{User}). \\
\sim\textit{recommend}(\textit{Item}, \textit{User}) &\multimap \textit{likes_by_postulate}(\textit{Item}, \textit{User}), \\
&\quad \textit{is_counter_example}(\textit{Item}, \textit{User}).
\end{aligned}$$

los argumentos construidos con la segunda regla serán más específicos que los construidos con la primera, por lo tanto serán los que prevalecerán.

6.1.4. Expandiendo aspectos: incorporando reglas

Ya se ha mostrado cómo los diferentes aspectos que se pueden tomar en cuenta para las recomendaciones pueden ser incluidos en un sistema de recomendación basado en argumentación modelándolo a través de combinaciones de reglas estrictas y rebatibles. Por otra parte, como se ha mencionado anteriormente, utilizando un enfoque argumentativo en la recomendación como el presentado anteriormente, se puede combinar el aspecto modelado en prácticamente cualquier forma que se quiera. Esto permite construir RS mixtos, y permite modificar fácilmente el enfoque para estar más orientado hacia lo cuantitativo o hacia lo cualitativo para ajustarse mejor a las limitaciones del contexto particular en que se va a utilizar. Tales características pueden ser sin duda muy útiles en el desarrollo de RS para varios dominios particulares, haciendo que este enfoque basado en argumentación sea una opción viable para tales desarrollos.

Otra ventaja importante del uso de reglas para codificar los criterios de recomendación es que las reglas representan los diferentes casos y éstos se modelan de una manera coloquial, lo que es fácil de entender y discutir por los usuarios finales. Esta característica facilita la interacción entre el usuario y el sistema. De esta manera, este enfoque facilita el proceso de retroalimentación que el usuario puede brindar al sistema, mediante una interacción más cercana al lenguaje natural basada en el uso de reglas. Esto se contrasta con métodos tradicionales que resultan menos intuitivos para el usuario, donde el mecanismo de retroalimentación se basa en el ajuste de parámetros que afectan fórmulas matemáticas difíciles de entender por el usuario.

Una característica útil de ciertos enfoques cuantitativos es que los mismos pueden ser modificados con el fin de perfeccionar las recomendaciones anteriores. Esto es algo que también es soportado en el escenario propuesto de recomendación basado en argumentación, la forma coloquial en la que las reglas están representados conduce a otra

característica: nuevos casos se pueden modelar mediante la adición de reglas en el sistema en cualquier momento, sin afectar a otras reglas en el sistema. Es decir, es posible añadir reglas para modelar nuevos aspectos de recomendación directamente mediante la codificación de nuevas reglas estrictas o rebatibles. Estas reglas serán considerados en las recomendaciones posteriores, brindando la posibilidad de modificar el universo de elementos recomendados para ciertos usuarios.

De esta manera es posible ampliar el conjunto de postulados para cambiar los criterios en los que un RS basado en argumentación genera sus sugerencias. Claramente, en cualquier entorno lo suficientemente dinámico (como suele ser el caso en todos los dominios de recomendación), adaptar el sistema a las intenciones y necesidades cambiantes del usuario es crucial para el éxito de las recomendaciones generadas. Básicamente, hay dos situaciones en las que se puede desear ampliar los aspectos que el recomendador tiene en cuenta. Por un lado, es posible que se necesite agregar una regla que refine a una ya existente. Por ejemplo, un refinamiento se considera conveniente si se descubre que cierta regla funciona bien en la mayoría de los casos, pero tiene un cierto número de contra ejemplos, y además, son conocidas cuáles son las características que comparten los contra ejemplos. Por otro lado, podría ser necesario añadir una nueva regla modelando una relación entre datos que no se han estado teniendo en cuenta, o si eventualmente se accede a datos con los que no se contaba anteriormente, como puede ser el caso cuando un nuevo conjunto de datos relacionados con el dominio de aplicación es desarrollado.

En cuanto al primer caso, una ventaja notable de la utilización de estas reglas de alto nivel para definir recomendaciones es que permiten incluir información contextual heterogénea, que es típicamente difícil de modelar en otros enfoques. Aunque algunos enfoques cuantitativos ya han abordado esta cuestión [LL07], la forma en que los mismos modelan las características contextuales (utilizando valores categóricos) sugiere que un enfoque cualitativo puede conducir a una forma más natural para tratar con estas características en el proceso de recomendación.

En el enfoque propuesto, si es posible tener acceso a la información contextual, es posible modelarla como predicados e incluirlos en los postulados para refinarlos. De esta manera, modelando y usando diferentes aspectos contextuales al mismo tiempo resulta más fácil (por ejemplo, clima, tiempo o el estado de ánimo del usuario), ya que no es necesario definir métricas de similitud para estos atributos. En su lugar, los aspectos

importantes podrían perfeccionarse mediante reglas más detalladas. Esto se ilustra con el ejemplo que se muestra a continuación.

Ejemplo 6.1 *Considerando la siguiente regla:*

$$\begin{aligned} \text{recommend}(\text{Track}, \text{User}) \quad \leftarrow \quad & \text{has_listened}(\text{User}, \text{Track}_1, \text{Artist}, l), \\ & \text{same_genre}(\text{Track}, \text{Track}_1). \end{aligned}$$

Por lo tanto, se recomienda una canción a un usuario basándose en el hecho de que le habían gustado canciones similares en el pasado. Sin embargo, se podría suponer que es sabido que, esto es típicamente cierto excepto cuando el género de la canción es de rock alternativo, el usuario está triste y el clima es lluvioso. Entonces, es posible refinar la regla anterior de la siguiente manera:

$$\begin{aligned} \sim\text{recommend}(\text{Track}, \text{User}) \quad \leftarrow \quad & \text{has_listened}(\text{User}, \text{Track}_1, \text{Artist}, l), \\ & \text{same_genre}(\text{Track}, \text{Track}_1), \\ & \text{genre}(\text{Track}, \text{"Alternative Rock"}), \\ & \text{mood}(\text{User}, \text{"Sad"}), \\ & \text{weather}(\text{"Rainy"}). \end{aligned}$$

Esto permite que el RS pueda modificar sus recomendaciones basadas en la información que tiene sobre el clima (mediante la conexión a uno de los varios servicios web que proporcionan esta información) y el estado de ánimo del usuario actual (por ejemplo, utilizando los servicios de inferencia del estado de ánimo como MoodScope [LLLZ13]).

La adición de reglas en recomendadores basados en argumentación también facilita la incorporación de nuevas dimensiones de análisis. Por ejemplo, es posible pensar que las preferencias de los usuarios cambian ligeramente con el tiempo, o que por lo menos se ven afectadas por éste. Si ese fuera el caso, el tiempo se puede utilizar como una característica contextual con el fin de generar mejores recomendaciones. Para lograr esto es posible añadir nuevas reglas que modifican las recomendaciones con el tiempo, usando formalismos basados en argumentación rebatible que toman en cuenta el tiempo [BLCS12].

Sin embargo, el refinamiento de reglas se puede utilizar para algo más que simplemente incluir información contextual. Si la retroalimentación de los usuarios está disponible, sería posible profundizar en aquellos casos en que la aplicación de un postulado resulta en falsos positivos o falsos negativos. Entonces, un refinamiento del postulado se puede formular con el fin de ayudar a superar el problema, como se ilustra en el siguiente ejemplo.

Ejemplo 6.2 *Considere la siguiente regla:*

$$\text{recommend}(\text{Movie}, \text{User}) \leftarrow \begin{array}{l} \text{top_genre}(\text{User}, \text{Genre}), \\ \text{genre}(\text{Movie}, \text{Genre}). \end{array}$$

Entonces, se le recomienda una película a un usuario determinado, basándose en el hecho de que al usuario le gusta su género. Sin embargo, se podría suponer que es sabido (a través de la retroalimentación de los usuarios) que recomendar películas de drama creadas en Hollywood a los usuarios que viven en América del Sur a menudo causa falsos positivos, ya que a dichos usuarios tiende a no gustarle esta combinación de género y origen de películas. Entonces, una nueva regla puede ser añadida al RS para indicar esto.

$$\sim\text{recommend}(\text{Movie}, \text{User}) \leftarrow \begin{array}{l} \text{top_genre}(\text{User}, \text{"Drama"}), \\ \text{genre}(\text{Movie}, \text{"Drama"}), \\ \text{origin}(\text{Movie}, \text{"Hollywood"}), \\ \text{lives_in}(\text{User}, \text{"South America"}). \end{array}$$

Hay que tener en cuenta, como se explicó antes, que la interacción entre las reglas es a nivel dialéctico, y por lo tanto pueden coexistir porque los posibles conflictos serán manejados por el proceso de argumentación. Es evidente que esto tiene un impacto directo en la flexibilidad del recomendador, como la de ofrecer la posibilidad de perfeccionar las recomendaciones para algún subconjunto de usuarios (o elementos) manteniendo el resto sin cambios. En el Ejemplo 6.2 fue refinado el conjunto de recomendaciones para los usuarios de América del Sur, sin perder el caso más general para los usuarios que viven en otras partes del mundo.

6.2. El rol de argumentación en recomendación

Como se mostrará a través de esta sección, el uso de argumentación permite dar prioridad a aquellos atributos que mejor definen una recomendación relevante. La argumentación también permite ofrecer explicaciones como conjuntos coherentes de razones a favor o en contra de una recomendación.

6.2.1. Recomendaciones basadas en argumentación

Esta sección describe cómo se construyen los argumentos, cómo se realiza el proceso dialéctico y cómo se utiliza para hacer recomendaciones a los usuarios. Como se discutió en el Capítulo 3, las reglas en un programa de DeLP se combinan para respaldar o rechazar una conclusión al construir argumentos. Un argumento puede estar a favor o en contra de una conclusión dada, pero no puede jugar ambos roles al mismo tiempo. A diferencia de otros sistemas de argumentación, los argumentos en DeLP se derivan del programa lógico que se está analizando y tienen una estructura interna. Esta estructura es lo que permite proporcionar una explicación sobre la aceptabilidad o no de una conclusión particular.

Como ya se ha mencionado anteriormente, el proceso de realizar una recomendación utilizando argumentación es bastante diferente a los enfoques más clásicos, a pesar de que persiguen la misma idea. Como parte del proceso argumentativo los argumentos atacan y son atacados para tratar de llegar a comprobar si es posible que una conclusión sea aceptada. En la Sección 3.2.1, se declaró que cada argumento atacado por al menos un argumento aceptado es derrotado, y que cada argumento sin atacantes (aceptados) se convierte en aceptado. Si el argumento raíz de un árbol dialéctico es aceptado, el árbol dialéctico representa un proceso de argumentación que apoya la conclusión de dicho argumento. Además, el razonador DeLP construye un conjunto de árboles dialécticos, tratando de dar soporte o contradecir los argumentos a favor o en contra de la consulta, y estos argumentos representan una explicación para dicha consulta.

Ejemplo 6.3 *Suponiendo que la consulta bajo análisis es:*

recommend(Gravity, Bob)

y que el RS de películas le otorga mayor prioridad al Postulado 4, es decir, el RS le dará preponderancia a aquellas películas en las que participan los actores que más le gustan al usuario y cuyo género es de su agrado, entonces, el sistema evaluará a los actores que interpretan la película Gravity (Sandra Bullock, George Clooney y Ed Harris), así como los géneros de la película (Suspense, Ciencia ficción) para determinar si a Bob le gustan algunos de los actores y/o algunos de los géneros. Dado que Sandra Bullock está entre los actores preferidos por Bob y a Bob le gustan las películas del género Suspense, esta evaluación será positiva.

La Figura 6.2 muestra los argumentos que se han calculado para hacer una recomendación al usuario *Bob*. En el Ejemplo 6.3, se puede ver cómo se pueden construir argumentos a favor o en contra de recomendar la película *Gravity* a *Bob*. La Figura 6.3 presenta un árbol dialéctico resultante de estos argumentos.

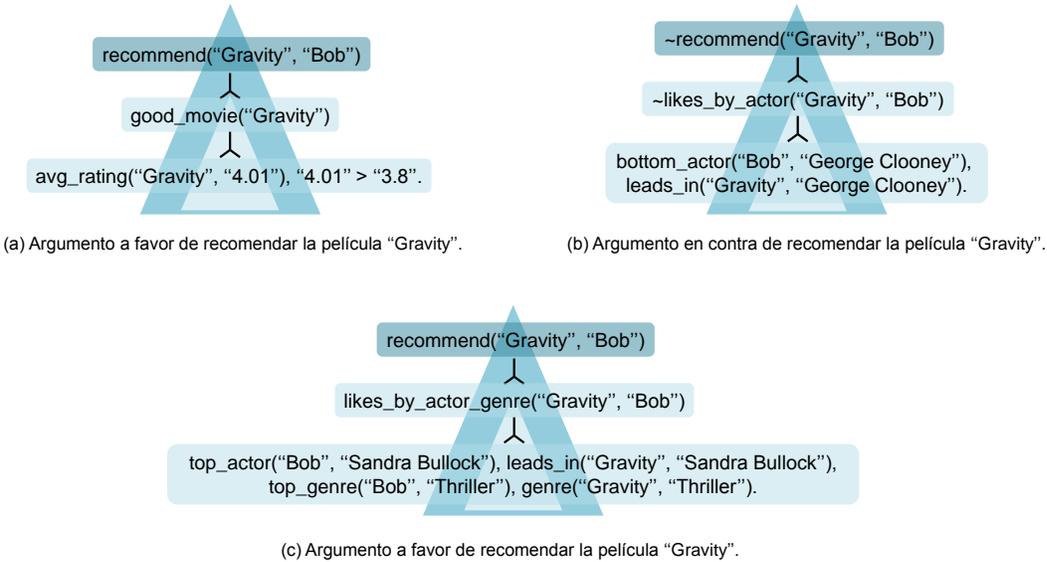


Figura 6.2: Argumentos a favor y en contra de recomendar una película.

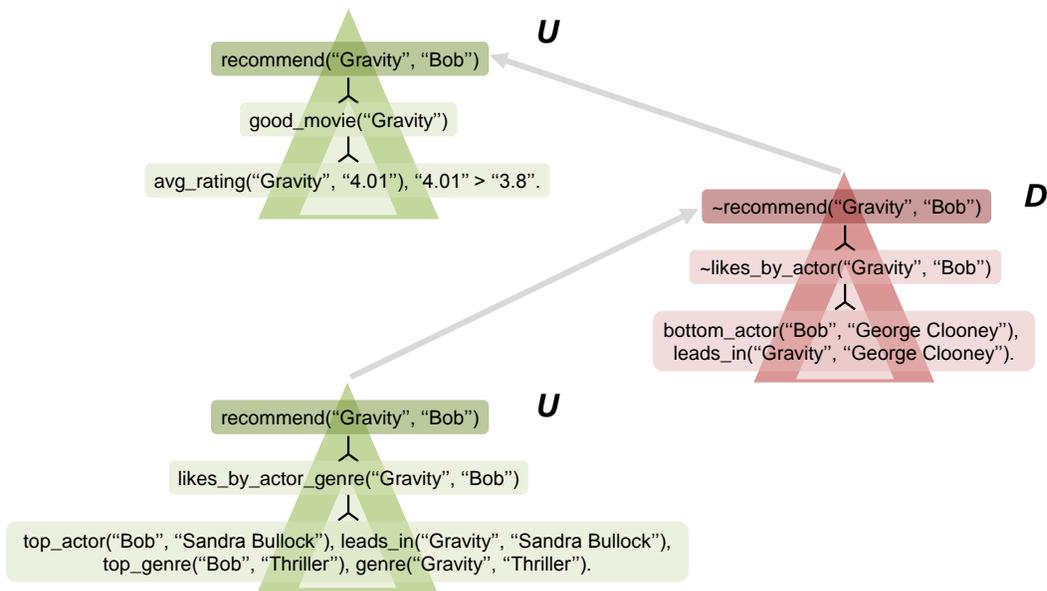


Figura 6.3: Árbol de dialéctica para recomendar una película.

Entonces, se tiene una razón para creer que al usuario *Bob* le gustaría la película *Gravity*, y de acuerdo con el sistema propuesto, esta razón es mejor que la razón para creer lo contrario, es decir, se le da prioridad a los argumentos creados a partir de esa regla sobre los demás. Por lo tanto, finalmente se recomienda la película al usuario.

6.2.2. Ofreciendo explicaciones para las recomendaciones

Como se mencionó anteriormente, a menudo las recomendaciones se consideraron como una “caja negra” y esta puede ser la razón por la que los RS han ganado mucho menos aceptación en dominios de alto riesgo, como paquetes de vacaciones o carteras de inversión, que en dominios de bajo riesgo como lo son el de la música, las películas o los libros. Es por este motivo que es importante proporcionar explicaciones complementando las recomendaciones. La explicación de la recomendación generalmente facilita la toma de decisiones de los usuarios, generando una mayor satisfacción y confianza en el sistema. Una explicación es una descripción que justifica una recomendación y ayuda a que los usuarios puedan determinar si el elemento recomendado es relevante para sus necesidades o no [TM12]. Como se mostró en el Capítulo 2 las explicaciones pueden perseguir distintos objetivos, pero además pueden presentarse al usuario de diferentes formas.

Una interfaz de explicación es una representación de la explicación proporcionada para un elemento sugerido por un RS. Existen varias interfaces de explicación utilizadas en la literatura, algunas de ellas son tradicionales y se pueden aplicar y adaptar a todos los dominios, y otras son específicas para cada dominio [DBB17, NJ17]. Algunas de las interfaces de explicación presentadas contienen datos en forma de calificaciones, como pueden ser un histograma de calificaciones, una tabla o un gráfico de torta de las calificaciones de los vecinos del usuario. Otro tipo de interfaz de explicación son las explicaciones basadas en nubes de etiquetas. Una de las interfaces de explicación tradicionales es la explicación en forma de una descripción textual que establece en lenguaje natural las razones para proporcionar las recomendaciones. Las ventajas de esta forma de explicación son que se adapta a todos los dominios y se puede usar en todos los casos [KSP⁺19], además de ser, en general, el tipo de interfaz que requiere menor esfuerzo cognitivo para poder ser comprendida.

En el Ejemplo 6.3, se ha expuesto la razón detrás de una recomendación utilizando un lenguaje coloquial para describir la estructura del argumento que ha prevalecido en el

análisis dialéctico. La simplicidad de poner la estructura de los argumentos prevalecientes en palabras del lenguaje natural lleva a otra ventaja del enfoque argumentativo. Dar explicaciones de las recomendaciones es bastante sencillo cuando se basan en argumentos, ya que pueden considerarse como un conjunto de razones que se explican por sí mismas. Como consecuencia, la implementación de un módulo para construir explicaciones de una recomendación basada en argumentos es una tarea sencilla. Todo lo que se necesita es mantener una asociación entre la estructura de los argumentos y los esquemas de explicaciones, donde los últimos están parametrizados por las variables en la estructura del primero. Esto puede realizarse de manera directa, ya que la cantidad de reglas definidas es finita (y a menudo pequeña). En la Figura 6.4 se muestra cómo es posible construir la asociación entre los postulados definidos para el RS de películas y su respectivo esquema de explicación, para poder ofrecerle al usuario las razones detrás de una recomendación.

Postulado	Estructura del argumento	Esquema de la razón detrás de la recomendación
P _{1,1}	$\langle \text{recommend}(\text{Movie}, \text{User}) , \text{good_movie}(\text{Movie}) \text{---} \langle \text{avg_rating}(\text{Movie}) > \text{AVG_Ratio} . \rangle \rangle$	<i>User</i> , se te recomienda la película <i>Movie</i> porque su calificación promedio está por encima del promedio general.
P _{2,1}	$\langle \text{recommend}(\text{Movie}, \text{User}) , \text{likes_by_genre}(\text{Movie}, \text{User}) \text{---} \langle \text{top_genre}(\text{User}, \text{Genre}), \text{genre}(\text{Movie}, \text{Genre}) . \rangle \rangle$	<i>User</i> , se te recomienda la película <i>Movie</i> porque su género es <i>Genre</i> , uno de tus favoritos.
P _{3,1}	$\langle \text{recommend}(\text{Movie}, \text{User}) , \text{likes_by_actor}(\text{Movie}, \text{User}) \text{---} \langle \text{top_actor}(\text{User}, \text{Actor}), \text{leads_in}(\text{Movie}, \text{Actor}) . \rangle \rangle$	<i>User</i> , se te recomienda la película <i>Movie</i> porque <i>Actor</i> , uno de tus actores favoritos es protagonista.
P _{4,1}	$\langle \text{recommend}(\text{Movie}, \text{User}) , \text{likes_by_actor_genre}(\text{Movie}, \text{User}) \text{---} \langle \text{top_actor}(\text{User}, \text{Actor}), \text{leads_in}(\text{Movie}, \text{Actor}), \text{top_genre}(\text{User}, \text{Genre}), \text{genre}(\text{Movie}, \text{Genre}) . \rangle \rangle$	<i>User</i> , se te recomienda la película <i>Movie</i> porque <i>Actor</i> , uno de tus actores favoritos es protagonista y además es de género <i>Genre</i> , uno de tus favoritos.

Figura 6.4: Explicaciones para los postulados positivos.

Si bien lo más importante es ofrecerle al usuario el porqué de una sugerencia, de manera similar a los esquemas planteados anteriormente, es posible definir esquemas de explicaciones, como se pueden ver en la Figura 6.5, que justifiquen por qué no es ofrecida una cierta recomendación.

Luego, cuando se obtiene un árbol dialéctico que soporta una recomendación, se puede usar el argumento de soporte en un nodo hoja para obtener los valores adecuados para las variables en el esquema de la explicación de la recomendación, obteniendo así una explicación completa que se puede presentar al usuario. Continuando con el escenario de recomendación introducido en el Ejemplo 6.3, se usa el argumento en el nodo hoja del

Postulado	Estructura del argumento	Esquema de la razón detrás de la no recomendación
P _{1.2}	$\langle \text{-recommend}(\text{Movie}, \text{User}), \text{-good_movie}(\text{Movie}) \text{-} \langle \text{avg_rating}(\text{Movie}) \leq \text{AVG_Ratio} \rangle \rangle$	<i>User</i> , no se te recomienda la película <i>Movie</i> porque su calificación promedio está por debajo del promedio general.
P _{2.2}	$\langle \text{-recommend}(\text{Movie}, \text{User}), \text{-likes_by_genre}(\text{Movie}, \text{User}) \text{-} \langle \text{bottom_genre}(\text{User}, \text{Genre}), \text{genre}(\text{Movie}, \text{Genre}) \rangle \rangle$	<i>User</i> , no se te recomienda la película <i>Movie</i> porque su género es <i>Genre</i> , uno de tus géneros menos preferidos.
P _{3.2}	$\langle \text{-recommend}(\text{Movie}, \text{User}), \text{-likes_by_actor}(\text{Movie}, \text{User}) \text{-} \langle \text{bottom_actor}(\text{User}, \text{Actor}), \text{leads_in}(\text{Movie}, \text{Actor}) \rangle \rangle$	<i>User</i> , no se te recomienda la película <i>Movie</i> porque, <i>Actor</i> , uno de tus actores menos preferidos es protagonista.
P _{4.2}	$\langle \text{-recommend}(\text{Movie}, \text{User}), \text{-likes_by_actor_genre}(\text{Movie}, \text{User}) \text{-} \langle \text{bottom_actor}(\text{User}, \text{Actor}), \text{leads_in}(\text{Movie}, \text{Actor}), \text{bottom_genre}(\text{User}, \text{Genre}), \text{genre}(\text{Movie}, \text{Genre}) \rangle \rangle$	<i>User</i> , no se te recomienda la película <i>Movie</i> porque, <i>Actor</i> , uno de tus actores menos preferidos es protagonista y además es de género <i>Genre</i> , uno de tus géneros menos preferidos.

Figura 6.5: Explicaciones para los postulados negativos.

árbol dialéctico que se muestra en la Figura 6.3 para justificar por qué se recomienda la película *Gravity* al usuario *Bob*. Esta explicación tomará la forma:

“Bob, se te recomienda la película Gravity porque, Sandra Bullock, uno de tus actores favoritos es protagonista y además su género es Suspense, uno de tus favoritos.”

Ofreciendo este tipo de explicaciones se podría ayudar a construir la confianza de los usuarios en el RS, como se ha estudiado en [TM07]. Además, es posible refinar las explicaciones aún más, mostrándole a *Bob* el conjunto de películas que le han gustado en las que *Sandra Bullock* es una protagonista, como la razón por la que se considera a esa actriz como una de las favoritas de *Bob*.

6.3. Evaluación empírica de los enfoques

En esta sección se evalúa el marco propuesto y se discuten los resultados de estas evaluaciones. En particular, se eligió evaluar el modelo presentado para el sistema de recomendación de películas. El objetivo de la evaluación presentada tiene un doble propósito. Por un lado, se lleva a cabo una comparación del desempeño de los postulados propuestos en la Sección 6.1.1, con el fin de descubrir cuáles son los más adecuadas para el desarrollo del sistema propuesto de recomendación de películas basado en argumentación. Por otro lado, se compara el rendimiento de un recomendador cuantitativo básico con el rendimiento de un sistema de recomendación mixto que refina recomendaciones cuantitativas con aspectos cualitativos.

Sin importar el doble objetivo de la evaluación, los experimentos completados para evaluar el desempeño de los recomendadores implican las mismas acciones: se toma una película y se trata de predecir si al usuario le gusta la película o no. Para llevar a cabo esta evaluación, se han completado una serie de pruebas, que se describen a continuación.

6.3.1. Definición del conjunto de datos

Los datos que se han utilizado para realizar las pruebas se basan en un conjunto de datos de *GroupLens*², conocido como el conjunto de datos *MovieLens 10M*³. Este conjunto de datos contiene 10.000.054 calificaciones. Las calificaciones están en una escala de 1 a 5 estrellas. El conjunto de datos contiene 95.580 etiquetas asignadas por 71.567 usuarios a 10.681 películas del servicio de recomendación de películas en línea *MovieLens*⁴. Para completar las pruebas realizadas, los usuarios fueron seleccionados al azar, pero con la salvedad de que, todos estos usuarios habían calificado al menos 20 películas. En este caso, a diferencia de otros conjuntos de datos ofrecidos por *GroupLens*, no se dispone de información demográfica. Cada usuario está representado por un id, sin proporcionar ningún tipo de información sobre el mismo.

La Figura 6.6 muestra la estructura de los tres archivos que contienen los datos sobre películas, calificaciones y etiquetas.

Movies	Ratings	Tags
ID	ID de la película	ID del usuario
Título (Año)	ID del usuario	ID de la película
Lista de géneros	Calificación	Etiqueta
	Estampilla de tiempo	Estampilla de tiempo

Figura 6.6: Estructura de los archivos de datos.

De entre los conjuntos de datos proporcionadas por *GroupLens* fue seleccionado *MovieLens 10M*, con el propósito de mostrar que el sistema de recomendación propuesto basado en DeLP es capaz de hacer frente a cantidades de datos masivas. Con el propósito de basar las recomendaciones en ciertos atributos que no se encuentran disponibles en el

²<https://grouplens.org/>

³<https://grouplens.org/datasets/movielens/10m/>

⁴<https://movielens.org/>

conjunto de datos seleccionado, se tomo la decisión de aumentarlo. Para llevar a cabo esta tarea, se tomo a *Internet Movie Database (IMDb)* como fuente adicional, para agregar estos atributos faltantes. En particular, cada película del conjunto original de *GroupLens* fue aumentada con datos como directores, actores principales, guionistas, etc. Estos atributos adicionales constituyen un ingrediente importante para estudiar el impacto de los aspectos cualitativos en el rendimiento de recomendación. La Figura 6.7 ilustra el conjunto de atributos resultante que ha sido utilizado para caracterizar una película.

Información de una película
ID
Título
Año
Fecha de estreno
URL de IMDB
Calificada
Duración
Calificación
Contador de calificaciones de IMDB
ID de IMDB
Lista de actores
Lista de directores
Lista de escritores
Lista de géneros

Figura 6.7: Datos almacenados por cada película.

Cabe mencionar que *GroupLens* también ha desarrollado conjuntos de datos con características similares al generado para éste trabajo, es decir aumentando al conjunto con datos de *IMDb*. Actualmente, el conjunto de datos más grande de *GroupLens* que se encuentra disponible es *MovieLens 20M*⁵, pero el mismo aún no existía al momento de la realización de este trabajo. Otro conjunto de datos similar creado por *GroupLens*, que sí se encontraba disponible, es *MovieLens HetRec*⁶. Sin embargo, el conjunto de datos utilizado para llevar a cabo los experimentos que ha resultado luego de aumentarlo, es del mismo orden de magnitud que *MovieLens 20M* y es más de diez veces mayor que

⁵<https://grouplens.org/datasets/movielens/20m/>

⁶<https://grouplens.org/datasets/hetrec-2011/>

MovieLens HetRec. El conjunto de datos utilizado en este trabajo se puede descargar en <http://ir.cs.uns.edu.ar/datasets>.

6.3.2. Configuración experimental

Como se mencionó previamente, el conjunto de datos utilizado para realizar la evaluación del RS de películas, se obtuvo del conjunto de datos *MovieLens 10M* ofrecido por *GroupLens* luego de aumentarlo. Para la evaluación del RS se tomaron 3000 calificaciones del conjunto de datos, definiendo así el conjunto de prueba. Estas calificaciones fueron seleccionadas al azar, con la única salvedad de que todas ellas provenían de diferentes usuarios. Garantizar que las calificaciones utilizadas para la configuración del experimento se asociaran con diferentes usuarios, brindó la capacidad de evaluar los postulados para el caso general, evitando la posible introducción de sesgo proveniente del comportamiento de los usuarios particulares. Por otro lado, una película puede aparecer más de una vez en el experimento, ya que puede aparecer la misma película en dos calificaciones procedentes de dos usuarios diferentes.

El objetivo de esta evaluación fue para determinar si el sistema era capaz de predecir si al usuario le gusta la película o no. Por lo tanto, para cada calificación en el conjunto de prueba se le plantea al sistema la consulta $recommend(Movie, User)$ para obtener la predicción del sistema. Para asegurar que el triplete (movie, user, rating) siendo validado no tuvo influencia en la predicción hecha, fue quitado el registro correspondiente del conjunto de entrenamiento.

Una observación importante acerca de los posibles resultados dados por un sistema de recomendación basado en argumentación es que los posibles resultados pueden ser positivos (es decir, la película se recomienda al usuario), negativos (es decir, la película no se recomienda al usuario) o indecisos. El último caso se presenta cuando el sistema no puede justificar recomendar o no recomendar una película a un usuario, lo que lleva a un resultado indeciso como se explico previamente en la Sección 3.2.1.

Hay que hacer hincapié en que tales resultados indecisos no deben interpretarse como que el recomendador está sugiriendo que una determinada película no debería ser recomendada a un usuario particular. Es decir, los resultados indecisos no son un tipo particular de recomendación negativa. En su lugar, significa que el recomendador ni afir-

ma ni niega la recomendación. Posteriormente se tendrá una mejor visión de cómo los resultados indecisos se consideran para medir la eficacia de las recomendaciones.

Cabe mencionar, que el valor promedio de todas las calificaciones del conjunto de datos *MovieLens 10M* es igual a 3.8. Como se aclaró en la Sección 6.1.1 es necesario calcularlo para poder instanciar el valor de *AVG_Rating* en el *Postulado 1*. Además es en base a este valor que se considerará cuándo una película obtuvo una buena o mala calificación.

Para realizar las pruebas, se plantearon cada una de las 3000 consultas al recomendador. Entonces, se clasificó cada respuesta en una de las siguientes clases, de acuerdo con la respuesta obtenida y la calificación que el usuario dio a la película:

- Verdadero positivo (TP): el sistema sugiere que al usuario le va a gustar la película, y el usuario calificó la película con un valor de cuatro o más.
- Verdadero negativo (TN): el sistema indica que al usuario no le va a gustar la película, y el usuario calificó la película con tres o un menor valor.
- Falso positivo (FP): el sistema indica que el usuario debe ver la película, y el usuario calificó la película con tres o un menor valor.
- Falso negativo (FN): el sistema indica que el usuario no debe ver la película, y el usuario calificó la película con un valor de cuatro o más.
- Indeciso (UN): el sistema no sugiere ni niega la recomendación.

Este experimento fue repetido para diferentes subconjuntos de postulados, permitiendo realizar una comparación entre estos postulados. Esta comparación se discutirá en detalle más adelante.

6.3.3. Métricas de rendimiento

Con el fin de evaluar los diferentes aspectos de la familia de recomendadores resultante, se han utilizado varias métricas de rendimiento de IR. Si bien estas métricas son comunes en entornos de IR, algunas de ellas no son totalmente apropiadas para este escenario, donde se tienen resultados indecisos. Esto se debe a que los resultados indecisos no son exactamente una clase por sí misma debido a que las recomendaciones de películas no son intencionadamente clasificadas como indecisos, sino que quedan sin resolver por el sistema.

Entonces, no se está en presencia de un problema de clasificación multi-clase, pero no se trata de una clasificación puramente binaria tampoco. Por ejemplo, no es posible tomar directamente los resultados obtenidos en una matriz de confusión estándar de dos por dos. Más bien, se tiene una clasificación binaria aumentada con el estado indeciso adicional.

A continuación se presentan las métricas de rendimiento utilizados en la evaluación de la familia de sistemas de recomendación, indicando si estos indicadores pueden estar influenciados por la presencia de resultados indecisos, y cómo se resuelve el problema en caso de que así sea.

Precisión (del inglés *Precision*). Esta es una medida tradicional que mide cómo de exactas son las recomendaciones dadas por el sistema. La precisión es definida como el número de recursos relevantes recuperados dividido por el número total de recursos recuperados. A menudo, la precisión de un sistema se ve como una medida de la calidad de los recursos recuperados. En el escenario propuesto de recomendación de películas, la precisión se calcula como el número de películas que el sistema predice correctamente como aquellas películas que el usuario desea ver (es decir, el número de verdaderos positivos), dividido por el número total de películas recomendadas positivamente (es decir, la suma de verdaderos positivos y falsos positivos). En resumen, se tiene que:

$$\text{Precisión} = \frac{TP}{TP + FP}$$

Hay que tener en cuenta que los resultados indecisos no afectan a la precisión del recomendador, lo cual tiene sentido ya que esas películas para las que la respuesta es indecisa no son recuperadas.

Cobertura (del inglés *Recall*). Mientras que la precisión es sobre la calidad, la cobertura involucra la noción de cantidad y mide qué tan exhaustivas son las recomendaciones formuladas por el sistema. Se define como el número de recursos relevantes recuperados dividido por el número de los recursos relevantes. En el dominio de la recomendación de películas se puede calcular dividiendo el número de películas correctamente recomendadas por el número total de películas que vale la pena recomendar (sin tener en cuenta si la predicción es positiva, negativa o indecisa). Por lo tanto, la cobertura debe ser influenciada por los resultados indecisos, así como los positivos y negativos, pero no es posible

afirmar si una respuesta indecisa significa una recomendación para la película o no. Por esta cuestión, se ha optado por utilizar la siguiente fórmula, que considera directamente el número de películas relevantes en el conjunto de datos:

$$Cobertura = \frac{TP}{RM}$$

Donde RM es el número de casos (de las 3000 calificaciones probadas) para los que la calificación real dada por el usuario es igual o superior a cuatro. Esto significa que RM representa el número total de tripletes (movie, user, rating) en el conjunto de pruebas donde la calificación es mayor o igual a 4 (lo que significa que la película debería ser recomendada, y por lo tanto es relevante). Hay que tener en cuenta que dicha medida puede ser inutilizable en algunos entornos de recomendación en los que no se puede obtener el número de elementos relevantes.

Exactitud (del inglés *Accuracy*). Esta métrica representa la fracción de los recursos predichos como positivos o negativos para los que la predicción fue correcta. De esto se deduce que las respuestas indecisas no deben influir en la exactitud ya que para estos casos no se hace ninguna predicción. Entonces, se utiliza la fórmula habitual para la exactitud, que es como sigue:

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

F-Calificación (del inglés *F-Score*). Una medida que considera tanto la calidad y la cantidad, al mismo tiempo, del conjunto de recursos recuperados por un sistema es la F-Calificación, que es la media armónica ponderada de precisión y cobertura. La fórmula general para la F-Calificación ponderada por un parámetro β , que se denota como $F_{(\beta)}$, es:

$$F_{\beta} - Calificación = \frac{(1 + \beta^2) \times (Precisión \times Cobertura)}{(\beta^2 \times Precisión + Cobertura)}$$

Donde β es un número real no negativo tal que si $\beta = 1$ entonces la fórmula mide la cobertura y la precisión con el mismo peso, si $\beta < 1$ entonces la precisión se considera más importante, y de otra manera la cobertura es considerada más importante.

6.3.4. Resultados

A continuación se presentan los resultados obtenidos en los experimentos que se llevaron a cabo para probar los diferentes enfoques de recomendación. El primer conjunto de resultados arrojó luz sobre la eficacia de diferentes combinaciones de postulados. Luego, el segundo conjunto de resultados se utilizó para comparar distintas formulaciones de recomendación mixta, es decir recomendadores que consideren aspectos tanto cuantitativos como cualitativos, con distintos enfoques puramente cuantitativos.

Eficacia de recomendación de los diferentes postulados

Con el fin de comparar la eficacia de los diferentes conjuntos de postulados, se han calculado las métricas propuestas en la Sección 6.3.3 para cada uno de los postulados, así como para diferentes combinaciones de ellos. Como observación, el número de registros con películas relevantes (es decir, aquellos registros de los 3000 registros del conjunto de prueba donde el usuario calificó la película con un valor igual o superior a cuatro) es 1977. Como se mencionó previamente, este valor es necesario para poder calcular la cobertura. La Figura 6.8 presenta los resultados para los diferentes conjuntos de postulados utilizando el conjunto de prueba que consiste en 3000 clasificaciones. Las métricas de rendimiento calculadas para las diferentes combinaciones de postulados se resumen en la Figura 6.9.

Postulados \ Tipo de respuesta	TP	TN	FP	FN	UN
Todos los postulados	1619	585	441	355	0
Postulado 1	1449	609	414	528	0
Postulado 2	551	163	142	132	2012
Postulado 3	576	97	57	21	2249
Postulado 4	129	17	3	2	2849
Postulados 2 + 3	991	234	194	117	1464
Postulados 2 + 4	551	163	142	132	2012
Postulados 3 + 4	529	85	47	14	2325
Postulados 2 + 3 + 4	991	234	194	117	1464

Figura 6.8: Resultados para diferentes conjuntos de postulados.

Cabe observar que la precisión más baja (0.78) se obtiene cuando solo se utiliza el *Postulado 1*. Este caso puede ser tomado como una línea de base para la comparación. El *Postulado 1* solo tiene en cuenta la información cuantitativa. Por otro lado, se consiguió

Postulados \ Métrica	Precisión	Cobertura	Exactitud	$F_{(1)}$	$F_{(0.5)}$
Todos los postulados	0.79	0.82	0.44	0.80	0.79
Postulado 1	0.78	0.73	0.41	0.75	0.77
Postulado 2	0.80	0.28	0.24	0.41	0.58
Postulado 3	0.91	0.29	0.25	0.44	0.64
Postulado 4	0.98	0.07	0.07	0.12	0.26
Postulados 2 + 3	0.84	0.50	0.35	0.63	0.58
Postulados 2 + 4	0.80	0.28	0.24	0.41	0.58
Postulados 3 + 4	0.92	0.27	0.23	0.41	0.62
Postulados 2 + 3 + 4	0.84	0.50	0.35	0.63	0.58

Figura 6.9: Resumen del desempeño de diferentes conjuntos de postulados.

el mejor valor de precisión (0.98) cuando se utiliza el *Postulado 4*. Este postulado combina información cualitativa sobre los actores y los géneros que al usuario le gustan. Sin embargo, es claro que la alta precisión alcanzada por el *Postulado 4*, se produce a costa de una cobertura muy pobre (0.07), que se puede interpretar como que el recomendador basado en este postulado solo recomienda una película a un usuario si tiene una alta confianza de que al usuario le va a gustar.

Además, es interesante observar que los postulados cualitativos (*Postulados 2, 3 y 4*) son superiores en términos de precisión a la combinación de postulados cualitativos y cuantitativos (*Todos los postulados*). Sin embargo, hay que observar que el enfoque cuantitativo logra una cobertura considerablemente superior que los enfoques cualitativos. Mientras que para el dominio de recomendación de películas la precisión es más importante que cobertura, otros dominios de aplicación, tales como el diagnóstico médico, pueden beneficiarse enormemente de lograr una alta cobertura. Como observación final sobre esta cuestión, habría que hacer hincapié en que, para el caso cuando se consideran en conjunto todos los postulados, el número de casos resueltos por las recomendaciones cualitativas para los experimentos presentados es la suma de los casos resueltos considerando los *Postulados 2, 3, y 4* en conjunto, es decir, la suma de los casos resueltos en la última fila en la Figura 6.8, que suma un total de 1536. Por lo tanto, cuando se consideran conjuntamente los postulados cuantitativos y cualitativos en un recomendador mixto, más de 50% de los casos (1536 de 3000) se resuelven tomando en cuenta aspectos que surgen del enfoque cualitativo.

Enfoque cuantitativo vs. enfoque mixto

El objetivo específico de esta parte de la experimentación es arrojar algo de luz sobre la cuestión de si aumentar un recomendador cuantitativo básico con aspectos cualitativos de hecho da lugar a mejoras significativas.

Para medir esto, se han planteado 3000 consultas a dos programas DBI-DeLP diferentes. El primer programa tiene como único caso de recomendación al *Postulado 1*, y por lo tanto se trata de un recomendador puramente cuantitativo utilizando solo la calificación promedio de las películas para las recomendaciones. El segundo programa representa el enfoque mixto y utiliza los *Postulados 1, 2, 3 y 4* para determinar si una película en particular debería ser recomendada para algún usuario o no. Como observación, para el recomendador mixto el criterio de prioridad entre reglas (o postulados) se instancia de la siguiente manera:

$$\textit{Postulado 4} > \textit{Postulado 3} > \textit{Postulado 2} > \textit{Postulado 1}.$$

La razón detrás de esta elección es que, para el escenario de recomendación de películas, se prefiere dar prioridad a la precisión sobre la cobertura, ya que es preferible recomendar una pequeña cantidad de películas a los usuarios si eso significa que los usuarios tenderán a estar de acuerdo con las recomendaciones ofrecidas. Entonces, se ha optado por establecer esta prioridad de acuerdo a los resultados que se muestran en la Sección 6.3.4. Hay que observar que el esquema de prioridades presentado significa que los postulados más importantes para este recomendador en particular son los cualitativos. Por lo tanto, el desacuerdo entre el enfoque cuantitativo y algunos de los enfoques cualitativos siempre se resuelve a favor de los cualitativos. En la práctica, esto hace que las respuestas obtenidas en función de los aspectos cuantitativos sean refinadas (invalidándolos) por los aspectos cualitativos, incluso en los casos en que las recomendaciones coinciden.

Postulados \ Tipo de respuesta	TP	TN	FP	FN
Recomendador cuantitativo	1449	609	414	528
Recomendador mixto	1619	585	441	355

Figura 6.10: Resultados obtenidos por los enfoques cuantitativos y mixto.

Los resultados de comparar el enfoque puramente cuantitativo con el enfoque mixto se encuentran resumidos en la Figura 6.10.

Con el fin de determinar si las mejoras logradas por los recomendadores mixtos fueron estadísticamente significativas, se dividieron las 3000 calificaciones en el conjunto de prueba en 30 muestras de 100 elementos cada una. Tomando como base los resultados obtenidos se realizó un análisis estadístico. La Figura 6.11 presenta los resultados obtenidos por el enfoque cuantitativo, y la Figura 6.12 presenta los resultados del enfoque mixto. Se muestran las medias (MEAN), las desviaciones estándar (STDEV) y los intervalos de confianza al 95 % (95 % CI) para cada tipo de respuesta.

Tipo de respuesta \ Parámetro	MEAN	STDEV	95% C.I.
TP	48.30	4.92	[46.54, 50.06]
TN	20.30	4.04	[18.86, 21.74]
FP	13.80	2.68	[12.84, 14.76]
FN	17.60	3.73	[16.27, 18.93]
TP + TN	68.60	4.38	[67.03, 70.17]

Figura 6.11: Media, desviación estándar e intervalo de confianza al 95 % para el enfoque cuantitativo.

Tipo de respuesta \ Parámetro	MEAN	STDEV	95% C.I.
TP	53.97	5.16	[52.14, 55.80]
TN	19.50	4.37	[17.94, 21.06]
FP	14.70	3.17	[13.56, 15.83]
FN	12.17	3.73	[10.83, 13.50]
TP + TN	73.47	3.32	[72.28, 74.65]

Figura 6.12: Media, desviación estándar e intervalo de confianza al 95 % para el enfoque mixto.

6.3.5. Análisis de los resultados

Se pueden extraer varias conclusiones de los experimentos anteriores, considerando primero cuál es la mejor combinación de los postulados en el escenario propuesto de recomendación de películas y segundo, si el refinamiento de un enfoque cuantitativo básico con aspectos cualitativos de hecho logra mejorar el rendimiento.

Análisis de la eficacia de los postulados

Las métricas de precisión y cobertura indican la eficacia de un sistema de recomendación desde dos perspectivas diferentes: la calidad y la cantidad de las recomendaciones. Típicamente, el aumento de la cobertura de un sistema viene a expensas de una disminución de su precisión [BG94, Alv02]. Como puede verse en el rendimiento del recomendador propuesto, la compensación entre la precisión y la cobertura está presente aquí también. Aquellas combinaciones de postulados que dan lugar a la precisión más baja, son aquellas que logran la más alta cobertura, mientras que la que tiene la mejor precisión (*Postulado 4*) tiene una cobertura muy baja.

Ciertamente, los diferentes dominios de aplicación darán diferente prioridad a la precisión y a la cobertura. Como se ha indicado antes, en ciertos dominios tales como el de diagnósticos médicos la cobertura se considera con frecuencia como más importante que la precisión, ya que es aceptable aumentar el número de falsos positivos si eso significa también aumentar el número de posibles causas sugeridas para un síntoma dado. Esto minimizaría la posibilidad de dejar una enfermedad sin ser detectada. Sin embargo, en el dominio de la recomendación de películas se considera que la calidad es un aspecto más importante que la cantidad. Es decir, es preferible recomendar un pequeño número de películas al usuario que, con alta certeza, le van a gustar que presentar un gran número de recomendaciones poco fiables. Hay que observar que no hay ninguna ventaja real en recomendar a los cientos de usuarios miles de películas ya que los usuarios seguramente no podrán ver todas al mismo tiempo. Además, presentar demasiadas recomendaciones de películas probablemente significa que varias de ellas están mal recomendadas, desperdiciando el tiempo de los usuarios, además de no cumplir con el objetivo del sistema como recomendador. También, puede ser el caso de que después de que los usuarios han visto las películas recomendadas piden recomendaciones más recientes, posiblemente teniendo en cuenta la nueva información (tanto la calificación otorgada a las películas recomendadas y películas más recientes), en el supuesto de que los conjuntos de datos se actualizan.

Teniendo en cuenta estos ajustes, se centró el análisis del desempeño de los postulados dando prioridad a la precisión sobre la cobertura. Por otra parte, se centró el análisis en las recomendaciones positivas (es decir, a favor de ver una película) y no en las negativas (es decir, a favor de no ver una película). Los resultados obtenidos con respecto a TP, TN, FP y FN se muestran en la Figura 6.13. Un gráfico completo de precisión y cobertura se puede ver en la Figura 6.14.

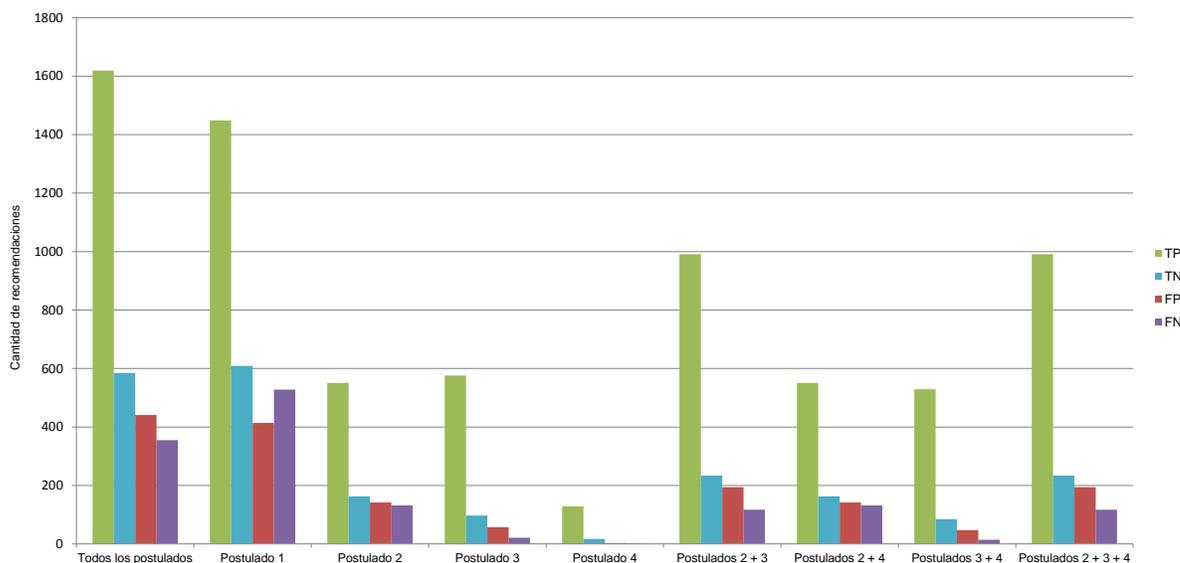


Figura 6.13: Los resultados obtenidos por las diferentes combinaciones de postulados.

Es importante destacar que los desempeños de todos los postulados evaluados es bastante bueno, teniendo en cuenta que la precisión más baja alcanzada fue de 0.78. A primera vista, es evidente que el *Postulado 4* (es decir, recomendar una película a un usuario basado en los actores y géneros de películas favoritos del usuario *al mismo tiempo*) es el enfoque más preciso, y prácticamente no falla cuando se hace una recomendación, con solo cinco respuestas incorrectas (tres de ellas son falsos positivos) entre las 141 recomendaciones que ha ofrecido (más de 132 recomendaciones positivas). Además, hay que observar que la gran proporción de *indecisos* del *Postulado 4* (2849 de 3000) afecta severamente la cobertura, que se explica por el hecho de que, siendo el postulado más específico, es difícil encontrar dos razones cualitativas (el género de la película y los actores) que soportan simultáneamente la recomendación.

Otro punto importante a notar es que el *Postulado 3* (que recomienda basado solo en los actores de una película) tiene también una muy alta precisión (la diferencia con el *Postulado 4* es de solo 0.07) y mejora en gran medida la cobertura (una diferencia de 0.22 a favor del *Postulado 3*). Aunque ya se mencionó que para este dominio de aplicación la precisión es más importante que la cobertura, esta pequeña diferencia en la calidad de los resultados, a lo largo de la diferencia significativa en la cantidad de resultados recomendados hace del *Postulado 3* un contendiente destacado para el *Postulado 4*.

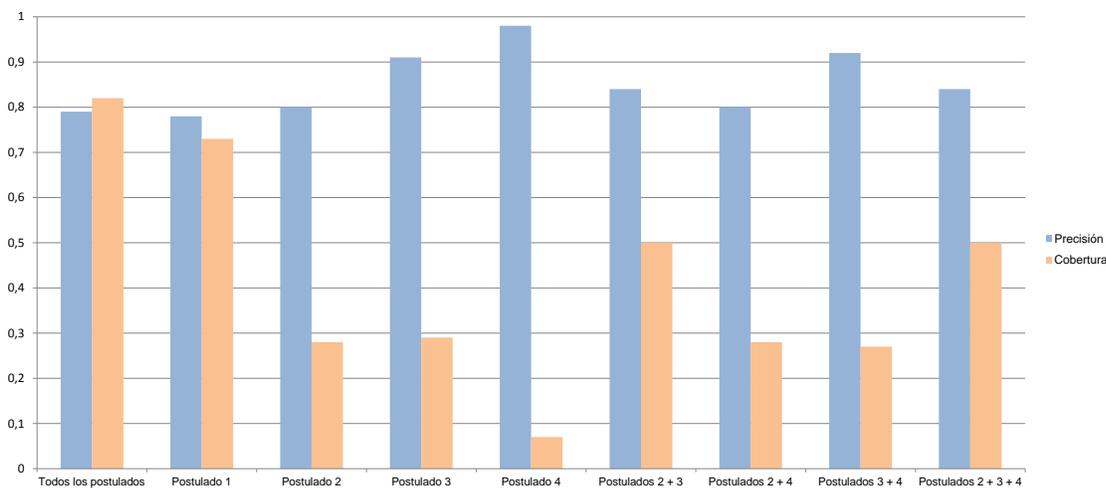


Figura 6.14: La precisión y cobertura obtenida combinando diferentes postulados.

Esto es más evidente si tenemos en cuenta sus $F_{(0,5)}$ -calificación (que prefiere precisión sobre cobertura), con una diferencia de 0.38 a favor del *Postulado 3*. Además, la combinación del *Postulado 3* y el *Postulado 4* tiene también un muy buen desempeño, como se esperaba.

Como observación final, hay que hacer hincapié en que la diferencia en la precisión entre recomendar teniendo en cuenta los actores favoritos de los usuarios (*Postulado 3*) y teniendo en cuenta los géneros de películas favoritos de los usuarios (*Postulado 2*) parece indicar que los actores que actúan en una película son mejores descriptores de las preferencias de los usuarios que los géneros de una película.

Análisis del enfoque cuantitativo vs. enfoque mixto

La siguiente parte del análisis tiene la intención de descubrir si hay una mejora significativa en el rendimiento del enfoque cuantitativo cuando se refinan sus respuestas por medio de aspectos cualitativos. La primera particularidad interesante a destacar es que esta combinación ha dado como resultado un aumento en el número de verdaderos positivos pero no se observaron otras mejoras. Por lo tanto, parece que, cuando se trata de recomendar *negativamente* una película, la diferencia entre los dos enfoques es pequeña y por otra parte se observa una sutil mejora en el enfoque cuantitativo. Sin embargo, como

se ha señalado antes, las recomendaciones más importantes para el escenario propuesto de recomendación de películas son los casos positivos.

Se puede ver que, para los 3000 casos probados, el enfoque mixto obtiene 170 verdaderos positivos más que el enfoque cuantitativo. Además, los resultados presentados en la Figura 6.11 y la Figura 6.12 indican que en los resultados del enfoque mixto hay mejoras significativas en el número de TP sobre el enfoque cuantitativo. Por último, hay que observar que las mejoras también son estadísticamente significativas si se tiene en cuenta todo el conjunto de respuestas *correctas* obtenidas por los enfoques, es decir, si se tiene en cuenta el número de TP y TN juntos, incluso cuando ese no es el caso si solo consideramos la proporción de TN de los recomendadores solos.

Los resultados experimentales presentados en esta sección destacan el hecho de que la incorporación de los aspectos cualitativos en un recomendador no solo ayuda a proporcionar transparencia, sino también pueden resultar en mejoras significativas con respecto a los enfoques cuantitativos más básicos.

6.3.6. Comparación con algunos métodos del estado del arte usados en la recomendación de películas

Con el fin de completar el análisis de la propuesta, se proporciona una comparación entre el enfoque mixto propuesto en este capítulo y algunos enfoques del estado del arte en la recomendación, derivados de técnicas cuantitativas. Para evaluar el desempeño de técnicas de recomendación propuestas para el dominio de las películas, la mayoría de los trabajos realizan un experimento offline mediante un conjunto de datos previamente recogidos de las calificaciones de los usuarios. Suelen ser adoptadas dos metodologías principales para la presentación de reportes de rendimiento. Por un lado, la predicción del sistema se considera como un problema de clasificación binaria y por lo tanto se calculan métricas como la precisión, cobertura y sus derivados. Por otro lado, si el sistema intenta predecir la calificación numérica que un usuario le daría a una película, se informa habitualmente una medida de la exactitud de las calificaciones del sistema de predicción, como la raíz del error cuadrado medio (RMSE).

Como se ha reconocido en investigaciones previas [GBC⁺14], la segunda metodología no es adecuada para escenarios donde son posibles solo los resultados booleanos (como recomendaciones positivas y negativas), como es el caso de la propuesta presentada para

realizar una recomendación basada en argumentación. Por lo tanto, el análisis comparativo es restringido a los trabajos que adoptan un enfoque de evaluación orientado a la precisión. Entre estos trabajos, no ha sido posible encontrar ningún ejemplo en la literatura que use exactamente el mismo conjunto de datos empleado en este trabajo, el conjunto de datos *MovieLens 10M*, sino que los experimentos se realizaron principalmente utilizando versiones más limitadas, tales como *MovieLens 100K*⁷ y *MovieLens 1M*⁸. Debido a estas limitaciones, una comparación completamente justa entre el enfoque propuesto y otras técnicas de vanguardia no es posible. Como consecuencia, se ofrece una comparación aproximada entre el enfoque propuesto y otras alternativas recientes diferentes. El primer método seleccionado para la comparación se basa en una nueva métrica de similitud bayesiana [GZY13], el segundo método es un enfoque basado en contenido compatible con Linked Open Data [NMO⁺12], el tercer método es un enfoque híbrido basado en algoritmos de estimación de la distribución (en inglés, *Estimation of Distribution Algorithms*) [LLFZ14], y el cuarto método es un enfoque basado en filtrado colaborativo usando una ontología y técnicas de reducción de dimensionalidad [NIB18].

Es necesario hacer hincapié en que, debido a esas diferencias, proporcionar una comparación justa entre los enfoques es una tarea muy difícil. Es decir, no es apropiado comparar directamente las precisiones sin tener en cuenta ciertas diferencias en el ambiente de recomendación. Luego, en cada comparación se establecen esas diferencias. En lo que sigue se presentan los resultados obtenidos por métodos probados en los conjuntos de datos mencionados, y se proporcionan las comparaciones, tanto en los resultados propiamente dichos como en los métodos utilizados para obtenerlos.

Comparación con un recomendador basado en una métrica de similaridad bayesiana

En [GZY13], se introduce un nuevo método de similitud bayesiano (BS) para medir la similitud de los usuarios. La medida de similitud propuesta se basa en la distribución de Dirichlet, teniendo en cuenta tanto la dirección y la longitud de los vectores de calificación. Esta medida ayuda a superar varios problemas que vienen del cálculo de la similitud del coseno o el coeficiente de correlación de Pearson en los enfoques de filtrado colaborativo.

⁷<https://grouplens.org/datasets/movielens/100k/>

⁸<https://grouplens.org/datasets/movielens/1m/>

Uno de los beneficios más importantes de esta novedosa medida de similitud es que disminuye la correlación por azar (donde los usuarios se consideran “similares” solo debido a un pequeño número de elementos clasificados conjuntamente). En el mencionado trabajo se informa sobre un análisis empírico en seis conjuntos de datos del mundo real, donde los experimentos indican que el método BS genera similitudes entre usuarios más realistas y distinguibles que los métodos de línea de referencia (*baselines*) seleccionados para la comparación.

Con el fin de comparar el método BS con el enfoque propuesto, es necesario centrarse en los experimentos reportados por los autores para un escenario de recomendación de películas basado en el conjunto de datos *MovieLens 100K* de *GroupLens*. Es importante mencionar que una comparación directa de los valores de precisión obtenidos por BS y el enfoque mixto propuesto ciertamente no será justa, ya que presentan una precisión en los resultados de n ($p@n$), yendo de 2 a 20 recomendaciones. Hay que tener en cuenta que no se han calificado ni restringido el número de recomendaciones consideradas del total de 3000 casos, ya que solo se tienen respuestas binarias, es decir, recomendar o no recomendar. Por lo tanto, en cierto sentido, los resultados obtenidos se pueden reportar como de “precisión a rango 3000”. Mientras que el rendimiento obtenido por BS es muy alto, la precisión disminuye a medida que crece la lista de recomendaciones. Mientras que para 2 recomendaciones, la precisión obtenida es 0.9801, para $n = 20$ la precisión es 0.8338. Por lo tanto, considerando que se ha obtenido un valor de precisión de 0.98 para una lista de longitud 3000 (*Postulado 4*), se puede concluir que el rendimiento en términos de precisión del enfoque propuesto no es inferior al obtenido por el método BS. Sin embargo, como se esperaba, hay una diferencia significativa en términos de cobertura a favor del método BS. Sin embargo, como se argumentó anteriormente, se considera que en un escenario de recomendación de películas la cobertura no es un factor crucial, y puede ser preferible perder algo de cobertura para obtener beneficios que podrían ser más significativos, como la interpretabilidad de los resultados y las explicaciones en lenguaje natural.

Comparación con un recomendador compatible con Linked Open Data

Un enfoque diferente a la recomendación de películas se toma en [NMO⁺12], donde la idea clave es aprovechar la información codificada en RDF disponible en la web de datos (en inglés, *Web of Data*). Esta información se utiliza para desarrollar un sistema

de recomendación basado en contenido que aprovecha los datos disponibles dentro de los conjuntos de datos de Linked Open Data (LOD), en particular, *DBpedia*, *Freebase* y *LinkedMDB*. Sobre la base de tal información, los autores proponen un enfoque de modelado de espacio vectorial que se utiliza para determinar similitudes entre recursos RDF en LOD para generar recomendaciones.

Hay algunas similitudes en el espíritu del enfoque mixto y el enfoque LOD. Por ejemplo, el uso de técnicas basadas en contenido (aunque, como se indicó anteriormente, en el enfoque mixto es posible implementar el filtrado colaborativo de manera casi directa). Además, también utilizan la calificación de usuario binaria, esto les permite estar en línea con las tendencias en la Web social 2.0 (me gusta/no me gusta), como expresan los autores. De manera similar al enfoque mixto, la recomendación basada en LOD aumenta la información asociada con las películas utilizando fuentes de datos adicionales. En este caso, los datos proporcionados por *GroupLens* están vinculados a fuentes LOD (*DBpedia*). Otra similitud con la propuesta de este capítulo es que el recomendador basado en LOD asume que si dos películas comparten cierta información (por ejemplo, parte del reparto), entonces están relacionadas entre sí, explotando dicha relación en el momento de definir un perfil de usuario. Por otro lado, a diferencia del enfoque mixto, la similitud se determina de forma cuantitativa, en función del gráfico RDF asociado a los recursos.

El recomendador basado en LOD fue evaluado utilizando el conjunto de datos *MovieLens 1M* de *GroupLens*, y como en el enfoque de BS, se informan precisión y cobertura a diferentes rangos. El enfoque se compara con otros sistemas de recomendación basados en palabras clave que no utilizan LOD, demostrando que el enfoque propuesto los supera. Como suele ser el caso, la precisión tiende a disminuir (y la cobertura a aumentar) cuando se toman en cuenta más recomendaciones. Si solo se considera la recomendación del tope de la lista, es decir ($p@1$), entonces el enfoque logra una precisión de 0.834 con una cobertura de 0.06. Para el rango máximo calculado ($p@20$) el recomendador logra una precisión de 0.707, con una cobertura perfecta. Una vez más, de acuerdo con los resultados observados, se puede concluir que el enfoque mixto supera al enfoque basado en LOD en términos de precisión, y aunque tiene una cobertura inferior, los resultados obtenidos no son sustancialmente inferiores que la mayoría de los resultados reportados.

Comparación con un recomendador basado en algoritmos de estimación de distribución

Otro enfoque para la recomendación de películas se presenta en [LLFZ14], donde se introduce un nuevo sistema de recomendación híbrido que aprovecha el filtrado colaborativo y los mecanismos basados en contenido. Una característica sobresaliente de este sistema es la aplicación de algoritmos de estimación de la distribución (EDA) para conocer las preferencias de los usuarios. Las preferencias de los usuarios se combinan en los perfiles de interés de los usuarios, que se utilizan para describir con precisión las características de interés de los usuarios. Sobre la base de los perfiles de interés del usuario, el mecanismo basado en contenido del modelo puede recomendar nuevos elementos a los usuarios. Además, el mecanismo de filtrado colaborativo sufre menos del problema de la dispersión porque la similitud del usuario está determinada por los perfiles de usuario en lugar de por la matriz de elementos-usuarios.

El enfoque EDA se prueba utilizando una extensión del conjunto de datos *MovieLens 1M* de *GroupLens*. Al igual que en el enfoque mixto propuesto, las características asociadas con cada película se aumentan con información adicional (directores, escritores, actores, géneros, etc.) obtenida de *IMDb*. El enfoque basado en EDA se compara con otros enfoques, como lo son el kNN (para $k = 30$) y el bayesiano naïve, demostrando que los supera. La precisión a rango 5 alcanzada por el enfoque basado en EDA es de 0.955, mientras que la precisión a rango 10 es de 0.873. Como se mencionó anteriormente, en el enfoque mixto propuesto no se ha restringido el número de recomendaciones considerando la totalidad de las 3000 realizadas, ya que solo se tienen respuestas binarias, es decir, recomendar o no. Por lo tanto, en cierto sentido, los resultados del enfoque mixto pueden verse como una precisión a rango 3000 ($p@3000$). La precisión general y los valores de cobertura también se calculan con diferentes ajustes de parámetros para los EDA, logrando una precisión que fluctúa entre 0.736 y 0.772, y una cobertura que varía de 0.734 a 0.726. Aunque la precisión obtenida es excelente, siempre que la longitud de la lista de elementos recomendados aumente, la precisión disminuye. Mientras que para 5 recomendaciones la precisión obtenida es de 0.955, para 10 recomendaciones la precisión es de 0.873. Por lo tanto, considerando solamente el *Postulado 4*, se ha medido la precisión para una lista de 3000 recomendaciones que alcanzan precisiones de hasta 0.98, por lo que se puede concluir que el desempeño del enfoque mixto en esos términos no es en absoluto inferior al obtenido por este otro enfoque. Una vez más, se está en presencia de un sistema cuya precisión es

inferior a la alcanzada por el enfoque mixto. Por otro lado, la cobertura obtenida por el enfoque basado en EDA es superior al logrado por el enfoque mixto, excepto cuando se utilizan todos los postulados, donde el enfoque mixto logra una cobertura de 0.82.

Comparación con un recomendador basado en filtrado colaborativo usando ontología y técnicas de reducción de dimensionalidad

El trabajo presentado en [NIB18] desarrolla un nuevo método de recomendación híbrido basado en el enfoque de filtrado colaborativo, utilizando técnicas de reducción de dimensionalidad y una ontología. Aquí se usa la ontología para mejorar la precisión de las recomendaciones en la parte del filtrado colaborativo. En esta parte, también utilizan una técnica de reducción de dimensionalidad, la de descomposición en valores singulares (SVD, del inglés *Singular Value Decomposition*), para encontrar los elementos y usuarios más similares en cada grupo de elementos y usuarios, lo que puede mejorar significativamente la escalabilidad del método de recomendación.

El recomendador basado en Ontología + SVD fue evaluado utilizando el conjunto de datos *MovieLens 1M* de *GroupLens*, y como en otros de los enfoques seleccionados, se informan precisión y F1 a diferentes rangos. Al igual que en el enfoque mixto propuesto, distintas características asociadas con cada una de las películas fueron aumentadas con información adicional obtenida de *IMDb*. Para evaluar el método propuesto utilizando métricas de precisión de sistemas de soporte de decisión, la precisión y F1 se calcularon a diferentes rangos n ($p@n$). En esta investigación, se consideró $n = 5, 10, 20, 30, 40$ y 50 , es decir, el método se evaluó recomendando las mejores 5, 10, 20, 30, 40 y 50 películas según el sistema de recomendación propuesto. En cambio en el enfoque mixto, como ya se observó anteriormente, se puede decir que los resultados son reportados en “precisión a rango 3000”.

La precisión lograda por el método Ontología + SVD va incrementándose a medida que aumenta n , hasta que se consideran 25 recomendaciones, donde alcanza una máxima precisión de 0.833, y a partir de ese punto, a mayor número de recomendaciones, como suele ser el caso, la precisión tiende a disminuir. El método Ontología + SVD tiene una precisión bastante pareja yendo desde las 5 a las 50 recomendaciones y oscila entre 0.787 y 0.833. Si se comparan estos valores con el enfoque que considera solo los postulados cualitativos (*Postulados 2, 3 y 4*) se observan valores muy similares en términos de precisión, aunque en cobertura el método Ontología + SVD es superior. En cambio si se

comparan con los valores obtenidos del *Postulado 4* se puede concluir que éste es superior en términos de precisión, sin embargo hay una diferencia considerable en términos de cobertura. De todos modos, como ya se argumentó con anterioridad, en el escenario de recomendación de películas, puede ser preferible perder algo de cobertura para obtener otras ventajas más significativas, como las explicaciones en lenguaje natural.

Comparación final

En resumen, la Figura 6.15 presenta una comparación final del desempeño de los métodos analizados. En esta figura se presenta el mejor rendimiento obtenido por cada uno de los enfoques en términos de precisión, incluyendo también el número de recomendaciones para el que se obtuvo como resultado.

A partir del análisis de la Figura 6.15 se puede concluir que la eficacia del enfoque mixto propuesto es de ninguna manera significativamente inferior a la de los otros enfoques. Además, es importante destacar las principales ventajas de la propuesta de este enfoque:

- **Transparencia.** Aunque los usuarios finales no pueden percibir el proceso dialéctico que conduce a la recomendación (que claramente no es necesario), tienen acceso a explicaciones en lenguaje natural claras de los motivos subyacentes a las recomendaciones dadas. Esta característica ayuda en la construcción de la confianza de los usuarios y mejora la capacidad de retroalimentación del sistema.
- **Flexibilidad.** El uso de reglas facilita la definición de criterios cuantitativos y cualitativos para guiar el proceso de recomendación. Esto se complementa con un enfoque simple para definir criterios de preferencia de los argumentos. Por otra parte, el enfoque propuesto hace posible pasar de un sistema de recomendación basado en contenido a uno basado en filtrado colaborativo con bastante facilidad, o incluso combinar los enfoques de manera arbitraria, simplemente cambiando las prioridades entre las reglas.

Para escenarios como el que se presenta en este capítulo, tales ventajas pueden ayudar a superar algunos de los problemas de la recomendación, y que vale la pena mantenerlas incluso cuando se producen a costa de un (pequeño) decremento en el desempeño. Por otra parte, para algunos otros escenarios, los enfoques puramente cuantitativos han demostrado ser adecuados y prácticos. Tanto la investigación sobre enfoques cuantitativos

Método	Resultados @ n	Precisión	Cobertura
Todos los postulados	n = 3000	0.79	0.82
Postulados cualitativos (2 + 3 + 4)	n = 3000	0.84	0.50
Solo Postulado 4	n = 3000	0.98	0.07
BS	n = 2	0.98	0.45
LOD	n = 1	0.83	0.06
EDA	n = 5	0.95	0.73
Ontología + SVD	n = 25	0.83	0.85

Figura 6.15: Comparación con los métodos del estado del arte en la recomendación de películas.

puros como la investigación sobre sistemas de recomendación basados en argumentación tienen un gran potencial, tanto por sí mismos como en conjunto, planteando preguntas importantes e interesantes. En términos generales, se podría decir que los enfoques cuantitativos y cualitativos no son opuestos ni excluyentes, sino que son complementarios. Podrían beneficiarse enormemente entre sí, de la misma manera que lo hacen las técnicas basadas en contenido y de filtrado colaborativo.

Como suele ocurrir en inteligencia artificial en general, y en los sistemas expertos en particular, la elección de un enfoque sobre el otro (o una combinación de ellos) dependerá en gran medida del dominio de aplicación y las necesidades de los usuarios. No parece haber un mejor enfoque general, y es el dominio de la aplicación y las necesidades de los usuarios lo que determina cuál es el mejor método para adoptar. Por ejemplo, para el escenario de recomendación de películas, la posibilidad de proporcionar explicaciones en lenguaje natural a las recomendaciones parece ser lo suficientemente atractiva, incluso si esto es a costa de una pérdida de precisión o cobertura. En otros escenarios, se podría argumentar que las explicaciones no son tan importantes, mientras que lograr un alto rendimiento es crucial.

6.4. Trabajo relacionado

Hasta donde sabemos, este es el primer intento de proporcionar un enfoque basado en argumentación para la recomendación de películas con la capacidad de justificar los elementos sugeridos. Sin embargo, el problema de proporcionar recomendaciones comple-

mentadas con justificaciones ya se ha estudiado en otros trabajos. En [SS02] se presenta un análisis en el que se examina el papel de la transparencia en los sistemas de recomendación, es decir, la importancia de que un usuario entienda por qué se hizo una recomendación particular. En ese trabajo, se presenta un estudio que indica que los usuarios prefieren recomendaciones transparentes a no transparentes. Para nuevos elementos, este resultado no es sorprendente, sin embargo, el estudio también muestra que los usuarios prefieren saber por qué se recomendó un artículo incluso para aquellos que ya conocen y les gustan. Esto sugiere que los usuarios no solo buscan recomendaciones ciegas de un sistema, sino que también buscan una justificación de la elección del sistema. En [BM05] se resuelve que la contribución más importante de las explicaciones no es convencer a los usuarios a adoptar recomendaciones (*promoción*), sino permitirles tomar decisiones más informadas y precisas sobre qué recomendaciones utilizar (*satisfacción*). En [GJG14] se revela que las explicaciones representadas mediante una nube de etiquetas basadas en el contenido son particularmente útiles para aumentar el nivel de transparencia percibido por el usuario y para aumentar la satisfacción del usuario, a pesar de que exigen un mayor esfuerzo cognitivo por parte del usuario. Además, los autores presentan evidencia que muestra que la satisfacción es un requisito previo a la confianza, siendo la transparencia percibida por el usuario importante para la satisfacción y la confianza del usuario. Cuando la eficiencia no tiene un efecto importante en la satisfacción y la confianza, la persuasión positiva y negativa puede causar la pérdida de confianza de los usuarios. Su análisis también reveló una fuerte relación entre la transparencia y la satisfacción. También llegaron a la conclusión de que los tipos de explicación tienen diferentes efectos en los usuarios, como decisiones más rápidas, mayor transparencia o mayor satisfacción. En [CFLH14] se estudió si la idea general de aprender de las explicaciones funciona y demostraron que las explicaciones pueden considerarse como una valiosa fuente de conocimiento que puede ser brindada por un RS. Esta investigación se centra en el uso de las opiniones de los vecinos (sobre los elementos previamente calificados por el usuario) en las explicaciones. Los autores concluyen que es posible aprender de un conjunto de explicaciones, aunque esto depende en gran medida del usuario, y se puede usar un procedimiento automático para analizar el papel de las diferentes características presentadas en una explicación. El RS de películas *MoviExplain* [SNM09] proporciona una justificación al presentar las características de la película que le podrían gustar al usuario en una tabla. Esta tabla muestra en una columna el nombre de la película, en otra llamada “*El motivo es*” indica la característica preferida del usuario, y por último, otra columna “*Porqué calificó*” indica el historial de

calificaciones del usuario. Otros enfoques basados en texto para respaldar la justificación se han estudiado en [TM11].

En [TM07] se proporciona una revisión muy amplia de las explicaciones en los sistemas de recomendación. En su trabajo, argumentan que “una buena explicación podría ayudar a inspirar confianza y lealtad de los usuarios, aumentar la satisfacción, hacer que los usuarios encuentren lo que quieren más rápida y fácilmente, y persuadirlos para que prueben o compren un artículo recomendado”. Además, se puede ver que las explicaciones deben poseer ciertas características. Por ejemplo, se puede medir qué tan comprensible es una explicación, lo que puede contribuir a la transparencia, como así también a la confianza y satisfacción del usuario.

Varios trabajos de investigación sobre sistemas de recomendación se han centrado en el dominio específico de la recomendación de música o de películas [CNHAVGGS12, HZF⁺13, OGDN⁺13, PZS14, KSP⁺19]. Por ejemplo en [BHC98] presentan un enfoque que utiliza la información de calificación y las preferencias del usuario para implementar un recomendador social basado en contenido de películas. En su trabajo, presentan un enfoque de recomendación de aprendizaje inductivo, formalizando el problema de recomendación como un problema de aprendizaje. El enfoque propuesto utiliza *Ripper*, un sistema de inducción de reglas, para aprender una función que toma un usuario y una película y predice si la película será de su agrado o no, combinando información colaborativa y de contenido. En otro trabajo [MMN02], proponen un marco para implementar un recomendador de películas híbrido. Este enfoque, denominado recomendación de filtrado colaborativo aumentado por contenido, utiliza información de contenido externo para producir predicciones para nuevos usuarios o nuevos elementos. El recomendador resultante resulta en un rendimiento de recomendación superior al de algunos recomendadores puramente basados en contenido o puramente basados en filtrado colaborativo. En [TH12] utilizan clustering y proponen un recomendador basado en filtrado colaborativo combinado con métodos basados en contenido. Para realizar el clustering los autores utilizaron k-means y mapas auto-organizados (SOM) mientras que para sus experimentos recurrieron al conjunto de datos de *MovieLens*. En otro trabajo [WCL14] desarrollaron un recomendador híbrido que combina el enfoque basado en contenido con filtrado colaborativo bajo un modelo unificado denominado co-clustering con matrices aumentadas (CCAM). El recomendador desarrollado fue evaluado en los conjuntos de datos *MovieLens 100K* y *MovieLens 1M*, demostrando que la información basada en contenido puede ayudar a reducir el problema

de la escasez al minimizar la pérdida de información de las matrices de datos basadas en CCAM. En otra propuesta [LHC16] desarrollaron un sistema de recomendación de películas basado en datos relacionales utilizando los datos de una ontología del dominio. Los autores plantearon algoritmos genéticos para el proceso de recomendación y utilizaron el conjunto de datos *MovieLens* para la evaluación del método mediante el uso de métricas como cobertura, diversidad y precisión. En [CPCP16] desarrollaron un sistema de recomendación híbrido que combina enfoques basados en contenido, filtrado colaborativo y conocimiento del contexto. Además, se utilizan técnicas de la web semántica para modelar la información del RS. Específicamente, la ontología del recomendador se definió con OWL 2. El método propuesto es evaluado utilizando el conjunto de datos de *MovieLens 100K* mediante las métricas F1, cobertura y precisión. Otro trabajo [MSL⁺16] propone un marco completo para abordar conjuntamente los problemas de escalabilidad, escasez, primer calificador y arranque en frío al combinar métodos de minería de la web y ontologías específicas del dominio. Las evaluaciones reportadas se basan en el conjunto de datos de *MovieLens 100K* usando la métrica precisión. En [KA17] desarrollaron un RS usando el sistema de interferencia difuso basado en redes adaptables (ANFIS) para sistemas de recomendación con múltiples criterios, incorporando información demográfica de los usuarios e información semántica ontológica basada en elementos. Los autores evaluaron el método propuesto utilizando los datos de la plataforma *Yahoo! Movies* mediante las métricas F1, MAE y precisión. Los resultados de este trabajo revelaron que el uso de información semántica mejora la precisión predictiva de los sistemas de recomendación de criterios múltiples. En la propuesta presentada en [SLC⁺17] se desarrolla un RS utilizando el filtrado colaborativo. En el enfoque propuesto, se utilizan datos demográficos estándar de usuarios, como el género, la edad y la ocupación. Los vecinos se identifican mediante el cálculo de la distancia euclidiana. Las recomendaciones se hacen en base a las películas mejores calificadas del vecino más cercano. Las recomendaciones también son sensibles al tiempo para mantenerse al día con los gustos cambiantes del usuario. Los resultados se evalúan utilizando las métricas de precisión, cobertura y F-calificación.

A diferencia de las alternativas anteriores, el trabajo propuesto en este capítulo incorpora inferencia basada en argumentación para el proceso de recomendación de películas o canciones. Para respaldar la recomendación de una película a un usuario, el sistema realiza un análisis cualitativo y cuantitativo sobre diversos aspectos de la película (como podrían ser el género, los actores, los directores, los guionistas, etc.) y sobre las preferencias que tiene un usuario en particular. Además, el recomendador no solo proporciona

una respuesta binaria (recomendar o no recomendar) sino que también proporciona una justificación para esta respuesta.

El problema de potenciar la recomendación con argumentación también ha sido abordado en [AP09, CMG09, BV15, FTG⁺17]. En el trabajo realizado en [GCRS13] se ofrecen explicaciones para dar soporte a las conclusiones del proceso argumentativo. Además, en [TGGS15] se propone un mecanismo para que el usuario seleccione los criterios de comparación de argumentos. Otros análisis teóricos y simulaciones dirigidas a combinar aspectos cuantitativos y cualitativos para generar recomendaciones basadas en argumentos se presentaron en [BCM13]. El trabajo presentado en este capítulo extiende estos resultados al describir un marco novedoso para los dominios específicos de las recomendaciones de películas o canciones. Lo más importante es que presenta un estudio empírico en el que se utilizan métricas de evaluación clásicas para determinar la efectividad de la propuesta.

6.5. Conclusiones

Se ha introducido un enfoque basado en argumentación con la capacidad de mejorar las tecnologías de recomendación. El enfoque propuesto tiene algunos puntos fuertes que pueden ser útiles en escenarios de recomendación. Por ejemplo, la argumentación se puede utilizar para realizar un análisis cualitativo de usuarios y elementos. Esto permite ir un paso más allá con respecto al enfoque clásico, dando lugar a sistemas de recomendación mixtos, es decir, sistemas que basan sus recomendaciones en aspectos tanto cualitativos como cuantitativos del dominio. En particular, este enfoque puede hacer que sea más fácil tomar en consideración varios aspectos antes de dar la recomendación final al usuario. Estos aspectos se pueden aplicar para modelar algunas características útiles que a veces son difíciles de tomar en cuenta en enfoques puramente cuantitativos. Por ejemplo, la naturaleza rebatible de las preferencias de los usuarios en entornos complejos. Los enfoques cuantitativos para la recomendación se basan típicamente en el uso de una serie de parámetros, lo que hace que la interpretación de los resultados no sea intuitiva. En la propuesta ofrecida en este capítulo, es posible expresar de manera coloquial la razón detrás de cada recomendación. Además, es posible modelar restricciones a las recomendaciones basadas en información contextual heterogénea (por ejemplo, el clima o el estado de ánimo de los usuarios).

Una característica útil del formalismo es que para incluir un nuevo aspecto (*postulado*) es posible expresarlo en la forma de una o más reglas DeLP, e indicar la preferencia asignada a estas reglas. Este modelado de aspectos basado en reglas permite dar recomendaciones basadas en criterios cuantitativos (si las reglas los modelan), o criterios cualitativos. Además, el uso de preferencias lleva a otra ventaja, ya que es posible pasar del enfoque basado en contenido al enfoque basado en filtrado colaborativo con bastante facilidad, simplemente cambiando las prioridades entre las reglas. Otra característica útil es el uso de estructuras coherentes de razones (*argumentos*) para generar explicaciones para los elementos recomendados. Estas explicaciones razonadas son útiles para generar confianza en los usuarios y favorecer la retroalimentación en el sistema.

También se proporciona un análisis comparativo entre el enfoque propuesto y algunas técnicas del estado del arte para la recomendación de películas. De este análisis se desprende que el rendimiento del enfoque propuesto es comparable al de otros enfoques existentes. Aunque no es superior, tampoco es sustancialmente inferior. Es importante enfatizar que las principales ventajas del enfoque propuesto con respecto a otros métodos del estado del arte son su transparencia (que promueve la confianza de los usuarios) y su flexibilidad (que facilita la modelización de diferentes aspectos de una manera natural).

Finalmente, es importante mencionar que los experimentos realizados mostraron que el proceso de argumentación en el que se basan las recomendaciones puede tratar con cantidades masivas de datos, como el conjunto de datos *MovieLens 10M*, ya que el tiempo de respuesta promedio para las consultas fue de aproximadamente dos segundos. Por lo que sabemos, esta es la primera vez que se prueba la programación en lógica rebatible (más precisamente, su extensión que tiene soporte para interactuar con bases de datos) en un conjunto de datos tan grande del mundo real (como lo es *MovieLens 10M*), proporcionando un precedente para su uso en sistemas expertos masivos del mundo real.

Capítulo 7

Conclusiones y trabajo a futuro

En la actualidad, existe una gran sobrecarga de información en Internet y muchas veces las personas tienen dificultades para encontrar las cosas que se ajustan a su gusto. Las personas necesitan filtrar la información, por lo que los RS son de gran ayuda. Los RS se volvieron populares debido a sus beneficios tanto para las empresas como para los usuarios. Las empresas desean ofrecer al usuario información específica para aumentar las ventas, mientras que los usuarios desean llegar fácilmente a los elementos relevantes.

Se podría decir que la precisión de las recomendaciones ha sido un foco importante en trabajos de investigación previos, por no decir el más importante. Sin embargo, existen otras cuestiones que también son muy importantes cuando las personas eligen y utilizan los sistemas de recomendación. La calidad percibida por el usuario de las recomendaciones no se basa únicamente en la precisión de la predicción, sino también en otras nociones. Para determinar la utilidad de una recomendación, estos factores también deben tenerse en cuenta.

En esta tesis se desarrollaron distintos modelos para la implementación de RS basados en argumentación, que incorporan aspectos o nociones que difieren de las más clásicas de un recomendador. Uno de los modelos desarrollados incorpora la noción de confiabilidad, mientras que otro fomenta la transparencia de la recomendación para los usuarios y la flexibilidad para instanciar las preferencias de recomendación en un RS. En la propuesta, no solo se intenta priorizar la incorporación de nociones importantes no tan comunes en un RS, sino que también, los desarrollos tienen el agregado de haberse logrado utilizando un enfoque poco común en el área como lo es argumentación, permitiendo así, de una

manera dialéctica, combinar tanto aspectos cuantitativos como cualitativos para arribar a una recomendación.

La idea general fue la de la creación de modelos de sistemas de recomendación a partir de un marco de argumentación rebatible. Para mejorar el soporte al usuario por medio de argumentación, los criterios de preferencia del usuario se modelan mediante hechos, reglas estrictas y rebatibles codificadas como parte de un programa DeLP. Estos criterios de preferencia se combinan con información de fondo adicional, referente a algún dominio de recomendación en particular, y son utilizados por el marco argumentativo para priorizar posibles sugerencias, mejorando así los resultados finales proporcionados al usuario activo.

Flexibilidad

En general, los RS convencionales, para formular recomendaciones, se basan en índices de popularidad o en medidas de similitud entre usuarios o elementos, calculados sobre la base de métodos provenientes de las ciencias sociales, la recuperación de información o el aprendizaje automático. La interacción hombre-máquina observada en los métodos de recomendación existentes es particularmente rígida. Dichos servicios no proporcionan mecanismos para reformular fácilmente los criterios bajo los cuales se basará una recomendación. Esto significa que una vez que se adopta una técnica de recomendación particular y se asigna un conjunto de valores de parámetros (que normalmente se establecen mediante la configuración del sistema de recomendación), el usuario no tiene la posibilidad de introducir naturalmente nuevas preferencias para que sean consideradas por el sistema. Las decisiones sobre las preferencias de los usuarios están basadas principalmente en heurísticas, que se basan en clasificar las elecciones anteriores de los usuarios o en la recopilación de información de otros usuarios con intereses similares. En otras palabras, los sistemas de recomendación existentes no proporcionan un mecanismo de interacción para tratar de forma natural y metódica las preferencias de los usuarios en entornos complejos.

Otro problema que enfrentan los sistemas de recomendación es que los criterios de preferencia de los usuarios generalmente incluyen un conocimiento incompleto y potencialmente inconsistente sobre el dominio. Esto se debe al hecho de que las preferencias de los usuarios son dinámicas y, por lo general, cambian a medida que el tiempo evoluciona o cuando se dispone de nuevo material para su análisis. Debido a que los enfoques cuantitativos adoptados por la mayoría de los sistemas de recomendación no están equipados

con un mecanismo para revisar las conclusiones previas, la naturaleza cambiante de las preferencias de los usuarios no se aborda adecuadamente. La adopción de un enfoque más cualitativo al aumentar la recomendación con inferencia de lógica clásica no resolverá los problemas mencionados anteriormente, ya que a menudo conducirá a conclusiones contradictorias, que son problemáticas a menos que se traten de manera adecuada.

En los modelos propuestos en los Capítulos 5 y 6 se ha definido un enfoque basado en argumentación con la capacidad de mejorar las tecnologías de recomendación. El enfoque propuesto tiene algunas fortalezas que ofrecen ventajas en escenarios de recomendación. Es posible utilizar argumentación para realizar un análisis cualitativo de los usuarios y los elementos. Esto permite ir un paso más allá con respecto al enfoque clásico, dando lugar a sistemas de recomendación mixtos, es decir, sistemas que basan sus recomendaciones en criterios cualitativos y cuantitativos.

Este enfoque permite modelar algunas características útiles que a veces son difíciles de tener en cuenta en los enfoques puramente cuantitativos. Por ejemplo, el carácter rebatible de las preferencias de los usuarios en entornos complejos.

En particular, los criterios de preferencia de los usuarios fueron modelados utilizando DeLP, un marco para la argumentación rebatible basado en la programación lógica que puede tratar efectivamente información incompleta y contradictoria. El uso de DeLP permite integrar el razonamiento dialéctico en el proceso de recomendación, que proporciona una base razonada para los elementos sugeridos al usuario. Además de ofrecer un medio natural para codificar datos sobre el dominio de recomendación, DeLP trata eficazmente la naturaleza rebatible de los patrones de razonamiento que surgen normalmente en la recomendación.

Una característica útil de los modelos de RS propuestos en esta tesis es que permiten flexibilizar cómo priorizar los aspectos de recomendación. En cada modelo, los aspectos a tener en cuenta para realizar una recomendación son modelados por postulados representados por reglas DeLP, las cuales tienen asignadas un orden de preferencia. El modelado de aspectos de recomendación basado en reglas permite dar recomendaciones basadas en criterios cuantitativos (si las reglas lo modelan), o criterios cualitativos. El uso de preferencias entre reglas ofrece la ventaja de permitir pasar de un RS con un enfoque basado en contenido a uno con un enfoque basado en filtrado colaborativo con bastante facilidad, simplemente cambiando las prioridades entre las reglas.

Confiabilidad

Un comportamiento común es buscar sugerencias de personas en las que se confía para decidir el mejor lugar para adquirir algún servicio o la mejor fuente para obtener información sobre un determinado tema. Las sugerencias de personas confiables también pueden ayudar a decidir en quién más confiar. Tratando de imitar este comportamiento que se da en la vida real, se desarrolló un modelo para un RS de noticias que contempla la noción de confiabilidad.

La dinámica de la confiabilidad de las noticias se ha estudiado principalmente a través de enfoques cuantitativos. Sin embargo, una perspectiva puramente cuantitativa de la confiabilidad de las noticias tiene varias limitaciones. En particular, los enfoques cuantitativos hacen que sea difícil proporcionar a los lectores una justificación de por qué ciertas noticias deben ser confiables, o pueden ser incapaces de lidiar con nociones sutiles como la desconfianza o la reparación de confiabilidad. Resulta más atractivo combinar criterios cuantitativos y cualitativos para filtrar y calificar las noticias. En este sentido, los métodos cuantitativos pueden ayudar a determinar si el tema de las noticias es relevante para el interés del usuario, mientras que los criterios cualitativos pueden ayudar a decidir si las noticias provienen de una fuente confiable. Por ejemplo, un enfoque cualitativo es más natural para tratar algunas propiedades deseables de la confiabilidad.

En el Capítulo 5 se presentó un modelo para un sistema de recomendación que incorpora la noción de confiabilidad. El primer aporte del modelo propuesto es el análisis de un conjunto de propiedades deseables de la confiabilidad, tales como ser subjetiva, dinámica, rebatible, dependiente del contexto y no siempre simétrica o transitiva. Se mostró que estas propiedades pueden ser modeladas de forma natural con el modelo propuesto.

Otra contribución principal fue la prueba de dos teoremas fundamentales que caracterizan el comportamiento de cualquier sistema de recomendación basado en el modelo propuesto de propagación de confiabilidad. El primer teorema establece que una premisa de confiabilidad explícita no puede ser revocada por el sistema. Más formalmente, establece que el conjunto de premisas de confiabilidad se incluyen en el conjunto de conclusiones de confiabilidad. El segundo teorema establece la consistencia o coherencia del sistema, lo que excluye la posibilidad de concluir que un usuario simultáneamente confía y no confía de la misma entidad.

Un ingrediente clave del modelo propuesto es la propagación de confiabilidad basada en mecanismos de inferencia. DeLP es el encargado de administrar la confiabilidad y la desconfianza interpersonales. La confiabilidad fue modelada mediante un conjunto de postulados básicos e intuitivos, que se pueden extender o relajar fácilmente. Los postulados se traducen en reglas rebatibles, que permiten inferir conclusiones provisionales sobre la confiabilidad de una noticia para un usuario. Esta propuesta difiere de trabajo previo en RS de noticias en permitir sacar conclusiones lógicas sobre la credibilidad de las noticias basándose en las opiniones de un grupo de usuarios.

Transparencia

Una debilidad central en la mayoría de los métodos actuales de recomendación es la falta de transparencia. La mayoría de los sistemas de recomendación en línea actúan como cajas negras, sin ofrecer al usuario información sobre la lógica del sistema o la justificación de las recomendaciones. Esto se debe al hecho de que los métodos cuantitativos adoptados por la mayoría de los sistemas de recomendación existentes no tienen un modelo subyacente limpio. Esto hace que sea difícil proporcionar a los usuarios una explicación clara de los factores y procedimientos que llevaron al sistema a realizar ciertas recomendaciones. En general, los usuarios prefieren las recomendaciones si pueden entender las razones por las que se presentan estas recomendaciones. Ciertamente, una recomendación es más convincente si el usuario conoce y está de acuerdo con las razones por las que se presentó la recomendación que si surgiera de una caja negra. Además, al ofrecer una justificación, el sistema está proporcionando información útil adicional a sus usuarios sobre temas relacionados o elementos que desconocen. Los enfoques cuantitativos de recomendación se basan típicamente en el uso de un número de parámetros, que hace que la interpretación de los resultados no sea intuitiva.

En el Capítulo 6 se presentó un modelo para sistemas de recomendación que fomenta la confianza del usuario en el RS ofreciendo transparencia en las recomendaciones. Para lograr este objetivo el RS complementa la recomendación junto con la justificación que la respalda. En esta propuesta, es posible expresar de una manera coloquial la razón detrás de cada recomendación. A partir del uso de estructuras coherentes de razones (argumentos) es posible generar explicaciones de los elementos recomendados. Estas explicaciones razonadas son útiles en la construcción de la confianza de los usuarios en el RS y permite la interacción favoreciendo la retroalimentación.

Se ha proporcionado un análisis comparativo entre el enfoque propuesto y algunas técnicas del estado del arte para la recomendación de películas. Se deduce de este análisis que el rendimiento del enfoque propuesto es comparable al de otros enfoques existentes. Aunque su rendimiento (precisión y/o cobertura) no es superior, no es sustancialmente inferior tampoco. Es importante destacar que las principales ventajas del enfoque propuesto con respecto a los métodos del estado del arte son otras características que aportan a la calidad de una recomendación, como lo son la transparencia (que promueve la confianza de los usuarios en el RS) y la flexibilidad (que facilita el modelado de diferentes criterios de una manera natural).

Por último, es importante mencionar que los experimentos realizados demostraron que el proceso de argumentación en el que se basan las recomendaciones puede hacer frente a grandes cantidades de datos. Debido a que el conjunto de datos utilizado fue *MovieLens 10M*, y a que el tiempo medio de respuesta de las consultas fue de unos dos segundos. Según nuestro conocimiento, esta es la primera vez que la programación lógica rebatible (más precisamente, su extensión habilitada para el uso de bases de datos) se prueba en un gran conjunto de datos del mundo real, proporcionando un precedente para su uso en sistemas expertos masivos.

Trabajo a futuro

Aunque se ha analizado un conjunto representativo de postulados para modelar la noción de confiabilidad, es una realidad que hay muchos otros postulados potencialmente útiles que un usuario podría querer agregar al sistema. Por ejemplo, para el caso particular de las noticias, un usuario podría querer incluir postulados que den preferencia a las noticias nuevas sobre las antiguas. Además un usuario podría no estar de acuerdo con el orden elegido para dar prioridad a los aspectos de recomendación y quisiera poder modificarlo. Todo esto es posible de hacer gracias a la flexibilidad de los modelos propuestos, pero en este punto, tales postulados tendrían que ser codificados a mano como reglas DeLP. Es por esto que podría ser de mucha utilidad contar con una herramienta que, a través de una interfaz adecuada, facilite la edición de postulados como reglas DeLP, permitiendo que el usuario los declare de manera intuitiva y que la herramienta los traduzca automáticamente a reglas DeLP.

Además, esta nueva herramienta podría integrarse con las API de distintas redes sociales, como las proporcionadas por *Facebook*, *Twitter* o *Instagram*. Mediante la recopilación de información social, como por ejemplo, amigos, seguidores, seguidos, etc., podría crearse una red de confiabilidad, interpretando esas relaciones entre usuarios y/o elementos de información como declaraciones de confiabilidad. Esta información luego podría ser utilizada para alimentar al modelo de administración de confiabilidad propuesto. Este mecanismo podría integrarse naturalmente con un sistema distribuido, para compartir no solo noticias sino también otros tipos de elementos de información. En este sentido, cada usuario podría contribuir y colaborar con comunidades específicas, así como con la red completa de usuarios.

Otra interesante vía de investigación futura consistirá en implementar mecanismos de aprendizaje automático que aprendan nuevas declaraciones de confiabilidad al monitorear el comportamiento del usuario. El modelo resultante combinará los beneficios de tener declaraciones explícitas proporcionadas por el usuario y nuevas declaraciones inferidas y actualizadas automáticamente a medida que el sistema aprende del comportamiento del usuario.

Otra cuestión interesante sería que la herramienta incorpore las principales características de los modelos aquí planteados, utilizando el modelo de administración de confiabilidad y lo combine con el modelo de recomendación mixto, mediante la creación de postulados adecuados para algún dominio específico. De esta manera sería posible comparar cómo se comportan juntos y por separado. La herramienta también podría aprovechar las explicaciones ofrecidas, permitiéndole al usuario corregir al RS cuando se equivoca, para así poder mejorar las futuras recomendaciones.

Otra dirección de investigación futura prometedora consiste en realizar un estudio de usuario para determinar la utilidad de las explicaciones presentadas por el sistema. Dado que estas explicaciones se basan en argumentos racionales, es decir están basadas en hechos y razones que se ajustan a las creencias de los usuarios, se puede suponer que las explicaciones dadas son adecuadas desde una perspectiva racional. Un estudio de usuario arrojará luz sobre temas interesantes como el papel de la racionalidad en la preferencia de ciertos elementos. En particular, podría ayudar a responder la pregunta esencial de si las personas se basan en argumentos racionales en el momento de seleccionar un elemento, como lo puede ser una película, una canción, etc., o si los argumentos irracionales o emocionales pueden tener una mayor influencia.

Como posible línea de investigación futura, se planea incorporar aspectos contextuales a los RS basados en argumentación propuestos en esta tesis, para averiguar si algunos de estos aspectos (por ejemplo, el estado de ánimo de los usuarios, el clima, la ubicación, etc.) juegan un papel importante en la recomendación. Un RS podría utilizar la información proporcionada por los dispositivos integrados en Internet para incorporar información contextual. Los sensores que comúnmente poseen estos dispositivos permitirían disponer de datos como por ejemplo, parámetros meteorológicos (temperatura, humedad y presión), ubicación geográfica y hasta señales de salud del usuario en tiempo real.

Finalmente, para profundizar más en este tema de los aspectos contextuales, se podría realizar una extensión del modelo aquí presentado para poder representar los cambios de contexto a través del tiempo, y así posiblemente, obtener diferentes resultados para la misma consulta cuando se formula en diferentes momentos.

Apéndice A

Simulaciones

En este apartado se muestran cada una de las matrices resultantes de las distintas simulaciones realizadas para validar el modelo de propagación de confiabilidad propuesto en el Capítulo 5. Se realizaron una serie de simulaciones en las cuales difieren la cantidad de usuarios, noticias, fuentes, tópicos y/o el grado de interacción entre las distintas entidades (relaciones de confiabilidad explícitas). Por cada simulación se muestran 2 matrices que contienen información sobre las relaciones de confiabilidad entre usuarios y noticias, una contiene las relaciones de confiabilidad explícitas y la otra contiene tanto las relaciones de confiabilidad explícitas como las inferidas por el modelo. En la Figura A.1 se muestra el porcentaje de relaciones de confiabilidad inferidas por el modelo durante las simulaciones.

Simulación	Entidades				Interacción	Relaciones inferidas	Indicis	
	#Usuarios	#Noticias	#Fuentes	#Tópicos			Antes	Después
Sim 1	5	4	2	2	5%	10%	95%	85%
Sim 2	5	4	2	2	25%	30%	75%	45%
Sim 3	5	4	2	2	40%	30%	60%	30%
Sim 4	10	9	3	3	5%	14%	95%	81%
Sim 5	10	9	3	3	25%	28%	75%	47%
Sim 6	15	12	4	4	5%	13%	95%	82%
Sim 7	15	12	4	4	25%	19%	75%	56%
Sim 8	20	50	5	5	5%	12%	95%	83%
Sim 9	20	50	5	5	10%	22%	90%	68%
Sim 10	20	50	5	5	50%	11%	50%	39%

Figura A.1: Información inferida por el sistema.

Simulación 1

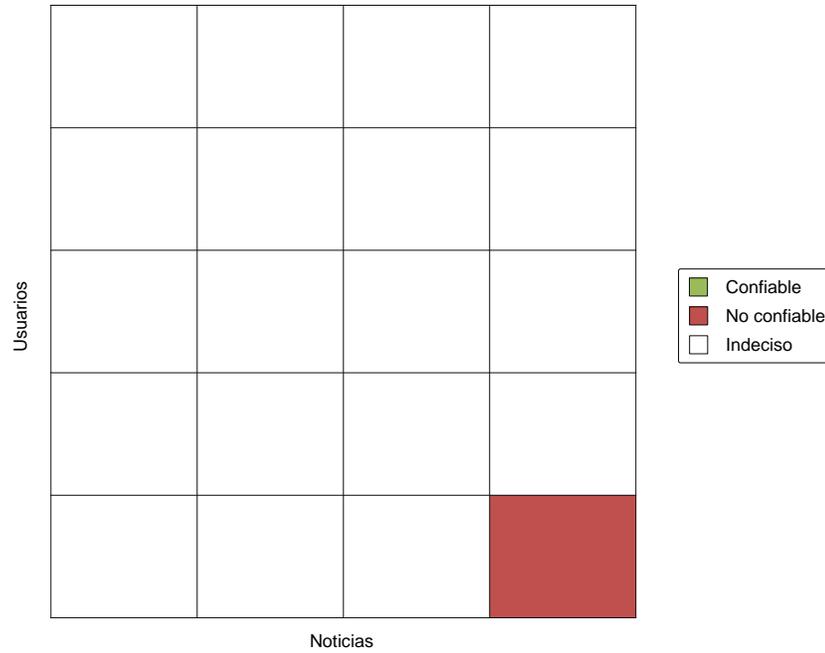


Figura A.2: Relaciones de confiabilidad explícita para la simulación 1.

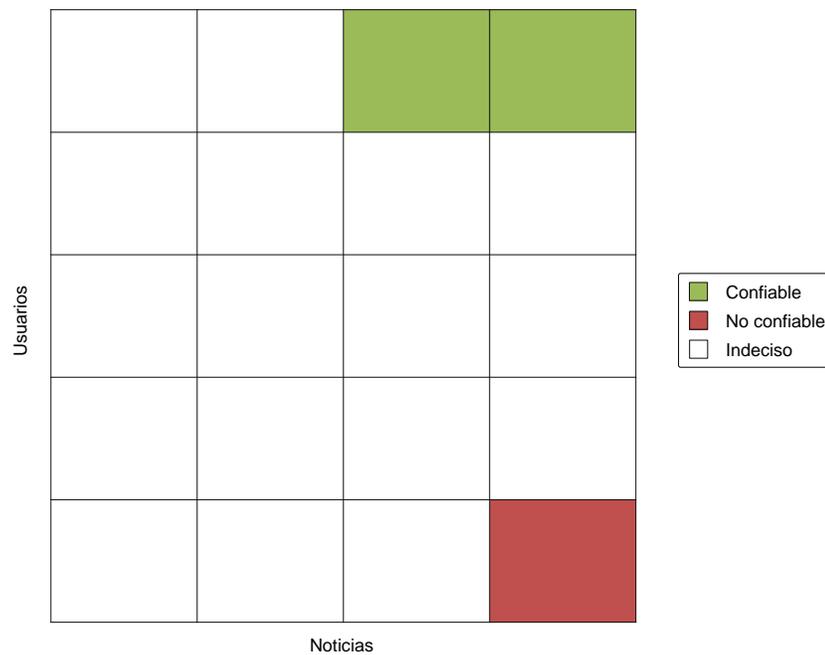


Figura A.3: Relaciones de confiabilidad explícita e inferida para la simulación 1.

Simulación 2

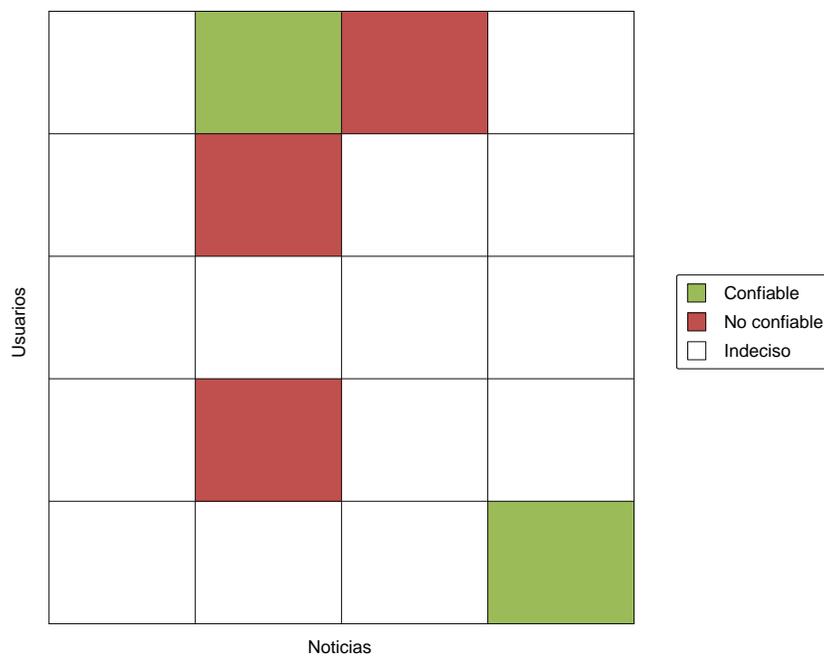


Figura A.4: Relaciones de confiabilidad explícita para la simulación 2.

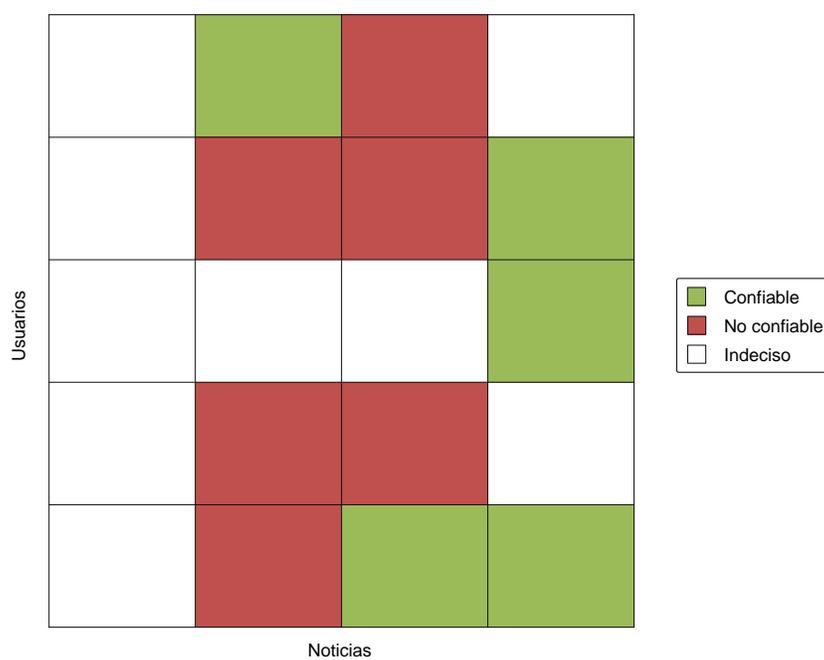


Figura A.5: Relaciones de confiabilidad explícita e inferida para la simulación 2.

Simulación 3

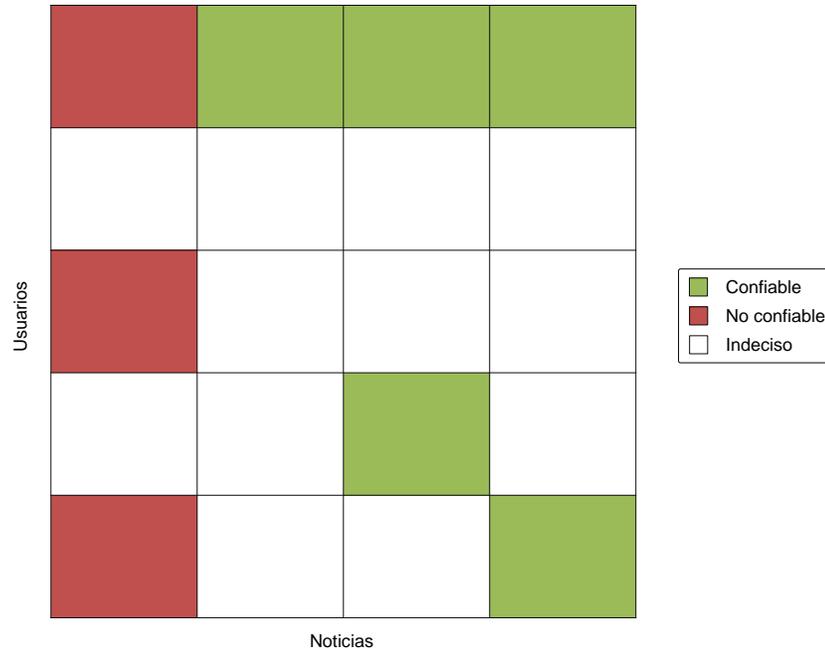


Figura A.6: Relaciones de confiabilidad explícita para la simulación 3.

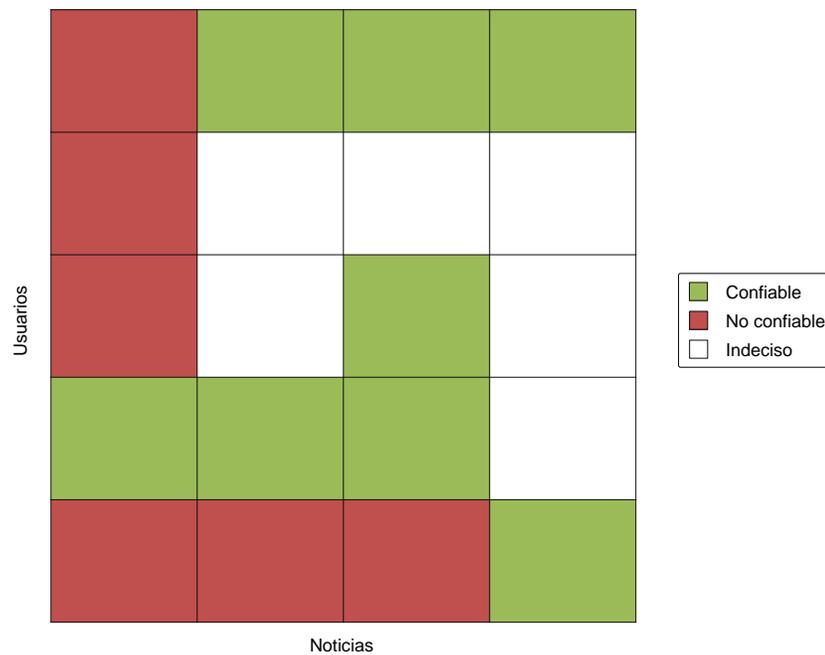


Figura A.7: Relaciones de confiabilidad explícita e inferida para la simulación 3.

Simulación 4

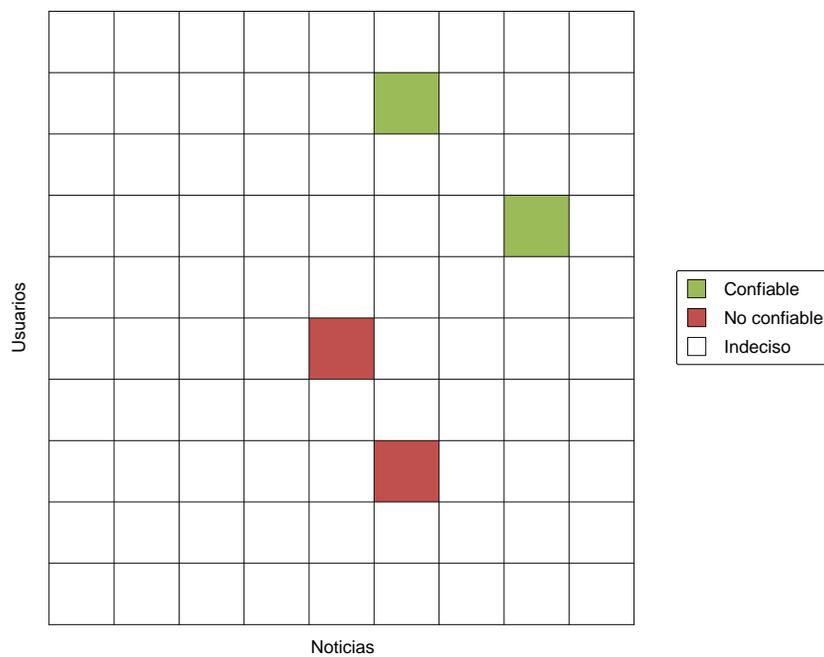


Figura A.8: Relaciones de confiabilidad explícita para la simulación 4.

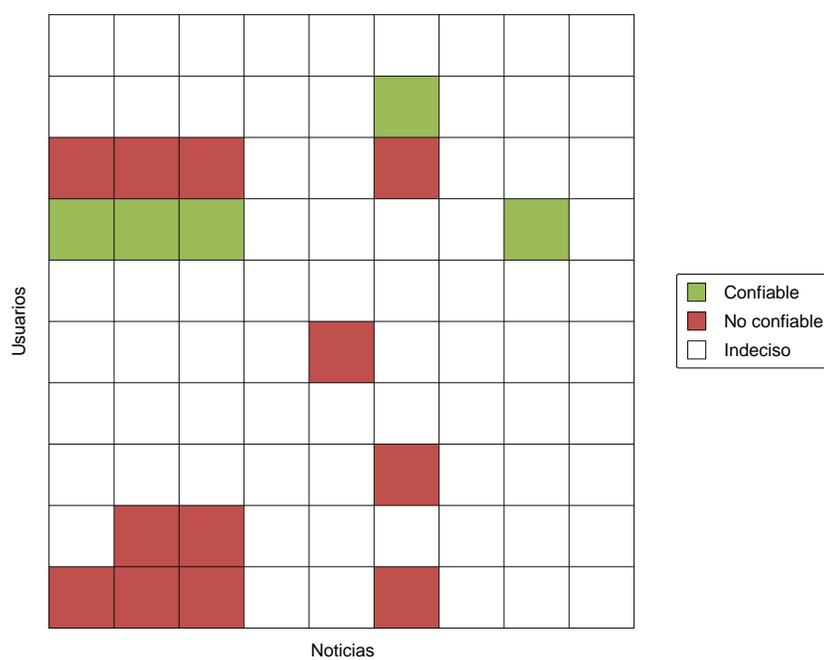


Figura A.9: Relaciones de confiabilidad explícita e inferida para la simulación 4.

Simulación 5

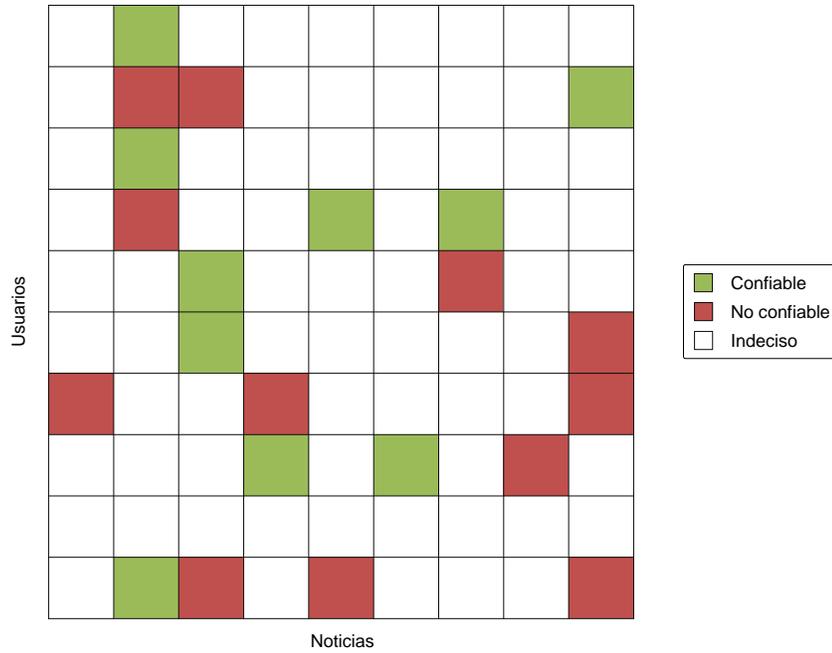


Figura A.10: Relaciones de confiabilidad explícita para la simulación 5.

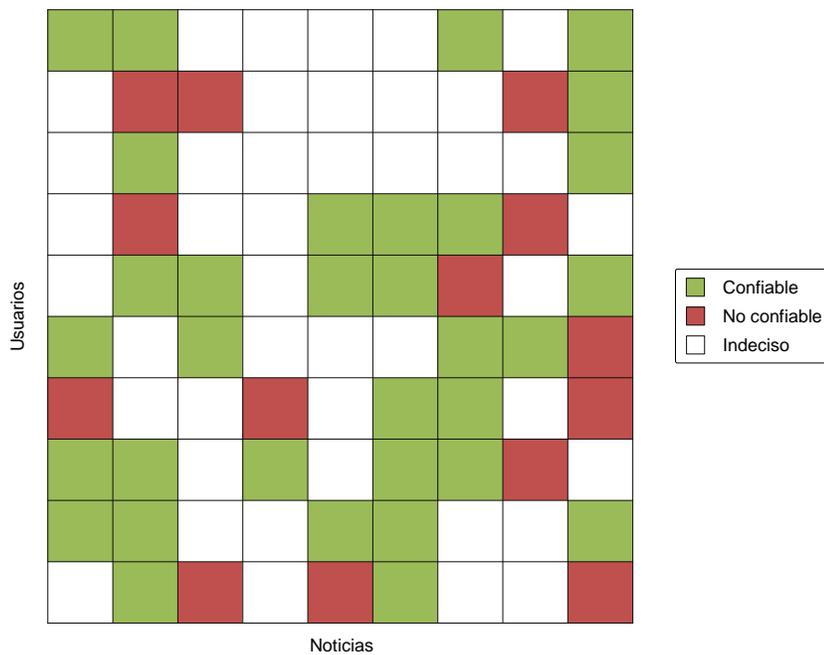


Figura A.11: Relaciones de confiabilidad explícita e inferida para la simulación 5.

Simulación 6

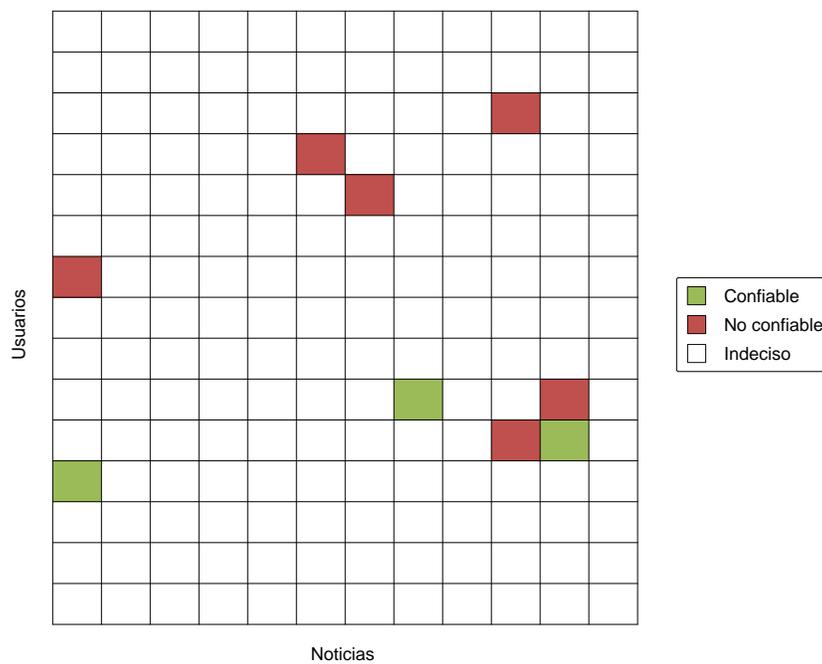


Figura A.12: Relaciones de confiabilidad explícita para la simulación 6.

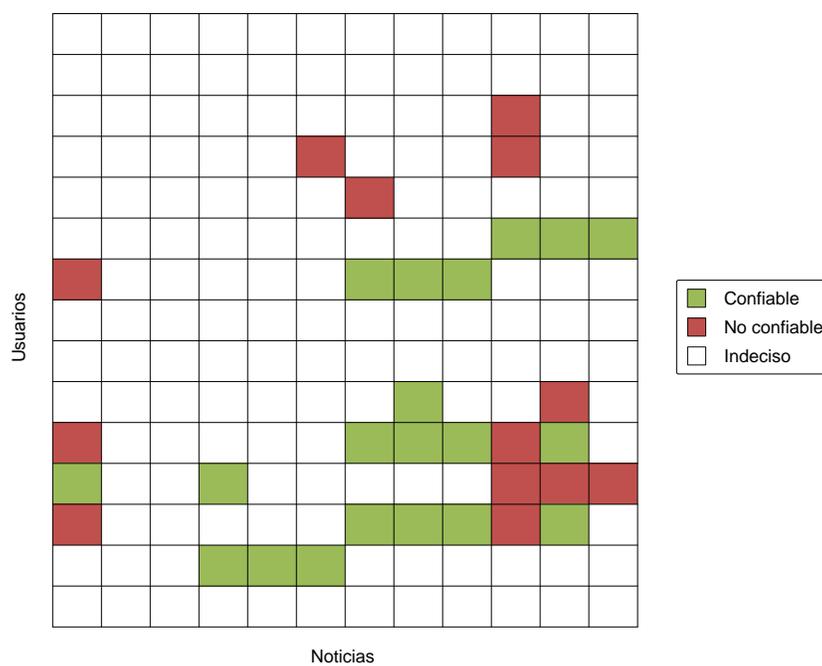


Figura A.13: Relaciones de confiabilidad explícita e inferida para la simulación 6.

Simulación 7

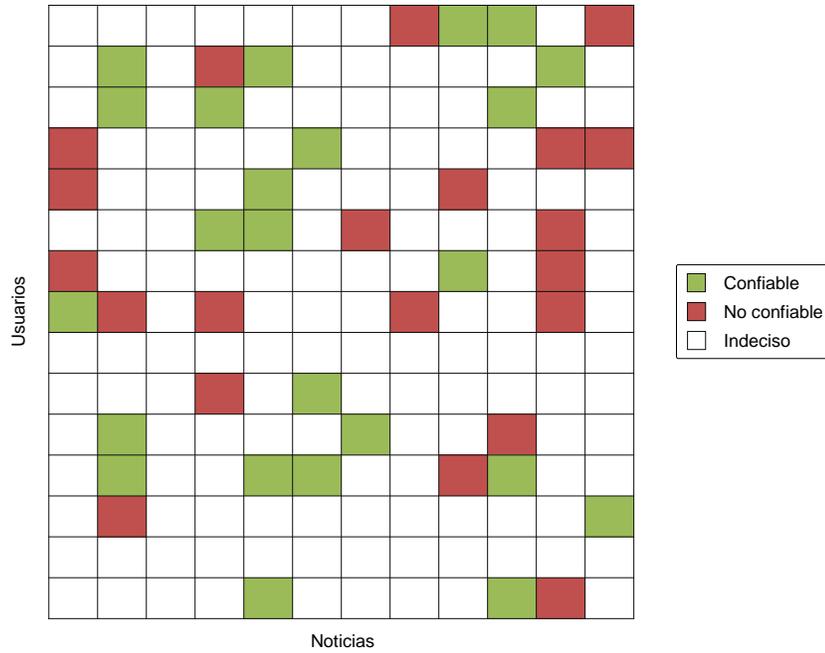


Figura A.14: Relaciones de confiabilidad explícita para la simulación 7.

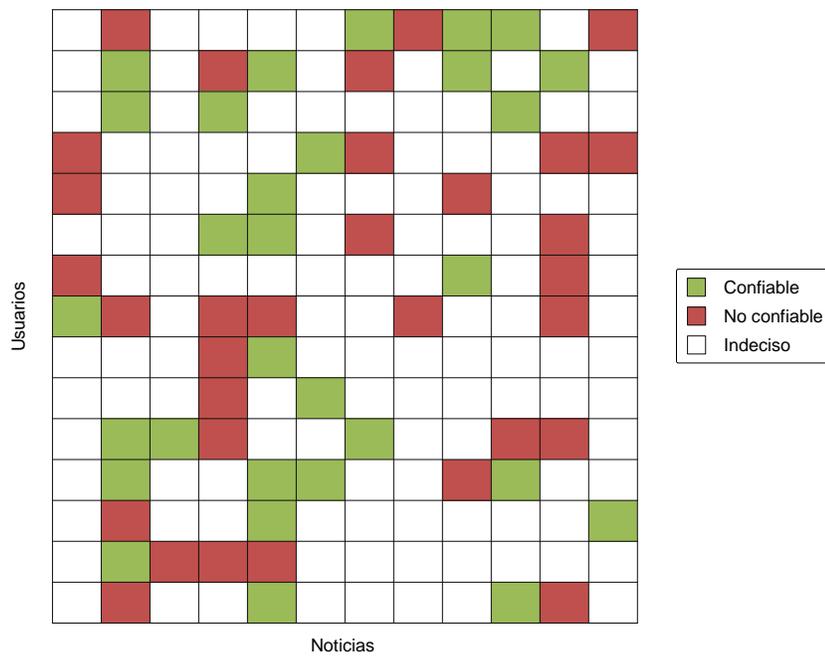


Figura A.15: Relaciones de confiabilidad explícita e inferida para la simulación 7.

Simulación 8

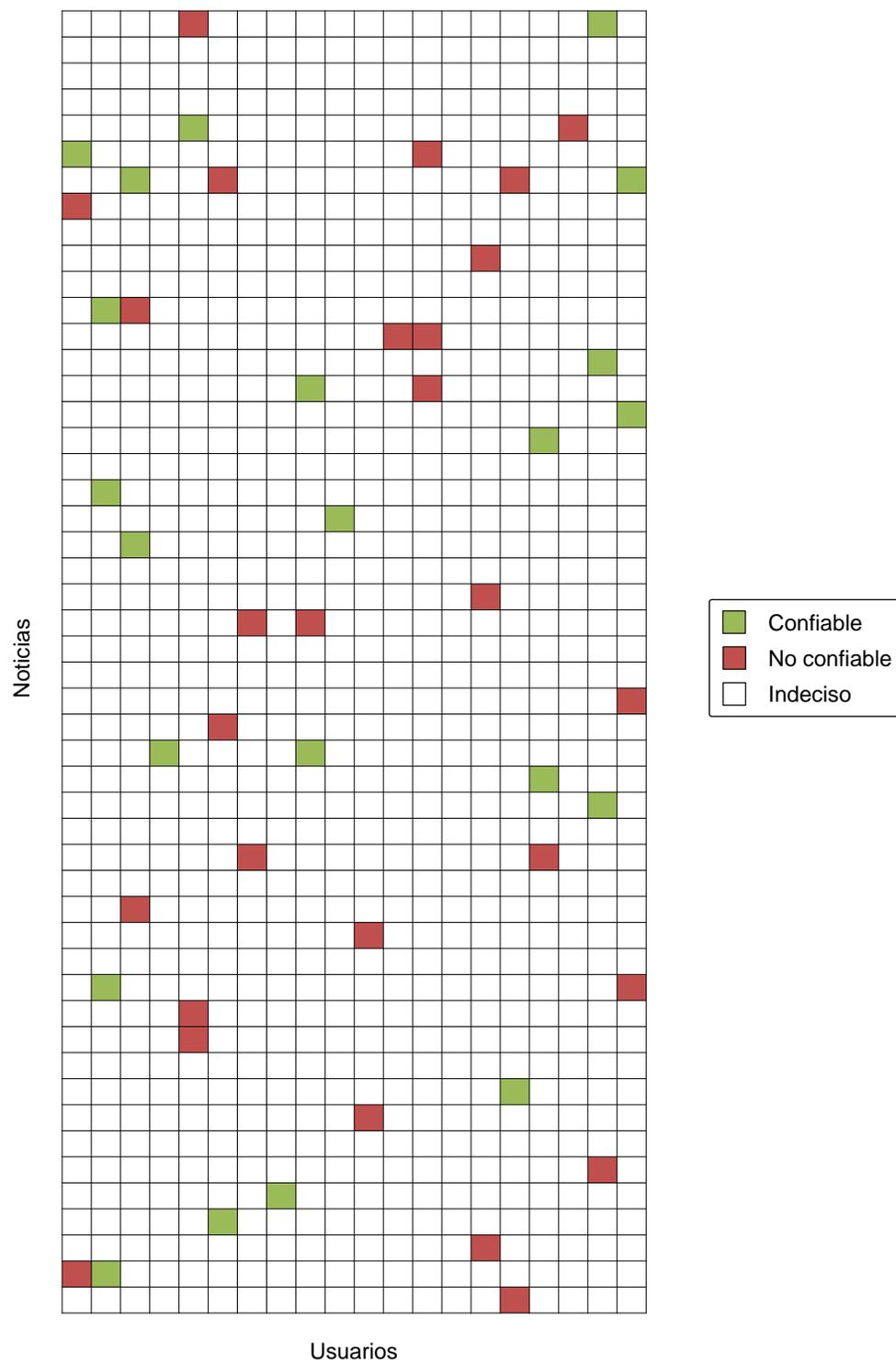


Figura A.16: Relaciones de confiabilidad explícita para la simulación 8.

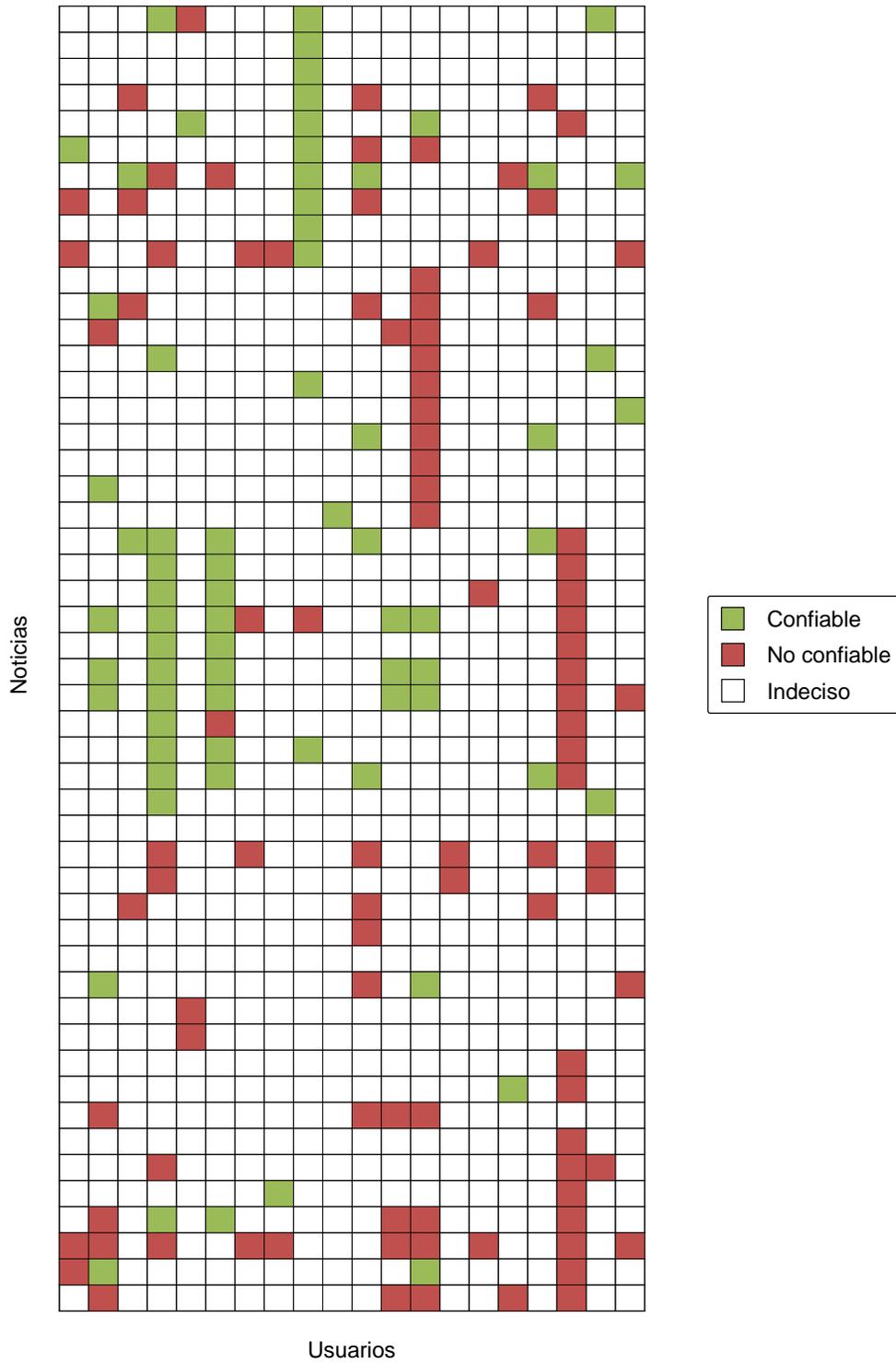


Figura A.17: Relaciones de confiabilidad explícita e inferida para la simulación 8.

Simulación 9

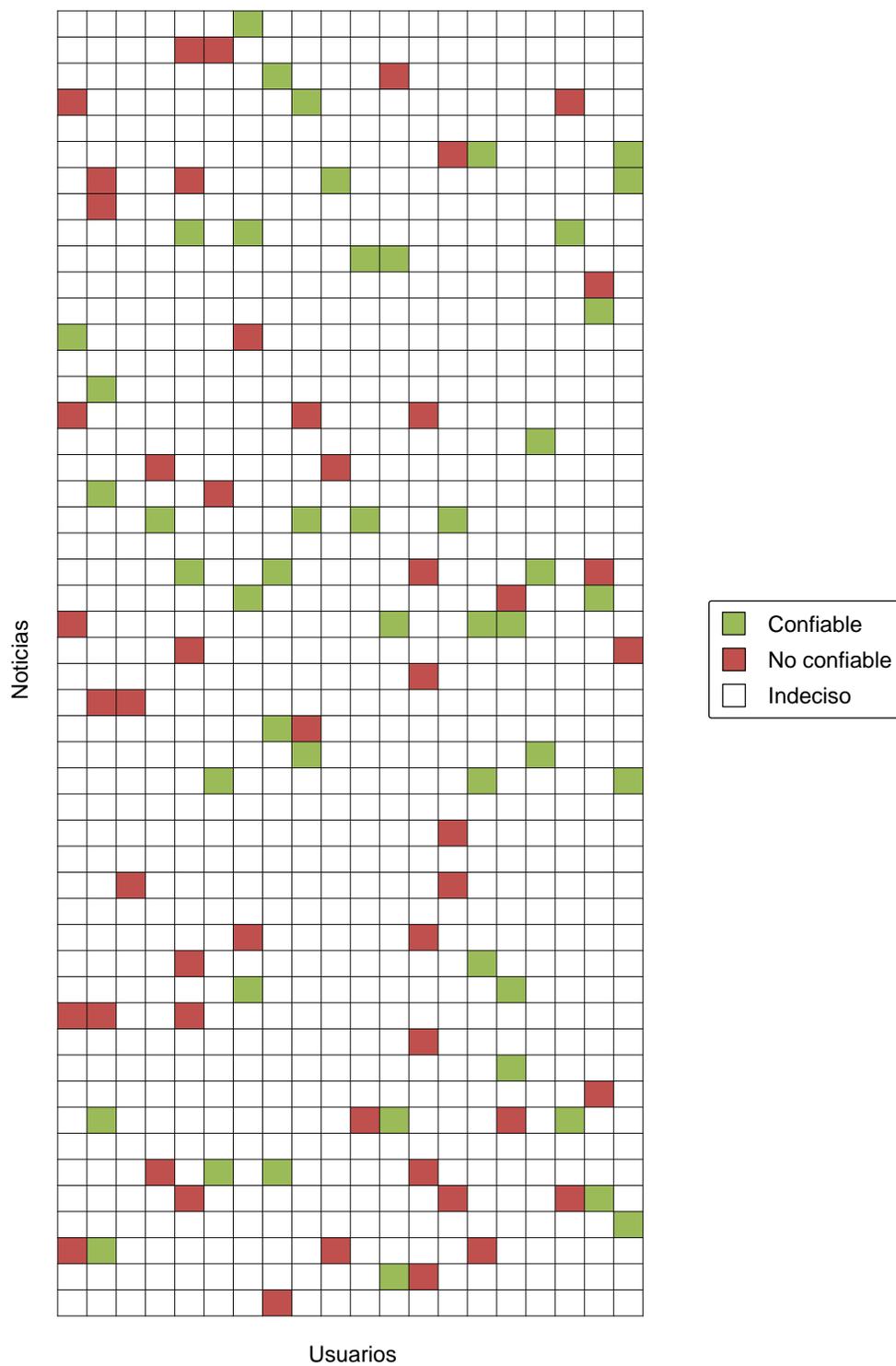


Figura A.18: Relaciones de confiabilidad explícita para la simulación 9.

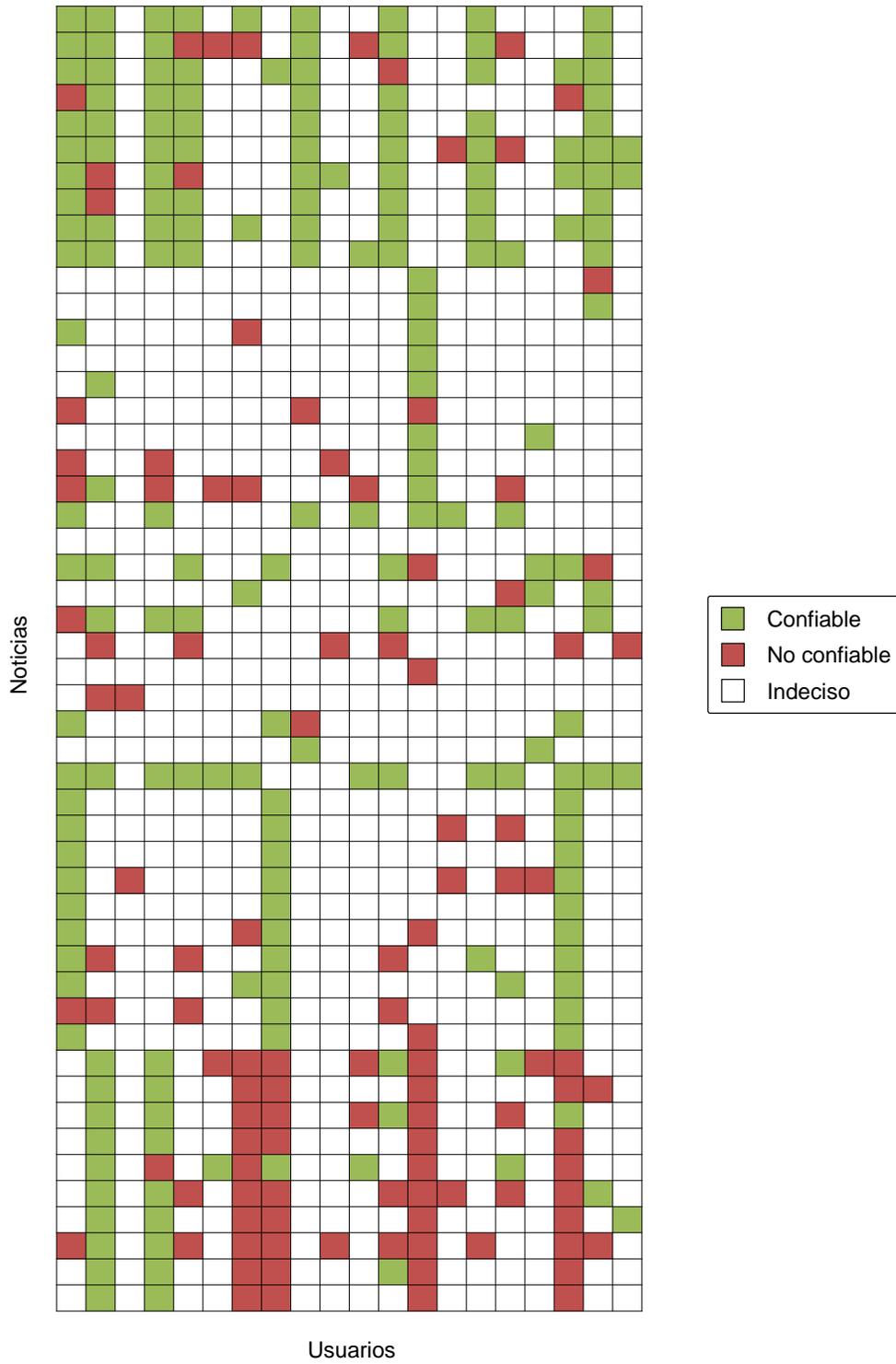


Figura A.19: Relaciones de confiabilidad explícita e inferida para la simulación 9.

Simulación 10

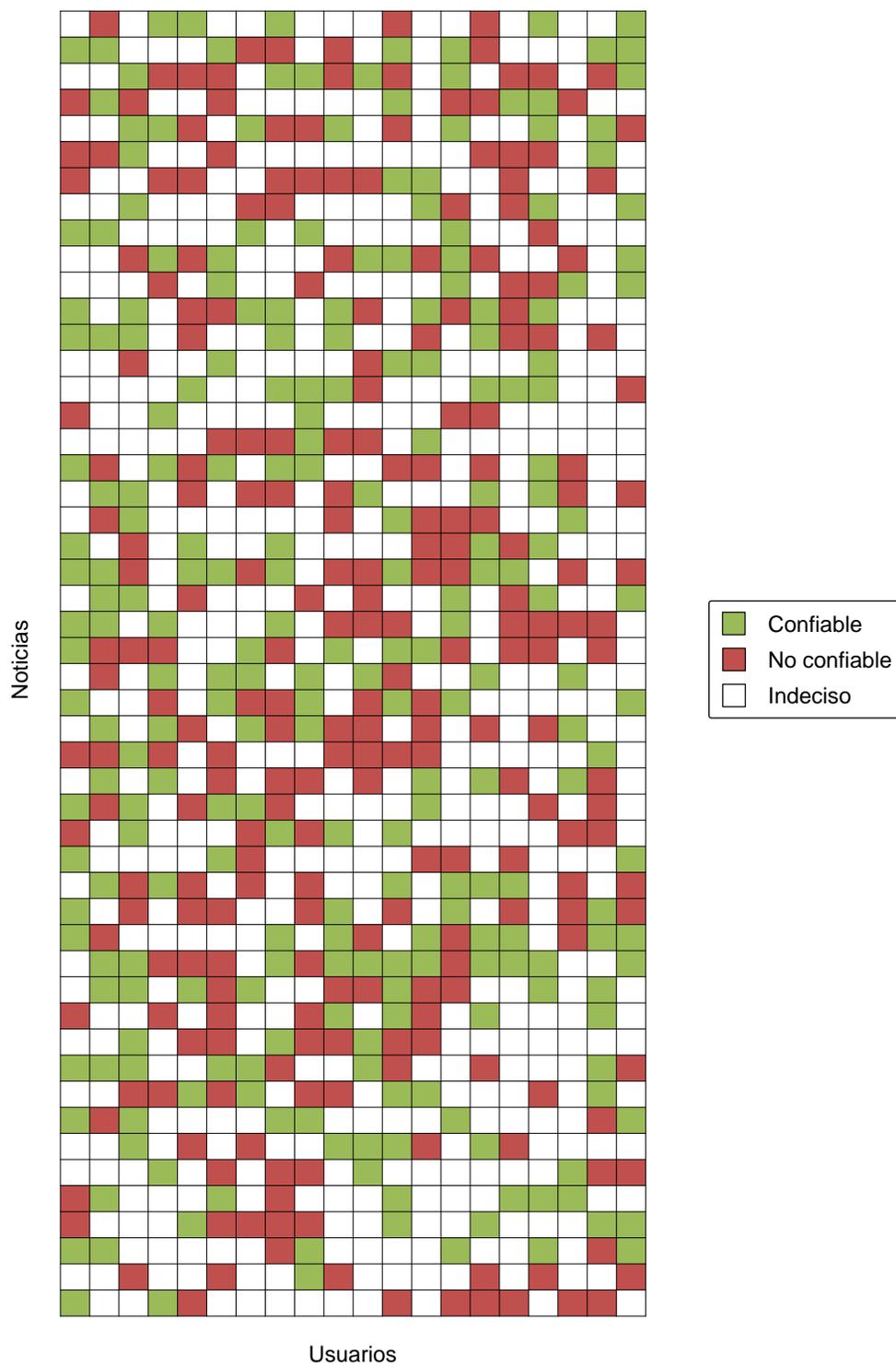


Figura A.20: Relaciones de confiabilidad explícita para la simulación 10.

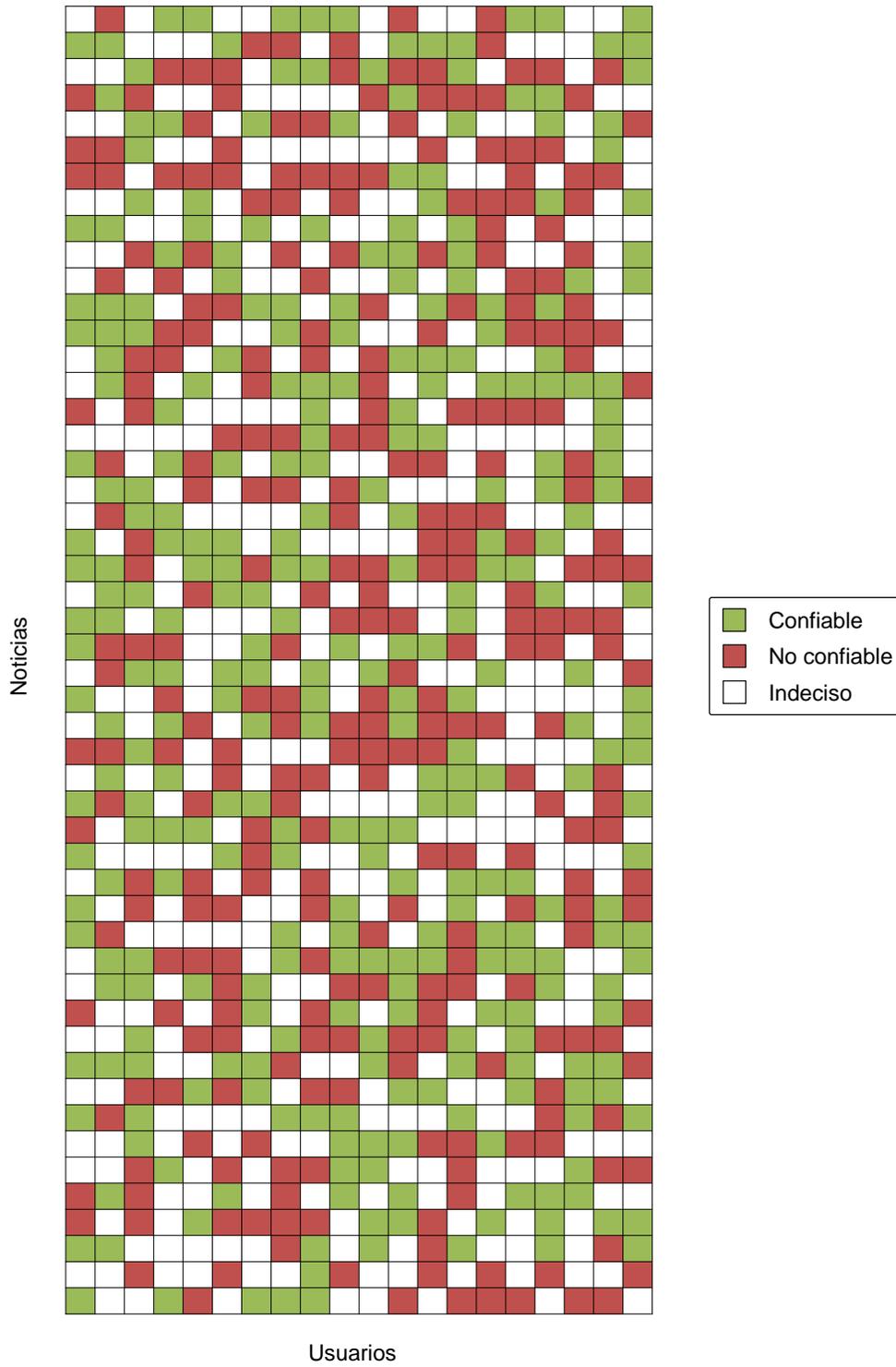


Figura A.21: Relaciones de confiabilidad explícita e inferida para la simulación 10.

Apéndice B

Glosario

B.1. Acrónimos

La siguiente tabla contiene los acrónimos utilizados en esta tesis, listados en orden alfabético.

Acrónimo	Expansión en español (<i>en inglés</i>)	Ubicación	Página
ANN	Red neuronal artificial (<i>Artificial Neural Network</i>)	Capítulo 2	31
CB	Basado en contenido (<i>Content-Based</i>)	Capítulo 1	3
CF	Filtrado colaborativo (<i>Collaborative Filtering</i>)	Capítulo 1	3
CI	Intervalo de confianza (<i>Confidence Interval</i>)	Capítulo 6	167
DBI-DeLP	Bases de datos integradas para la programación en lógica rebatible (<i>Database Integration for Defeasible Logic Programming</i>)	Capítulo 3	90
DeLP	Programación en lógica rebatible (<i>Defeasible Logic Programming</i>)	Capítulo 1	8
FN	Falso negativo (<i>False Negative</i>)	Capítulo 2	50

La tabla continúa de una página previa

Acrónimo	Expansión en español (en inglés)	Ubicación	Página
FOAF	Amigo de un amigo (<i>Friend Of A Friend</i>)	Capítulo 5	123
FP	Falso positivo (<i>False Positive</i>)	Capítulo 2	50
HMM	Modelo oculto de Markov (<i>Hidden Markov Model</i>)	Capítulo 4	105
IDF	Frecuencia inversa de documento (<i>Inverse Document Frequency</i>)	Capítulo 2	29
IR	Recuperación de información (<i>Information Retrieval</i>)	Capítulo 2	29
LSI	Indexación semántica latente (<i>Latent Semantic Indexing</i>)	Capítulo 2	43
ML	Aprendizaje automático (<i>Machine Learning</i>)	Capítulo 2	31
MAE	Error absoluto medio (<i>Mean Absolute Error</i>)	Capítulo 2	49
MEAN	Valor medio (<i>Mean</i>)	Capítulo 6	167
RMSE	Raíz del error cuadrático medio (<i>Root Mean Squared Error</i>)	Capítulo 2	49
RS	Sistemas de recomendación (<i>Recommender Systems</i>)	Capítulo 2	13
STDEV	Desviación estándar (<i>Standard Deviation</i>)	Capítulo 6	167
SVD	Descomposición en valores singulares (<i>Singular Value Decomposition</i>)	Capítulo 2	39
TF	Frecuencia de término (<i>Term Frequency</i>)	Capítulo 2	29
TN	Verdadero negativo (<i>True Negative</i>)	Capítulo 2	50

La tabla continúa de una página previa

Acrónimo	Expansión en español (en inglés)	Ubicación	Página
TP	Verdadero positivo (<i>True Positive</i>)	Capítulo 2	50
TREC	Conferencia de recuperación de texto (<i>Text Retrieval Conference</i>)	Capítulo 2	32

B.2. Terminología

La siguiente tabla contiene los términos utilizados en esta tesis junto a su traducción, listados en orden alfabético.

Término en inglés	Traducción utilizada	Ubicación	Página
Accuracy	Exactitud	Capítulo 2	50
Adaptive filtering	Filtrado adaptativo	Capítulo 2	32
Assumption attack	Ataque a una suposición	Capítulo 3	57
Cautious monotonicity	Monotonicidad cautelosa	Capítulo 5	125
Cluster	Agrupamiento	Capítulo 2	38
Cut	Corte	Capítulo 5	125
Description logics	Lógicas descriptivas	Capítulo 6	133
F-Score	F-Calificación	Capítulo 2	50
Planning	Planificación	Capítulo 6	134
Precision	Precisión	Capítulo 2	50
Rebutting attack	Ataque por refutación	Capítulo 3	57
Recall	Cobertura	Capítulo 2	50
Reinstatement	Reinstalación	Capítulo 3	59
Strong negation	Negación fuerte	Capítulo 3	65
Text retrieval	Recuperación de texto	Capítulo 2	32
Threshold setting	Configuración del umbral	Capítulo 2	32
Undermining attack	Ataque a una premisa	Capítulo 3	58

Bibliografía

- [ABM05] ATKINSON, K., BENCH-CAPON, T. J. M., AND MCBURNEY, P. Multi-agent argumentation for eDemocracy. In *EUMAS 2005 - Proceedings of the Third European Workshop on Multi-Agent Systems, Brussels, Belgium, December 7-8, 2005* (2005), pp. 35–46.
- [AC02] AMGOUD, L., AND CAYROL, C. Inferring from inconsistency in preference-based argumentation frameworks. *Journal of Automated Reasoning* 29, 2 (2002), 125–169.
- [ADMF⁺15] AGRESTE, S., DE MEO, P., FERRARA, E., PICCOLO, S., AND PROVETTI, A. Trust networks: Topology, dynamics, and measurements. *IEEE Internet Computing* 19, 6 (2015), 26–35.
- [AEG⁺10] ADALI, S., ESCRIVA, R., GOLDBERG, M. K., HAYVANOVYCH, M., MAGDON-ISMAIL, M., SZYMANSKI, B. K., WALLACE, W. A., AND WILLIAMS, G. Measuring behavioral trust in social networks. In *2010 IEEE International Conference on Intelligence and Security Informatics* (2010), IEEE, pp. 150–152.
- [Agg16] AGGARWAL, C. C. Content-based recommender systems. In *Recommender Systems*. Springer, 2016, pp. 139–166.
- [AHT10] AHAMED, S. I., HAQUE, M. M., AND TALUKDER, N. A formal context specific trust model (FTM) for multimedia and ubiquitous computing environment. *Telecommunication Systems* 44, 3-4 (2010), 221–240.
- [AK07] AMGOUD, L., AND KACI, S. An argumentation framework for merging conflicting knowledge bases. *International Journal of Approximate Reasoning* 45, 2 (2007), 321–340.

- [Alv02] ALVAREZ, S. An exact analytical relation among recall, precision, and classification accuracy in information retrieval. *Technical Report BCCS-02-01* (01 2002), 1–22.
- [Amg03] AMGOUD, L. A formal framework for handling conflicting desires. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty* (2003), Springer, pp. 552–563.
- [AMP02] AMGOUD, L., MAUDET, N., AND PARSONS, S. An argumentation-based semantics for agent communication languages. In *Proceedings of the 15th European Conference on Artificial Intelligence, ECAI'2002, Lyon, France, July 2002* (2002), pp. 38–42.
- [AP93a] ALFERES, J. J., AND PEREIRA, L. M. Contradiction: When avoidance equals removal - Part I. In *Extensions of Logic Programming, 4th International Workshop, ELP'93, St. Andrews, UK, March 29 - April 1, 1993, Proceedings* (1993), pp. 11–23.
- [AP93b] ALFERES, J. J., AND PEREIRA, L. M. Contradiction: When avoidance equals removal - Part II. In *Extensions of Logic Programming, 4th International Workshop, ELP'93, St. Andrews, UK, March 29 - April 1, 1993, Proceedings* (1993), pp. 268–281.
- [AP09] AMGOUD, L., AND PRADE, H. Using arguments for making and explaining decisions. *Artificial Intelligence* 173, 3-4 (2009), 413–436.
- [APM00] AMGOUD, L., PARSONS, S., AND MAUDET, N. Arguments, dialogue, and negotiation. In *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, August 20-25, 2000* (2000), pp. 338–342.
- [Arg91] ARGYLE, M. *Cooperation, the basis of sociability*. Problems in Practice Series. Routledge, 1991.
- [ARH00] ABDUL-RAHMAN, A., AND HAILES, S. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000*. (2000), IEEE, pp. 1–9.

- [ASA⁺10] ALMAZRO, D., SHAHATAH, G., ALBDULKARIM, L., KHEREES, M., MARTINEZ, R., AND NZOUKOU, W. A survey paper on recommender systems. *CoRR abs/1006.5278* (2010).
- [ASST05] ADOMAVICIUS, G., SANKARANARAYANAN, R., SEN, S., AND TUZHILIN, A. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems* 23, 1 (January 2005), 103–145.
- [AT05] ADOMAVICIUS, G., AND TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* 17, 6 (June 2005), 734–749.
- [AV12] AMGOUD, L., AND VESIC, S. A formal analysis of the role of argumentation in negotiation dialogues. *Journal of Logic and Computation* 22, 5 (2012), 957–978.
- [AWBT18] ADELMEYER, M., WALTERBUSCH, M., BIERMANSKI, P., AND TEUTEBERG, F. Trust transitivity and trust propagation in cloud computing ecosystems. In *Proceedings of the 26th European Conference on Information Systems. Portsmouth, UK* (2018).
- [AWWY99] AGGARWAL, C. C., WOLF, J. L., WU, K.-L., AND YU, P. S. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (1999), ACM, pp. 201–212.
- [AyC14] *Argument & Computation - Special Issue: Tutorials on structured argumentation* (2014), vol. 5, Taylor & Francis Online.
- [AZ12] ADOMAVICIUS, G., AND ZHANG, J. Stability of recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 30, 4 (2012), 23.
- [BA10] BIKAKIS, A., AND ANTONIOU, G. Defeasible contextual reasoning with arguments in ambient intelligence. *IEEE Transactions on Knowledge and Data Engineering* 22, 11 (2010), 1492–1506.

- [Bai86] BAIER, A. Trust and antitrust. *Ethics* 96, 2 (1986), 231–260.
- [Bar83] BARBER, B. *The logic and limits of trust*. Rutgers University Press, 1983.
- [Bat00] BATESON, P. The biological evolution of cooperation and trust. *Trust: Making and Breaking Cooperative Relations* (2000), 14–30.
- [BBD⁺12] BRIGUEZ, C. E., BUDÁN, M. C., DEAGUSTINI, C. A. D., MAGUITMAN, A. G., CAPOBIANCO, M., AND SIMARI, G. R. Towards an argument-based music recommender system. In *Fourth International Conference on Computational Models of Argument (COMMA 2012)* (2012), B. Verheij, S. Szeider, and S. Woltran, Eds., vol. 245 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, pp. 83–90.
- [BBD⁺14] BRIGUEZ, C. E., BUDÁN, M. C., DEAGUSTINI, C. A. D., MAGUITMAN, A. G., CAPOBIANCO, M., AND SIMARI, G. R. Argument-based mixed recommenders and their application to movie suggestion. *Expert Systems with Applications* 41, 14 (2014), 6467–6482.
- [BC92] BELKIN, N. J., AND CROFT, W. B. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM* 35, 12 (1992), 29–38.
- [BC08] BÉLANGER, F., AND CARTER, L. Trust and risk in e-government adoption. *The Journal of Strategic Information Systems* 17, 2 (2008), 165–176.
- [BCJ⁺18] BEUTEL, A., COVINGTON, P., JAIN, S., XU, C., LI, J., GATTO, V., AND CHI, E. H. Latent cross: Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (2018), ACM, pp. 46–54.
- [BCM⁺03] BAADER, F., CALVANESE, D., MCGUINNESS, D., PATELSCHNEIDER, P., AND NARDI, D. *The description logic handbook: Theory, implementation and applications*. Cambridge University Press, 2003.

- [BCM13] BRIGUEZ, C. E., CAPOBIANCO, M., AND MAGUITMAN, A. G. A theoretical framework for trust-based news recommender systems and its implementation using defeasible argumentation. *International Journal on Artificial Intelligence Tools* 22, 4 (2013).
- [BCS98] BREWER, W. F., CHINN, C. A., AND SAMARAPUNGAN, A. Explanation in scientists and children. *Minds and Machines* 8, 1 (Feb 1998), 119–136.
- [BD07] BENCH-CAPON, T. J. M., AND DUNNE, P. E. Argumentation in artificial intelligence. *Artificial Intelligence* 171, 10-15 (2007), 619–641.
- [Ben11] BENEDICKTUS, R. L. The effects of 3rd party consensus information on service expectations and online trust. *Journal of Business Research* 64, 8 (2011), 846 – 853.
- [BG94] BUCKLAND, M. K., AND GEY, F. C. The relationship between recall and precision. *Journal of the American Society for Information Science (JASIS)* 45, 1 (1994), 12–19.
- [BGMS05] BRIDGE, D., GÖKER, M. H., MCGINTY, L., AND SMYTH, B. Case-based recommender systems. *The Knowledge Engineering Review* 20, 3 (2005), 315–320.
- [BGOZ18] BOBADILLA, J., GUTIÉRREZ, A., ORTEGA, F., AND ZHU, B. Reliability quality measures for recommender systems. *Information Sciences* 442 (2018), 145–157.
- [BH01] BESNARD, P., AND HUNTER, A. A logic-based theory of deductive arguments. *Artificial Intelligence* 128, 1-2 (2001), 203–235.
- [BH08] BESNARD, P., AND HUNTER, A. *Elements of argumentation*. MIT Press, 2008.
- [BH09] BESNARD, P., AND HUNTER, A. Argumentation based on classical logic. In *Argumentation in Artificial Intelligence*. Springer, 2009, pp. 133–152.

- [BHC98] BASU, C., HIRSH, H., AND COHEN, W. W. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, July 26-30, 1998, Madison, Wisconsin, USA.* (1998), pp. 714–720.
- [BHK98] BREESE, J. S., HECKERMAN, D., AND KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (1998), Morgan Kaufmann Publishers Inc., pp. 43–52.
- [BHM17] BAGHER, R. C., HASSANPOUR, H., AND MASHAYEKHI, H. User trends modeling for a content-based recommender system. *Expert Systems with Applications* 87 (2017), 209–219.
- [BHOB11] BOBADILLA, J., HERNANDO, A., ORTEGA, F., AND BERNAL, J. A framework for collaborative filtering recommender systems. *Expert Systems with Applications* 38, 12 (2011), 14609–14623.
- [BK06] BEDI, P., AND KAUR, H. Trust based personalized recommender system. *INFOCOMP* 5, 1 (2006), 19–26.
- [BKB⁺06] BANOS, E., KATAKIS, I., BASSILIADES, N., TSOUMAKAS, G., AND VLAHAVAS, I. P. PersoNews: A personalized news reader enhanced by machine learning and semantic filtering. In *On the Move to Meaningful Internet Systems 2006* (2006), pp. 975–982.
- [BKR08] BERKOVSKY, S., KUFLIK, T., AND RICCI, F. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction* 18, 3 (August 2008), 245–286.
- [BKR09] BERKOVSKY, S., KUFLIK, T., AND RICCI, F. Cross-representation mediation of user models. *User Modeling and User-Adapted Interaction* 19, 1 (Feb 2009), 35–63.

- [BL16] BENOURET, I., AND LENNE, D. A package recommendation framework for trip planning activities. In *Proceedings of the 10th ACM Conference on Recommender Systems* (2016), ACM, pp. 203–206.
- [BLCS12] BUDÁN, M. C., LUCERO, M. J. G., CHESÑEVAR, C. I., AND SIMARRI, G. R. Modelling time and reliability in structured argumentation frameworks. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012* (2012), G. Brewka, T. Eiter, and S. A. McIlraith, Eds., AAAI Press.
- [BLSR13] BEX, F., LAWRENCE, J., SNAITH, M., AND REED, C. Implementing the argument web. *Communications of the ACM* 56, 10 (2013), 66–73.
- [BM05] BILGIC, M., AND MOONEY, R. J. Explaining recommendations: Satisfaction vs. promotion. In *Beyond Personalization Workshop, IUI* (2005), vol. 5.
- [BMCMB⁺10] BARRAGÁNS-MARTÍNEZ, A. B., COSTA-MONTENEGRO, E., BURGUILLO, J. C., REY-LÓPEZ, M., MIKIC-FONTE, F. A., AND PELETEIRO, A. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences* 180, 22 (2010), 4290–4311.
- [BOHG13] BOBADILLA, J., ORTEGA, F., HERNANDO, A., AND GUTIÉRREZ, A. Recommender systems survey. *Knowledge-Based Systems* 46 (July 2013), 109–132.
- [Bok99] BOK, S. *Lying: Moral choice in public and private life*. Vintage, 1999.
- [BOvS09] BAKKER, A., OGSTON, E., AND VAN STEEN, M. Collaborative filtering using random neighbours in peer-to-peer networks. In *Proceeding of the ACM First International Workshop on Complex Networks Meet Information & Knowledge Management, CIKM-CNIKM 2009, Hong Kong, China, November 6, 2009* (2009), pp. 67–75.

- [BP98a] BILLSUS, D., AND PAZZANI, M. J. Learning collaborative information filters. In *International Conference on Machine Learning (1998)*, vol. 98, pp. 46–54.
- [BP98b] BRIN, S., AND PAGE, L. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30*, 1–7 (1998), 107–117.
- [BP00] BILLSUS, D., AND PAZZANI, M. J. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction 10*, 2-3 (2000), 147–180.
- [BP06] BACCIGALUPO, C., AND PLAZA, E. Case-based sequential ordering of songs for playlist recommendation. In *Advances in Case-Based Reasoning, 8th European Conference, ECCBR 2006, Fethiye, Turkey, September 4-7, 2006, Proceedings (2006)*, pp. 286–300.
- [BP12] BELLOGÍN, A., AND PARAPAR, J. Using graph partitioning techniques for neighbour selection in user-based collaborative filtering. In *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012 (2012)*, pp. 213–216.
- [BR99] BAEZA-YATES, R. A., AND RIBEIRO-NETO, B. A. *Modern information retrieval*. ACM Press / Addison-Wesley, 1999.
- [Bro88] BROWN, R. *Group processes: Dynamics within and between groups*. Basil Blackwell, 1988.
- [BS97] BALABANOVIĆ, M., AND SHOHAM, Y. Fab: Content-based, collaborative recommendation. *Communications of the ACM 40*, 3 (1997), 66–72.
- [BSCM11] BRIGUEZ, C. E., SAGUI, F. M., CAPOBIANCO, M., AND MAGUITMAN, A. G. System architecture for trust-based news recommenders on the web. In *XVII Workshop de Agentes y Sistemas Inteligentes - CA-CIC 2011: XVII Congreso Argentino de Ciencias de la Computación (La Plata, Buenos Aires, Argentina, Octubre 2011)*.
- [BTC17] BERKOVSKY, S., TAIB, R., AND CONWAY, D. How to recommend?: User trust factors in movie recommender systems. In *Proceedings of the*

- 22nd International Conference on Intelligent User Interfaces* (2017), ACM, pp. 287–300.
- [BTG19] BRARDA, M. E. B., TAMARGO, L. H., AND GARCÍA, A. J. An approach to enhance argument-based multi-criteria decision systems with conditional preferences and explainable answers. *Expert Systems with Applications* 126 (2019), 171–186.
- [BTR⁺07] BEN-SHIMON, D., TSIKINOVSKY, A., ROKACH, L., MEISELS, A., SHANI, G., AND NAAMANI, L. Recommender system from personal social networks. In *Advances in Intelligent Web Mastering, Proceedings of the 5th Atlantic Web Intelligence Conference - AWIC 2007, Fontainebleau, France, June 25 - 27, 2007* (2007), pp. 47–55.
- [Bur00] BURKE, R. Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*, vol. 69. Marcel Dekker, New York, NY, USA, 2000, pp. 180–200.
- [Bur02] BURKE, R. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12 (November 2002), 331–370.
- [Bur07] BURKE, R. D. Hybrid web recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization* (2007), pp. 377–408.
- [BV11] BEDI, P., AND VASHISTH, P. A fuzzy trust model for argumentation-based recommender systems. In *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011* (2011), Springer, pp. 493–502.
- [BV14] BEDI, P., AND VASHISTH, P. Empowering recommender systems using trust and argumentation. *Information Sciences* 279 (2014), 569–586.
- [BV15] BEDI, P., AND VASHISTH, P. Argumentation-enabled interest-based personalised recommender system. *Journal of Experimental & Theoretical Artificial Intelligence* 27, 2 (2015), 199–226.

- [CA07] CAMINADA, M., AND AMGOUD, L. On the evaluation of argumentation formalisms. *Artificial Intelligence* 171, 5–6 (2007), 286–310.
- [Cap03] CAPOBIANCO, M. *Argumentación rebatible en entornos dinámicos*. PhD thesis, Universidad Nacional del Sur, Bahía Blanca, Argentina, 2003.
- [CCFF11] CACHEDA, F., CARNEIRO, V., FERNÁNDEZ, D., AND FORMOSO, V. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web* 5, 1 (2011).
- [CCS05] CAPOBIANCO, M., CHESÑEVAR, C. I., AND SIMARI, G. R. Argumentation and the dynamics of warranted beliefs in changing environments. *Autonomous Agents and Multi-Agent Systems* 11, 2 (2005), 127–151.
- [CDE08] CZENKO, M., DOUMEN, J., AND ETALLE, S. Trust management in P2P systems using standard TuLiP. In *Trust Management II - Proceedings of IFIPTM 2008: Joint iTrust and PST Conferences on Privacy, Trust Management and Security, June 18-20, 2008, Trondheim, Norway* (2008), pp. 1–16.
- [CFLH14] CLEGER, S., FERNÁNDEZ-LUNA, J. M., AND HUETE, J. F. Learning from explanations in recommender systems. *Information Sciences* 287 (2014), 90–108.
- [CG99] CHIEN, Y.-H., AND GEORGE, E. I. A bayesian model for collaborative filtering. In *AISTATS* (1999).
- [CH96] CHRISTIANSON, B., AND HARBISON, W. S. Why isn't trust transitive? In *International Workshop on Security Protocols* (1996), Springer, pp. 171–176.
- [CI15] CARRERA, Á., AND IGLESIAS, C. A. A systematic review of argumentation techniques for multi-agent systems research. *Artificial Intelligence Review* 44, 4 (Dec 2015), 509–535.
- [CLSQ15] CHEN, K., LIU, G., SHEN, H., AND QI, F. Sociallink: Utilizing social network and transaction links for effective trust management in P2P

- file sharing systems. In *2015 IEEE International Conference on Peer-to-Peer Computing (P2P)* (2015), IEEE, pp. 1–10.
- [CM04a] CHESÑEVAR, C. I., AND MAGUITMAN, A. G. ArgueNet: An argument-based recommender system for solving web search queries. In *Proceedings of the International IEEE Conference on Intelligent Systems (IS 2004)* (June 2004), IEEE, pp. 282–287.
- [CM04b] CHESÑEVAR, C. I., AND MAGUITMAN, A. G. An argumentative approach to assessing natural language usage based on the web corpus. In *Proceedings of the 16th European Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004* (2004), pp. 581–585.
- [CMB07] CANDILLIER, L., MEYER, F., AND BOULLÉ, M. Comparing state-of-the-art collaborative filtering systems. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (2007), Springer, pp. 548–562.
- [CMG09] CHESÑEVAR, C. I., MAGUITMAN, A. G., AND GONZÁLEZ, M. P. Empowering recommendation technologies through argumentation. In *Argumentation in Artificial Intelligence*, G. Simari and I. Rahwan, Eds. Springer US, 2009, pp. 403–422.
- [CMGE16] CHESNEVAR, C. I., MAGUITMAN, A. G., GONZÁLEZ, M. P., AND ESTEVEZ, E. Opinion aggregation and conflict resolution in e-government platforms: Contrasting social media information through argumentation. In *Interdisciplinary Perspectives on Contemporary Conflict Resolution*. IGI Global, 2016, pp. 183–203.
- [CML00] CHESÑEVAR, C. I., MAGUITMAN, A. G., AND LOUI, R. P. Logical models of argument. *ACM Computing Surveys* 32, 4 (2000), 337–383.
- [CMS04] CHESNEVAR, C. I., MAGUITMAN, A. G., AND SIMARI, G. R. A first approach to argument-based recommender systems based on defeasible logic programming. In *10th International Workshop on Non-Monotonic Reasoning (NMR-2004)* (2004), pp. 109–117.

- [CMVGRG⁺15] COLOMBO-MENDOZA, L. O., VALENCIA-GARCÍA, R., RODRÍGUEZ-GONZÁLEZ, A., ALOR-HERNÁNDEZ, G., AND SAMPER-ZAPATER, J. J. RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications* 42, 3 (2015), 1202–1222.
- [CNHAVGGS12] CARRER-NETO, W., HERNÁNDEZ-ALCARAZ, M. L., VALENCIA-GARCÍA, R., AND GARCÍA-SÁNCHEZ, F. Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with Applications* 39, 12 (2012), 10990–11000.
- [CNS03] CARBONE, M., NIELSEN, M., AND SASSONE, V. A formal model for trust in dynamic networks. In *Proceedings of First International Conference on Software Engineering and Formal Methods, 2003* (2003), IEEE, pp. 54–61.
- [Coo01] COOK, K. *Trust in society*. Russell Sage Foundation, 2001.
- [CPCP16] CELDRÁN, A. H., PÉREZ, M. G., CLEMENTE, F. J. G., AND PÉREZ, G. M. Design of a recommender system based on users' behavior and collaborative location and tracking. *Journal of Computational Science* 12 (2016), 83–94.
- [CPGPSM17] COLOMO-PALACIOS, R., GARCÍA-PEÑALVO, F. J., STANTCHEV, V., AND MISRA, S. Towards a social and context-aware mobile recommendation system for tourism. *Pervasive and Mobile Computing* 38 (2017), 505–515.
- [CRL00] CARBOGIM, D. V., ROBERTSON, D., AND LEE, J. Argument-based applications to knowledge engineering. *The Knowledge Engineering Review* 15, 2 (2000), 119–149.
- [CRR⁺13] COBOS, C., RODRIGUEZ, O., RIVERA, J., BETANCOURT, J., MENDOZA, M., LEÓN, E., AND HERRERA-VIEDMA, E. A hybrid system of pedagogical pattern recommendations based on singular value decomposition and variable data attributes. *Information Processing & Management* 49, 3 (2013), 607–625.

- [CSCC13] CHOW, H. K. H., SIU, W., CHAN, C.-K., AND CHAN, H. C. B. An argumentation-oriented multi-agent system for automating the freight planning process. *Expert Systems with Applications* 40, 10 (2013), 3858–3871.
- [CSS99] COHEN, W. W., SCHAPIRE, R. E., AND SINGER, Y. Learning to order things. *Journal of Artificial Intelligence Research* 10 (1999), 243–270.
- [CVS07] CHRISTAKOU, C., VRETTOS, S., AND STAFYLOPATIS, A. A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools* 16, 05 (2007), 771–792.
- [CYKS12] CHOI, K., YOO, D., KIM, G., AND SUH, Y. A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications* 11, 4 (2012), 309–317.
- [Das00] DASGUPTA, P. Trust as a commodity. *Trust: Making and Breaking Cooperative Relations* 4 (2000), 49–72.
- [DBB17] DAHER, J., BRUN, A., AND BOYER, A. *A review on explanations in recommender systems*. PhD thesis, LORIA-Université de Lorraine, 2017.
- [DBES09] DONG, X. L., BERTI-EQUILLE, L., AND SRIVASTAVA, D. Integrating conflicting data: The role of source dependence. *Proceedings of the 35th International Conference on Very Large Databases* 2, 1 (2009), 550–561.
- [dBRB05] DU BOUCHER-RYAN, P., AND BRIDGE, D. Collaborative recommending using formal concept analysis. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence* (2005), Springer, pp. 205–218.
- [dCPLM⁺17] DA COSTA PEREIRA, C., LIAO, B., MALERBA, A., ROTOLO, A., TETTAMANZI, A. G. B., VAN DER TORRE, L. W. N., AND VILLATA,

- S. Handling norms in multi-agent systems by means of formal argumentation. *The IfCoLog Journal of Logics and their Applications* 4, 9 (2017), 3039–3073.
- [DDE⁺17] DEAGUSTINI, C. A. D., DALIBÓN, F., EMANUEL, S., GOTTIFREDI, S., FALAPPA, M. A., CHESÑEVAR, C. I., AND SIMARI, G. R. Defeasible argumentation over relational databases. *Argument & Computation* 8, 1 (2017), 35–59.
- [DDG⁺13] DEAGUSTINI, C. A. D., DALIBÓN, S. E. F., GOTTIFREDI, S., FALAPPA, M. A., CHESÑEVAR, C. I., AND SIMARI, G. R. Relational databases as a massive information source for defeasible argumentation. *Knowledge-Based Systems* 51 (2013), 93–109.
- [Deu62] DEUTSCH, M. Cooperation and trust: Some theoretical notes. *Nebraska Symposium on Motivation* (1962).
- [Deu73] DEUTSCH, M. The resolution of conflict. *American Behavioral Scientist* 17, 2 (1973).
- [DF01] DIRKS, K. T., AND FERRIN, D. L. The role of trust in organizational settings. *Organization Science* 12, 4 (2001), 450–467.
- [DGLM⁺15] DE GEMMIS, M., LOPS, P., MUSTO, C., NARDUCCI, F., AND SEMERARO, G. Semantics-aware content-based recommender systems. In *Recommender Systems Handbook*. Springer, 2015, pp. 119–159.
- [DGS11] DUBOIS, T., GOLBECK, J., AND SRINIVASAN, A. Predicting trust and distrust in social networks. In *PASSAT/SocialCom 2011, Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Conference on Social Computing (SocialCom), Boston, MA, USA, 9-11 Oct., 2011* (2011), IEEE, pp. 418–424.
- [DHS01] DUDA, R. O., HART, P. E., AND STORK, D. G. *Pattern classification, 2nd edition*. Wiley, 2001.

- [DHWW14] DENG, S.-G., HUANG, L.-T., WU, J., AND WU, Z.-H. Trust-based personalized service recommendation: A network perspective. *Journal of Computer Science and Technology* 29, 1 (2014), 69–80.
- [DHX⁺16] DENG, S., HUANG, L., XU, G., WU, X., AND WU, Z. On deep learning for trust-aware recommendations in social networks. *IEEE Transactions on Neural Networks and Learning Systems* 28, 5 (2016), 1164–1177.
- [DI99] DELGADO, J., AND ISHII, N. Memory-based weighted-majority prediction for recommender systems. In *ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation* (1999).
- [DJE10] DE JONG, B. A., AND ELFRING, T. How does trust affect the performance of ongoing teams? The mediating role of reflexivity, monitoring, and effort. *Academy of Management Journal* 53, 3 (2010), 535–549.
- [DK04] DESHPANDE, M., AND KARYPIS, G. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 143–177.
- [DKT06] DUNG, P. M., KOWALSKI, R. A., AND TONI, F. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence* 170, 2 (2006), 114–159.
- [DRP17] D'ANGELO, G., RAMPONE, S., AND PALMIERI, F. Developing a trust model for pervasive computing based on apriori association rules learning and bayesian classification. *Soft Computing* 21, 21 (2017), 6297–6315.
- [Dun95] DUNG, P. M. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77, 2 (1995), 321–358.
- [DYD16] DOU, Y., YANG, H., AND DENG, X. A survey of collaborative filtering algorithms for social recommender systems. In *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)* (2016), IEEE, pp. 40–46.

- [EGVT18] EIRINAKI, M., GAO, J., VARLAMIS, I., AND TSERPES, K. Recommender systems for large-scale social networks: A review of challenges and solutions. *Future Generation Computer Systems* 78 (2018), 413–418.
- [EK09] EVANS, A. M., AND KRUEGER, J. I. The psychology (and economics) of trust. *Social and Personality Psychology Compass* 3, 6 (2009), 1003–1017.
- [ELV13] EIRINAKI, M., LOUTA, M. D., AND VARLAMIS, I. A trust-aware system for personalized user recommendations in social networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44, 4 (2013), 409–421.
- [ESN09] ELSALAMOUNY, E., SASSONE, V., AND NIELSEN, M. HMM-based trust model. In *International Workshop on Formal Aspects in Security and Trust* (2009), Springer, pp. 21–35.
- [FAM15] FILIERI, R., ALGUEZAU, S., AND MCLEAY, F. Why do travelers trust TripAdvisor? Antecedents of trust towards consumer-generated media and its influence on recommendation adoption and word of mouth. *Tourism Management* 51 (2015), 174–185.
- [FB08] FELFERNIG, A., AND BURKE, R. Constraint-based recommender systems: technologies and research issues. In *Proceedings of the 10th International Conference on Electronic Commerce* (2008), ACM, p. 3.
- [FECS07] FERRETTI, E., ERRECALDE, M., GARCÍA, A. J., AND SIMARI, G. R. An application of defeasible logic programming to decision making in a robotic environment. In *Logic Programming and Nonmonotonic Reasoning, 9th International Conference, LPNMR 2007, Tempe, AZ, USA, May 15-17, 2007, Proceedings* (2007), pp. 297–302.
- [Fis01] FISCHER, G. User modeling in human–computer interaction. *User Modeling and User-Adapted Interaction* 11, 1 (Mar 2001), 65–86.
- [Fis16] FISHER, C. The trouble with “trust” in news media. *Communication Research and Practice* 2, 4 (2016), 451–465.

- [FISS98] FREUND, Y., IYER, R. D., SCHAPIRE, R. E., AND SINGER, Y. An efficient boosting algorithm for combining preferences. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998* (1998), pp. 170–178.
- [Fog99] FOGG, B. J. Persuasive technologies - Introduction. *Communications ACM* 42, 5 (1999), 26–29.
- [FTG⁺17] FERRETTI, E., TAMARGO, L. H., GARCÍA, A. J., ERRECALDE, M. L., AND SIMARI, G. R. An approach to decision making based on dynamic argumentation systems. *Artificial Intelligence* 242 (2017), 107–131.
- [FZ11] FRIEDRICH, G., AND ZANKER, M. A taxonomy for generating explanations in recommender systems. *AI Magazine* 32, 3 (2011), 90–98.
- [FZŞMT14] FANG, H., ZHANG, J., ŞENSOY, M., AND MAGNENAT-THALMANN, N. Reputation mechanism for e-commerce in virtual reality environments. *Electronic Commerce Research and Applications* 13, 6 (2014), 409–422.
- [Gab85] GABBAY, D. Theoretical foundations for nonmonotonic reasoning in expert systems. In *Logics and Models of Concurrent Systems*, K. Apt, Ed. Springer-Verlag, Berlin, Germany, 1985.
- [GAH19] GOHARI, F. S., ALIEE, F. S., AND HAGHIGHI, H. A dynamic local–global trust-aware recommendation approach. *Electronic Commerce Research and Applications* 34 (2019), 100838.
- [Gam00a] GAMBETTA, D. Can we trust trust? *Trust: Making and Breaking Cooperative Relations* 13 (2000), 213–237.
- [Gam00b] GAMBETTA, D. Mafia: The price of distrust. *Trust: Making and Breaking Cooperative Relations* (2000), 158–175.
- [GBC⁺14] GRIVOLLA, J., BADIA, T., CAMPO, D. M., SONSONA, M., AND PULIDO, J.-M. A hybrid recommender combining user, item and inter-

action data. *2014 International Conference on Computational Science and Computational Intelligence 1* (2014), 297–301.

- [GC04] GÓMEZ, S. A., AND CHESNEVAR, C. I. A hybrid approach to pattern classification using neural networks and defeasible argumentation. In *Flairs Conference* (2004), pp. 393–398.
- [GCRS13] GARCÍA, A. J., CHESÑEVAR, C. I., ROTSTEIN, N. D., AND SIMARI, G. R. Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems. *Expert Systems with Applications* 40, 8 (2013), 3233–3247.
- [GCS08] GÓMEZ, S. A., CHESÑEVAR, C. I., AND SIMARI, G. R. Defeasible reasoning in web-based forms through argumentation. *International Journal of Information Technology and Decision Making* 7, 1 (2008), 71–101.
- [GDBJ10] GE, M., DELGADO-BATTENFELD, C., AND JANNACH, D. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (2010), ACM, pp. 257–260.
- [GE07] GROH, G., AND EHMIG, C. Recommendations in taste related domains: Collaborative filtering vs. social filtering. In *Proceedings of the 2007 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2007, Sanibel Island, Florida, USA, November 4-7, 2007* (2007), pp. 127–136.
- [GGGL16] GÓMEZ, S. A., GORON, A., GROZA, A., AND LETIA, I. A. Assuring safety in air traffic control systems with argumentation and model checking. *Expert Systems With Applications* 44 (2016), 367–385.
- [GGS10] GOTTIFREDI, S., GARCÍA, A. J., AND SIMARI, G. R. Query-based argumentation in agent programming. In *Advances in Artificial Intelligence - IBERAMIA 2010, 12th Ibero-American Conference on AI, Bahía Blanca, Argentina, November 1-5, 2010. Proceedings* (2010), pp. 284–295.

- [GJG14] GEDIKLI, F., JANNACH, D., AND GE, M. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies* 72, 4 (2014), 367–382.
- [GK99] GORDON, T. F., AND KARACAPILIDIS, N. I. The Zeno argumentation framework. *Künstliche Intelligenz* 13, 3 (1999), 20–29.
- [GKK03] GRABNER-KRÄUTER, S., AND KALUSCHA, E. A. Empirical research in on-line trust: A review and critical assessment. *International Journal of Human-Computer Studies* 58, 6 (2003), 783–812.
- [GKRT04] GUHA, R., KUMAR, R., RAGHAVAN, P., AND TOMKINS, A. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web* (2004), ACM, pp. 403–412.
- [GM75] GOLEMBIEWSKI, R. T., AND MCCONKIE, M. The centrality of interpersonal trust in group processes. *Theories of Group Processes* 7 (1975), 131–185.
- [GNOT92] GOLDBERG, D., NICHOLS, D., OKI, B. M., AND TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35, 12 (December 1992), 61–70.
- [Gol05] GOLBECK, J. A. *Computing and applying trust in web-based social networks*. PhD thesis, University of Maryland, College Park, MD, USA, 2005.
- [Gol06] GOLBECK, J. Generating predictive movie recommendations from trust in social networks. In *Trust Management, 4th International Conference, iTrust 2006, Pisa, Italy, May 16-19, 2006, Proceedings* (2006), pp. 93–104.
- [GPB14] GHAZANFAR, M. A., AND PRÜGEL-BENNETT, A. Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. *Expert Systems with Applications* 41, 7 (2014), 3261–3275.

- [GPH03] GOLBECK, J., PARSIA, B., AND HENDLER, J. Trust networks on the semantic web. In *International Workshop on Cooperative Information Agents* (2003), Springer, pp. 238–249.
- [GRD⁺15] GRBOVIC, M., RADOSAVLJEVIC, V., DJURIC, N., BHAMIDIPATI, N., SAVLA, J., BHAGWAN, V., AND SHARP, D. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), ACM, pp. 1809–1818.
- [GRGP01] GOLDBERG, K. Y., ROEDER, T., GUPTA, D., AND PERKINS, C. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval Journal* 4, 2 (2001), 133–151.
- [GRTS07] GARCÍA, A. J., ROTSTEIN, N. D., TUCAT, M., AND SIMARI, G. R. An argumentative reasoning service for deliberative agents. In *Knowledge Science, Engineering and Management, Second International Conference, KSEM 2007, Melbourne, Australia, November 28-30, 2007, Proceedings* (2007), pp. 128–139.
- [GS⁺99] GETOOR, L., SAHAMI, M., ET AL. Using probabilistic relational models for collaborative filtering. In *Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)* (1999).
- [GS04] GARCÍA, A. J., AND SIMARI, G. R. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming* 4, 1-2 (2004), 95–138.
- [GS09] GUNAWARDANA, A., AND SHANI, G. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research* 10 (2009), 2935–2962.
- [GSK⁺99] GOOD, N., SCHAFFER, J. B., KONSTAN, J. A., BORCHERS, A., SARWAR, B., HERLOCKER, J., RIEDL, J., ET AL. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI-99)* (1999), pp. 439–446.

- [GTGS18] GOTTIFREDI, S., TAMARGO, L. H., GARCÍA, A. J., AND SIMARI, G. R. Arguing about informant credibility in open multi-agent systems. *Artificial Intelligence 259* (2018), 91–109.
- [GUH16] GOMEZ-URIBE, C. A., AND HUNT, N. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS) 6*, 4 (2016), 13:1–13:19.
- [Guy15] GUY, I. Social recommender systems. In *Recommender Systems Handbook*. Springer, 2015, pp. 511–543.
- [GYBO18] GHAFARI, S. M., YAKHCHI, S., BEHESHTI, A., AND ORGUN, M. Social context-aware trust prediction: Methods for identifying fake news. In *International Conference on Web Information Systems Engineering* (2018), Springer, pp. 161–177.
- [GZC⁺09] GUY, I., ZWERDLING, N., CARMEL, D., RONEN, I., UZIEL, E., YOGEV, S., AND OFEK-KOIFMAN, S. Personalized recommendation of social software items based on social relations. In *Proceedings of the third ACM Conference on Recommender Systems* (2009), ACM, pp. 53–60.
- [GZHS16] GU, Y., ZHAO, B., HARDTKE, D., AND SUN, Y. Learning global term weights for content-based recommender systems. In *Proceedings of the 25th International Conference on World Wide Web* (2016), International World Wide Web Conferences Steering Committee, pp. 391–400.
- [GZT14] GUO, G., ZHANG, J., AND THALMANN, D. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems 57* (2014), 57–68.
- [GZY13] GUO, G., ZHANG, J., AND YORKE-SMITH, N. A novel bayesian similarity measure for recommender systems. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013* (2013), pp. 2619–2625.
- [GZYS15a] GUO, G., ZHANG, J., AND YORKE-SMITH, N. Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. *Knowledge-Based Systems 74* (2015), 14–27.

- [GZYS15b] GUO, G., ZHANG, J., AND YORKE-SMITH, N. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015).
- [GZYS16] GUO, G., ZHANG, J., AND YORKE-SMITH, N. A novel recommendation model regularized with user trust and item ratings. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2016), 1607–1620.
- [HAC90] HART, D. M., ANDERSON, S. D., AND COHEN, P. R. Envelopes as a vehicle for improving the efficiency of plan execution. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control* (1990), Morgan Kaufman, pp. 71–76.
- [Har91] HARCOURT, A. H. Help, cooperation and trust in animals. *Cooperation and Prosocial Behaviour* (1991), 15–26.
- [HB18] HEUER, H., AND BREITER, A. Trust in news on social media. In *Proceedings of the 10th Nordic Conference on Human-Computer Interaction* (2018), ACM, pp. 137–147.
- [HC01] HAYES, C., AND CUNNINGHAM, P. Smart radio - community based music radio. *Knowledge-Based Systems* 14, 3-4 (2001), 197–201.
- [HC07] HWANG, C.-S., AND CHEN, Y.-P. Using trust in collaborative filtering recommendation. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (2007), Springer, pp. 1052–1060.
- [HCZ04] HUANG, Z., CHEN, H., AND ZENG, D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 116–142.
- [HdOG08] HERNÁNDEZ DEL OLMO, F., AND GAUDIOSO, E. Evaluation of recommender systems: A new approach. *Expert Systems with Applications* 35, 3 (2008), 790–804.
- [Her88] HERTZBERG, L. On the attitude of trust. *Inquiry* 31, 3 (1988), 307–322.

- [Hes06] HESS, C. Trust-based recommendations for publications - A multi-layer network approach. *IEEE Technical Committee on Digital Libraries Bulletin* 2, 2 (2006).
- [HF06] HUANG, J., AND FOX, M. S. An ontology of trust: Formal semantics and transitivity. In *Proceedings of the 8th International Conference on Electronic Commerce: The New E-Commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet* (New York, NY, USA, 2006), ICEC '06, ACM, pp. 259–270.
- [HJS06] HUYNH, T. D., JENNINGS, N. R., AND SHADBOLT, N. R. Certified reputation: How an agent can trust a stranger. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006* (2006), ACM, pp. 1217–1224.
- [HKBR17] HERLOCKER, J. L., KONSTAN, J. A., BORCHERS, A., AND RIEDL, J. An algorithmic framework for performing collaborative filtering. *SIGIR Forum* 51, 2 (2017), 227–234.
- [HKTR04] HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G., AND RIEDL, J. T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (January 2004), 5–53.
- [Hof99] HOFMANN, T. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (1999), Morgan Kaufmann Publishers Inc., pp. 289–296.
- [Hof03] HOFMANN, T. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (2003), ACM, pp. 259–266.
- [Hof04] HOFMANN, T. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 89–115.

- [HRW08] HUBERMAN, B. A., ROMERO, D. M., AND WU, F. Social networks that matter: Twitter under the microscope. *CoRR abs/0812.1045* (2008).
- [HS08] HESS, C., AND SCHLIEDER, C. Trust-based recommendations for documents. *AI Communications* 21, 2-3 (2008), 145–153.
- [HSRF95] HILL, W., STEAD, L., ROSENSTEIN, M., AND FURNAS, G. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1995), ACM Press/Addison-Wesley Publishing Co., pp. 194–201.
- [Hul98] HULL, D. A. The TREC-7 filtering track: Description and analysis. *Proceedings of The Seventh Text REtrieval Conference, TREC 1998, Gaithersburg, Maryland, USA, November 9-11, 1998* (1998), 45–68.
- [Hun01] HUNTER, A. Hybrid argumentation systems for structured news reports. *The Knowledge Engineering Review* 16, 4 (2001), 295–329.
- [HZ11] HURLEY, N., AND ZHANG, M. Novelty and diversity in top-n recommendation—analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)* 10, 4 (2011), 14.
- [HZF⁺13] HORNING, T., ZIEGLER, C.-N., FRANZ, S., PRZYJACIEL-ZABLOCKI, M., SCHÄTZLE, A., AND LAUSEN, G. Evaluating hybrid music recommender systems. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01* (2013), IEEE Computer Society, pp. 57–64.
- [Jac02] JACKSON, P. *Virtual working: Social and organisational dynamics*. Routledge, 2002.
- [Jan08] JANNACH, D. *Finding preferred query relaxations in content-based recommenders*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 81–97.

- [Jay03] JAYNES, E. T. *Probability theory: The logic of science*. Cambridge University Press, 2003.
- [JE10] JAMALI, M., AND ESTER, M. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems* (New York, NY, USA, 2010), RecSys '10, ACM, pp. 135–142.
- [JF03] JURCA, R., AND FALTINGS, B. An incentive compatible reputation mechanism. In *CEC 2003. IEEE International Conference on E-Commerce, 2003.* (2003), IEEE, pp. 285–292.
- [JGK03] JØSANG, A., GRAY, E., AND KINATEDER, M. Analysing topologies of transitive trust. In *Proceedings of the First International Workshop on Formal Aspects in Security & Trust (FAST2003)* (2003), Pisa, Italy, pp. 9–22.
- [JHP06] JØSANG, A., HAYWARD, R., AND POPE, S. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference-Volume 48* (2006), Australian Computer Society, Inc., pp. 85–94.
- [JI02] JØSANG, A., AND ISMAIL, R. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference* (2002), vol. 5, pp. 2502–2511.
- [Jøs01] JØSANG, A. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 9, 03 (2001), 279–311.
- [JSZ03] JIN, R., SI, L., AND ZHAI, C. Preference-based graphic models for collaborative filtering. In *UAI '03, Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence, Acapulco, Mexico, August 7-10 2003* (2003), pp. 329–336.
- [JSZC03] JIN, R., SI, L., ZHAI, C., AND CALLAN, J. P. Collaborative filtering with decoupled models for preferences and ratings. In *Proceedings of the 2003 ACM CIKM International Conference on Information and*

Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003 (2003), pp. 309–316.

- [JTG19] JOAQUÍN, F., TAMARGO, L. H., AND GARCÍA, A. J. A taxonomy approach for multi-context trust: Formalization and implementation. *Expert Systems with Applications* 127 (2019), 295–307.
- [JZFF10] JANNACH, D., ZANKER, M., FELFERNIG, A., AND FRIEDRICH, G. *Recommender systems: An introduction*. Cambridge University Press, 2010.
- [KA08] KIM, K.-J., AND AHN, H. A recommender system using GA K-means clustering in an online shopping market. *Expert Systems with Applications* 34, 2 (2008), 1200–1209.
- [KA17] KERMANY, N. R., AND ALIZADEH, S. H. A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. *Electronic Commerce Research and Applications* 21 (2017), 50–64.
- [KB12] KRAVARI, K., AND BASSILIADES, N. HARM: A hybrid rule-based agent reputation model based on temporal defeasible logic. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web* (2012), Springer, pp. 193–207.
- [KB15] KOREN, Y., AND BELL, R. M. Advances in collaborative filtering. In *Recommender Systems Handbook*. Springer, 2015, pp. 77–118.
- [KB17] KAMINSKAS, M., AND BRIDGE, D. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 7, 1 (2017), 2.
- [KBGM15] KUMAR BOKDE, D., GIRASE, S., AND MUKHOPADHYAY, D. An item-based collaborative filtering using dimensionality reduction techniques on Mahout framework. In *4th Post Graduate Conference for Information Technology (iPGCon-2015), Sangamner, Published in Spvryan's International Journal of Engineering Science and Technology (SEST)* (2015), pp. 2394–0905.

- [KBV09] KOREN, Y., BELL, R., AND VOLINSKY, C. Matrix factorization techniques for recommender systems. *IEEE Computer* 42, 8 (August 2009), 30–37.
- [KK97] KNACK, S., AND KEEFER, P. Does social capital have an economic payoff? A cross-country investigation. *The Quarterly Journal of Economics* 112, 4 (1997), 1251–1288.
- [KK17] KOOHI, H., AND KIANI, K. A new method to find neighbor users that improves the performance of collaborative filtering. *Expert Systems with Applications* 83 (2017), 30–39.
- [KLM90] KRAUS, S., LEHMANN, D., AND MAGIDOR, M. Non-monotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44 (1990), 167–207.
- [KMM⁺97] KONSTAN, J. A., MILLER, B. N., MALTZ, D., HERLOCKER, J. L., GORDON, L. R., AND RIEDL, J. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM* 40, 3 (1997), 77–87.
- [Kon88] KONOLIGE, K. Defeasible argumentation in reasoning about events. In *Methodologies for Intelligent Systems, 3, Proceedings of the Third International Symposium on Methodologies for Intelligent Systems, Turin, Italy, October 12-15, 1988* (1988), pp. 380–390.
- [KP17] KUNAVER, M., AND POŽRL, T. Diversity in recommender systems - A survey. *Knowledge-Based Systems* 123 (2017), 154–162.
- [KRRT01] KUMAR, R., RAGHAVAN, P., RAJAGOPALAN, S., AND TOMKINS, A. Recommendation systems: A probabilistic analysis. *Journal of Computer and System Sciences* 63, 1 (2001), 42–61.
- [Kru97] KRULWICH, B. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine* 18, 2 (1997), 37–37.
- [KSGM03] KAMVAR, S. D., SCHLOSSER, M. T., AND GARCIA-MOLINA, H. The eigentrust algorithm for reputation management in p2p networks. In

- Proceedings of the 12th International Conference on World Wide Web* (2003), ACM, pp. 640–651.
- [KSP⁺19] KOUKI, P., SCHAFFER, J., PUJARA, J., O'DONOVAN, J., AND GETTOOR, L. Personalized explanations for hybrid recommender systems. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (2019), ACM, pp. 379–390.
- [KWV16] KOTKOV, D., WANG, S., AND VEIJALAINEN, J. A survey of serendipity in recommender systems. *Knowledge-Based Systems 111* (2016), 180–192.
- [Lag92] LAGENSPETZ, O. Legitimacy and trust. *Philosophical Investigations 15*, 1 (1992), 1–21.
- [LAM18] LAIFA, M., AKROUF, S., AND MAMMERI, R. Forgiveness and trust dynamics on social networks. *Adaptive Behavior 26*, 2 (2018), 65–83.
- [Lan95] LANG, K. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*. Elsevier, 1995, pp. 331–339.
- [LBRS12] LAWRENCE, J., BEX, F., REED, C., AND SNAITH, M. AIFdb: Infrastructure for the argument web. In *COMMA* (2012), pp. 515–516.
- [LC10] LABONNE, J., AND CHASE, R. S. A road to trust. *Journal of Economic Behavior & Organization 74*, 3 (2010), 253–261.
- [LCK10] LEE, S. K., CHO, Y. H., AND KIM, S. H. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences 180*, 11 (2010), 2142–2155.
- [LCW02] LEE, M., CHOI, P., AND WOO, Y. A hybrid recommender system combining collaborative filtering with neural network. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (2002), Springer, pp. 531–534.
- [Lee01] LEE, W. S. Collaborative learning for recommender systems. In *In Proc. 18th International Conference on Machine Learning* (2001), Morgan Kaufmann, pp. 314–321.

- [LHC08] LATHIA, N., HAILES, S., AND CAPRA, L. Trust-based collaborative filtering. In *IFIP international Conference on Trust Management* (2008), Springer, pp. 119–134.
- [LHC16] LV, G., HU, C., AND CHEN, S. Research on recommender system based on ontology and genetic algorithm. *Neurocomputing* 187 (2016), 92–97.
- [Li07] LI, P. P. Towards an interdisciplinary conceptualization of trust: A typological approach. *Management and Organization Review* 3, 3 (2007), 421–445.
- [LL07] LEE, J. S., AND LEE, J. C. Context awareness by case-based reasoning in a music recommendation system. In *International Symposium on Ubiquitous Computing Systems* (2007), H. Ichikawa, W.-D. Cho, I. Satoh, and H. Y. Youn, Eds., vol. 4836 of *Lecture Notes in Computer Science*, Springer, Springer, pp. 45–58.
- [LLFZ14] LIANG, T., LIANG, Y., FAN, J., AND ZHAO, J. A hybrid recommendation model based on estimation of distribution algorithms. *Journal of Computational Information Systems* 10 (01 2014), 781–788.
- [LLLZ13] LIKAMWA, R., LIU, Y., LANE, N. D., AND ZHONG, L. MoodScope: Building a mood sensor from smartphone usage patterns. In *The 11th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys'13, Taipei, Taiwan, June 25-28, 2013* (2013), pp. 465–466.
- [LNA⁺97] LOUI, R. P., NORMAN, J., ALTEPETER, J., PINKARD, D., CRAVEN, D., LINSDAY, J., AND FOLTZ, M. A. Progress on room 5: A testbed for public interactive semi-formal legal argumentation. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law, ICAIL '97, Melbourne, Victoria, Australia, June 30 - July 3, 1997* (1997), pp. 207–214.
- [LS19] LOGESH, R., AND SUBRAMANIASWAMY, V. Exploring hybrid recommender systems for personalized travel applications. In *Cognitive Informatics and Soft Computing*. Springer, 2019, pp. 535–544.

- [Luh00] LUHMANN, N. Familiarity, confidence, trust: Problems and alternatives. *Trust: Making and Breaking Cooperative Relations 6* (2000), 94–107.
- [Luh18] LUHMANN, N. *Trust and power*. John Wiley & Sons, 2018.
- [LW94] LITTLESTONE, N., AND WARMUTH, M. K. The weighted majority algorithm. *Information and Computation 108*, 2 (1994), 212–261.
- [MA04] MASSA, P., AND AVESANI, P. Trust-aware collaborative filtering for recommender systems. In *On the Move to Meaningful Internet Systems* (2004), Springer, pp. 492–508.
- [MA09] MASSA, P., AND AVESANI, P. Trust metrics in recommender systems. In *Computing with Social Trust*. Springer, 2009, pp. 259–285.
- [MAB09] MALIK, Z., AKBAR, I., AND BOUGUETTAYA, A. Web services reputation assessment using a hidden Markov model. In *Service-Oriented Computing*. Springer, 2009, pp. 576–591.
- [Mae94] MAES, P. Agents that reduce work and information overload. *Communications of the ACM 37*, 7 (1994), 30–40.
- [Mak94] MAKINSON, D. General patterns in nonmonotonic reasoning. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, H. Gabbay and Robinson, Eds., vol. 3. Oxford University Press, 1994, pp. 35–110.
- [Mar04] MARLIN, B. M. Modeling user rating profiles for collaborative filtering. In *Advances in Neural Information Processing Systems* (2004), pp. 627–634.
- [MBR98] MOONEY, R. J., BENNETT, P. N., AND ROY, L. Book recommending using text categorization with extracted information. In *Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08* (1998).
- [MC96] MCKNIGHT, D. H., AND CHERVANY, N. L. The meanings of trust. *Technical Report MISRC Working Paper Series 96-04, University of Minnesota, Management Information Systems Research Center* (1996).

- [MC01] MCKNIGHT, D. H., AND CHERVANY, N. L. Trust and distrust definitions: One bite at a time. In *Trust in Cyber-Societies*. Springer, 2001, pp. 27–54.
- [McA95] MCALLISTER, D. J. Affect-and cognition-based trust as foundations for interpersonal cooperation in organizations. *Academy of Management Journal* 38, 1 (1995), 24–59.
- [MCG⁺99] MIRANDA, T., CLAYPOOL, M., GOKHALE, A., MIR, T., MURNIKOV, P., NETES, D., AND SARTIN, M. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems* (1999).
- [MDM15] MOIN, S. M. A., DEVLIN, J., AND MCKECHNIE, S. Trust in financial services: Impact of institutional trust and dispositional trust on trusting belief. *Journal of Financial Services Marketing* 20, 2 (2015), 91–106.
- [MDS95] MAYER, R. C., DAVIS, J. H., AND SCHOORMAN, F. D. An integrative model of organizational trust. *Academy of Management Review* 20, 3 (1995), 709–734.
- [MKL09] MA, H., KING, I., AND LYU, M. R. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2009), ACM, pp. 203–210.
- [ML13] MCAULEY, J., AND LESKOVEC, J. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems* (2013), ACM, pp. 165–172.
- [MM09] MCSHERRY, F., AND MIRONOV, I. Differentially private recommender systems: Building privacy into the Netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2009), KDD '09, ACM, pp. 627–636.

- [MMH02] MUI, L., MOHTASHEMI, M., AND HALBERSTADT, A. A computational model of trust and reputation. In *HICSS. Proceedings of the 35th Annual Hawaii International Conference on System Sciences, 2002.* (2002), IEEE, pp. 2431–2439.
- [MMN02] MELVILLE, P., MOONEY, R. J., AND NAGARAJAN, R. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)* (Edmonton, Alberta, 2002), pp. 187–192.
- [MMO07] MURAKAMI, T., MORI, K., AND ORIHARA, R. Metrics for evaluating the serendipity of recommendation lists. In *Annual Conference of the Japanese Society for Artificial Intelligence* (2007), Springer, pp. 40–46.
- [MPB16] MELO, V. S., PANISSON, A. R., AND BORDINI, R. H. Argumentation-based reasoning using preferences over sources of information. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems* (2016), International Foundation for Autonomous Agents and Multiagent Systems, pp. 1337–1338.
- [MPZ03] MCEVILY, B., PERRONE, V., AND ZAHEER, A. Trust as an organizing principle. *Organization Science* 14, 1 (2003), 91–103.
- [MR07] MAHMOOD, T., AND RICCI, F. Towards learning user-adaptive state models in a conversational recommender system. In *LWA* (2007), pp. 373–378.
- [MR09] MAHMOOD, T., AND RICCI, F. Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia* (New York, NY, USA, 2009), HT '09, ACM, pp. 73–82.
- [MRK06] MCNEE, S. M., RIEDL, J., AND KONSTAN, J. A. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems* (2006), ACM, pp. 1097–1101.

- [MS16] MAO, Y., AND SHEN, H. Web of credit: Adaptive personalized trust network inference from online rating data. *IEEE Transactions on Computational Social Systems* 3, 4 (2016), 176–189.
- [MS17] MELVILLE, P., AND SINDHWANI, V. Recommender systems. *Encyclopedia of Machine Learning and Data Mining* (2017), 1056–1066.
- [MSdGL16] MUSTO, C., SEMERARO, G., DE GEMMIS, M., AND LOPS, P. Learning word embeddings from wikipedia for content-based recommender systems. In *European Conference on Information Retrieval* (2016), Springer, pp. 729–734.
- [MSL⁺16] MORENO, M. N., SEGRERA, S., LÓPEZ, V. F., MUÑOZ, M. D., AND SÁNCHEZ, Á. L. Web mining based framework for solving usual problems in recommender systems. A case study for movies' recommendation. *Neurocomputing* 176 (2016), 72–80.
- [MSR04] MIDDLETON, S. E., SHADBOLT, N., AND ROURE, D. D. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems* 22, 1 (2004), 54–88.
- [Mul11] MULLER, T. Semantics of trust. In *Proceedings of the 7th International conference on Formal aspects of security and trust* (Berlin, Heidelberg, 2011), FAST'10, Springer-Verlag, pp. 141–156.
- [MWZW18] MA, G., WANG, Y., ZHENG, X., AND WANG, M. Leveraging transitive trust relations to improve cross-domain recommendation. *IEEE Access* 6 (2018), 38012–38025.
- [NA98] NAKAMURA, A., AND ABE, N. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning* (San Francisco, CA, USA, 1998), ICML '98, Morgan Kaufmann Publishers Inc., pp. 395–403.
- [NbII14] NILASHI, M., BIN IBRAHIM, O., AND ITHNIN, N. Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and neuro-fuzzy system. *Knowledge-Based Systems* 60 (2014), 82–101.

- [new] <http://newstrust.net/>. Accedido Mayo 2012.
- [NHTK18] NGUYEN, T. T., HARPER, F. M., TERVEEN, L., AND KONSTAN, J. A. User personality and user satisfaction with recommender systems. *Information Systems Frontiers* 20, 6 (2018), 1173–1189.
- [NIB18] NILASHI, M., IBRAHIM, O., AND BAGHERIFARD, K. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications* 92 (2018), 507–520.
- [NJ17] NUNES, I., AND JANNACH, D. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction* 27, 3-5 (2017), 393–444.
- [NMO⁺12] NOIA, T. D., MIRIZZI, R., OSTUNI, V. C., ROMITO, D., AND ZANKER, M. Linked open data to support content-based recommender systems. In *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012* (2012), pp. 1–8.
- [NSKA06] NAGURA, R., SEKI, Y., KANDO, N., AND AONO, M. A method of rating the credibility of news documents on the web. In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY, USA, 2006), ACM Press, pp. 683–684.
- [Nut88] NUTE, D. Defeasible reasoning: A philosophical analysis in prolog. In *Aspects of Artificial Intelligence*. Springer, 1988, pp. 251–288.
- [OBR⁺08] OFUONYE, E., BEATTY, P., REAY, I., DICK, S., AND MILLER, J. How do we build trust into e-commerce web sites? *IEEE Software* 25, 5 (2008), 7–9.
- [OGDN⁺13] OSTUNI, V. C., GENTILE, G., DI NOIA, T., MIRIZZI, R., ROMITO, D., AND DI SCIASCIO, E. Mobile movie recommendations with linked data. In *International Conference on Availability, Reliability, and Security* (2013), Springer, pp. 400–415.

- [OMDG17] OBADIĆ, I., MADJAROV, G., DIMITROVSKI, I., AND GJORGJEVIKJ, D. Addressing item-cold start problem in recommendation systems using model based approach and deep learning. In *International Conference on ICT Innovations* (2017), Springer, pp. 176–185.
- [OS05] O'DONOVAN, J., AND SMYTH, B. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces* (2005), ACM, pp. 167–174.
- [Pag00] PAGDEN, A. The destruction of trust and its economic consequences in the case of eighteenth-century Naples. *Trust: Making and Breaking Cooperative Relations* (2000), 127–144.
- [PAL⁺14] PARSONS, S., ATKINSON, K., LI, Z., MCBURNEY, P., SKLAR, E., SINGH, M., HAIGH, K., LEVITT, K., AND ROWE, J. Argument schemes for reasoning about trust. *Argument & Computation* 5, 2-3 (2014), 160–190.
- [Paz99] PAZZANI, M. J. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review* 13, 5-6 (1999), 393–408.
- [PB97] PAZZANI, M. J., AND BILLSUS, D. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning* 27, 3 (1997), 313–331.
- [PBW17] PARADARAMI, T. K., BASTIAN, N. D., AND WIGHTMAN, J. L. A hybrid recommender system using artificial neural networks. *Expert Systems with Applications* 83 (2017), 300–313.
- [PG18] PARDO, P., AND GODO, L. A temporal argumentation approach to cooperative planning using dialogues. *Journal of Logic and Computation* 28, 3 (2018), 551–580.
- [PH13] PAN, X. S., AND HOUSER, D. Cooperation during cultural group formation promotes trust towards members of out-groups. *Proceedings of the Royal Society B: Biological Sciences* 280, 1762 (2013), 20130606.

- [PHC07] PARK, M.-H., HONG, J.-H., AND CHO, S.-B. Location-based recommendation system using bayesian user's preference model in mobile devices. In *International Conference on Ubiquitous Intelligence and Computing* (2007), Springer, pp. 1130–1139.
- [PHLG00] PENNOCK, D. M., HORVITZ, E., LAWRENCE, S., AND GILES, C. L. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence* (2000), Morgan Kaufmann Publishers Inc., pp. 473–480.
- [PKS06] PARK, S., KANG, S., AND SONG, J. PolyNews: Delivering multiple aspects of news to mitigate media bias. Technical report, KAIST, 2006.
- [PM03] PARSONS, S., AND MCBURNEY, P. Argumentation-based dialogues for agent coordination. *Group Decision and Negotiation* 12, 5 (2003), 415–439.
- [PM06] PITSILIS, G., AND MARSHALL, L. A trust-enabled P2P recommender system. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on* (2006), IEEE, pp. 59–64.
- [PMS10] PARSONS, S., MCBURNEY, P., AND SKLAR, E. Reasoning about trust using argumentation: A position paper. In *International Workshop on Argumentation in Multi-Agent Systems* (2010), Springer, pp. 159–170.
- [Pol87] POLLOCK, J. L. Defeasible reasoning. *Cognitive Science* 11, 4 (1987), 481–518.
- [Pol95] POLLOCK, J. L. *Cognitive carpentry: A blueprint for how to build a person*. MIT Press, Cambridge, MA, USA, 1995.
- [Poo85] POOLE, D. On the comparison of theories: Preferring the most specific explanation. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, USA, August 1985* (1985), pp. 144–147.

- [PP03] PAVLOV, D. Y., AND PENNOCK, D. M. A maximum entropy approach to collaborative filtering in dynamic, sparse, high-dimensional domains. In *Advances in neural information processing systems* (2003), pp. 1465–1472.
- [PPK05] PAPAGELIS, M., PLEXOUSAKIS, D., AND KUTSURAS, T. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *Trust Management*. Springer, 2005, pp. 224–239.
- [PPL01] POPESCU, A., PENNOCK, D. M., AND LAWRENCE, S. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence* (2001), Morgan Kaufmann Publishers Inc., pp. 437–444.
- [Pra07] PRADE, H. A qualitative bipolar argumentative view of trust. In *International Conference on Scalable Uncertainty Management* (2007), Springer, pp. 268–276.
- [Pra10] PRAKKEN, H. An abstract framework for argumentation with structured arguments. *Argument & Computation* 1, 2 (2010), 93–124.
- [PS96] PRAKKEN, H., AND SARTOR, G. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law* 4, 3 (Sep 1996), 331–368.
- [PS97] PRAKKEN, H., AND SARTOR, G. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-Classical Logics* 7, 1 (1997), 25–75.
- [PS02] PRAKKEN, H., AND SARTOR, G. The role of logic in computational models of legal argument: A critical survey. In *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II* (2002), pp. 342–381.
- [PSJ98] PARSONS, S., SIERRA, C., AND JENNINGS, N. R. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8, 3 (1998), 261–292.

- [PSM13] PINYOL, I., AND SABATER-MIR, J. Computational trust and reputation models for open multi-agent systems: A review. *Artificial Intelligence Review* 40, 1 (2013), 1–25.
- [PTLMHV12] PORCEL, C., TEJEDA-LORENTE, A., MARTÍNEZ, M. A., AND HERRERA-VIEDMA, E. A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. *Information Sciences* 184, 1 (2012), 1–19.
- [PTS⁺11] PARSONS, S., TANG, Y., SKLAR, E., MCBURNEY, P., AND CAI, K. Argumentation-based reasoning in agents with varying degrees of trust. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2* (2011), International Foundation for Autonomous Agents and Multiagent Systems, pp. 879–886.
- [PV02] PRAKKEN, H., AND VREESWIJK, G. Logics for defeasible argumentation. In *Handbook of Philosophical Logic*, D. Gabbay and F. Guenther, Eds., vol. 4. Kluwer Academic Publishers, 2002, pp. 218–319.
- [PZS14] PICHL, M., ZANGERLE, E., AND SPECHT, G. Combining Spotify and Twitter data for generating a recent and public dataset for music recommendation. In *Grundlagen von Datenbanken* (2014), pp. 35–40.
- [QCJ18] QUADRANA, M., CREMONESI, P., AND JANNACH, D. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)* 51, 4 (2018).
- [RA06] RAHWAN, I., AND AMGOUD, L. An argumentation based approach for practical reasoning. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006* (2006), pp. 347–354.
- [RAC⁺02] RASHID, A. M., ALBERT, I., COSLEY, D., LAM, S. K., MCNEE, S. M., KONSTAN, J. A., AND RIEDL, J. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces* (2002), ACM, pp. 127–134.

- [Rah08] RAHWAN, I. Mass argumentation and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web* 6, 1 (2008), 29–37.
- [RD16] RUAN, Y., AND DURRESI, A. A survey of trust management systems for online social communities—trust modeling, trust inference and attacks. *Knowledge-Based Systems* 106 (2016), 150–163.
- [RGS07] ROTSTEIN, N. D., GARCÍA, A. J., AND SIMARI, G. R. Reasoning from desires to intentions: A dialectical framework. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada* (2007), pp. 136–141.
- [Ric79] RICH, E. User modeling via stereotypes. *Cognitive Science* 3, 4 (1979), 329–354.
- [Ric02] RICCI, F. Travel recommender systems. *IEEE Intelligent Systems* 17 (2002), 55–57.
- [RIS⁺94] RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., AND RIEDL, J. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (1994), ACM, pp. 175–186.
- [RKR08] RASHID, A. M., KARYPIS, G., AND RIEDL, J. Learning preferences of new users in recommender systems: An information theoretic approach. *ACM Sigkdd Explorations Newsletter* 10, 2 (2008), 90–100.
- [RRJ⁺03] RAHWAN, I., RAMCHURN, S. D., JENNINGS, N. R., MCBURNEY, P., PARSONS, S., AND SONENBERG, L. Argumentation-based negotiation. *The Knowledge Engineering Review* 18, 4 (2003), 343–375.
- [RRSK11] RICCI, F., ROKACH, L., SHAPIRA, B., AND KANTOR, P. B., Eds. *Recommender systems handbook*. Springer, 2011.
- [RS09] RAHWAN, I., AND SIMARI, G. R. *Argumentation in artificial intelligence*, 1st ed. Springer Publishing Company, Incorporated, 2009.

- [RSBC98] ROUSSEAU, D. M., SITKIN, S. B., BURT, R. S., AND CAMERER, C. Not so different after all: A cross-discipline view of trust. *Academy of Management Review* 23, 3 (1998), 393–404.
- [RV97] RESNICK, P., AND VARIAN, H. R. Recommender systems. *Communications of the ACM* 40, 3 (1997), 56–58.
- [RVN18] REDDY, V. B., VENKATARAMAN, S., AND NEGI, A. A dynamic trust evolution model for MANETs based on mobility. *International Journal of Ad Hoc and Ubiquitous Computing* 28, 4 (2018), 230–246.
- [RW00] ROBERTSON, S. E., AND WALKER, S. Threshold setting in adaptive filtering. *Journal of Documentation* 56, 3 (2000), 312–331.
- [RW01] REED, C., AND WALTON, D. Applications of argumentation schemes. *Conference of the Ontario Society for the Study of Argument (OSSA2001)* (2001).
- [SA06] SALTER, J., AND ANTONOPOULOS, N. CinemaScreen recommender agent: Combining collaborative and content-based filtering. *IEEE Intelligent Systems* 21, 1 (2006), 35–41.
- [SA09] SCHIAFFINO, S., AND AMANDI, A. Building an expert travel agent as a software agent. *Expert Systems with Applications* 36, 2 (2009), 1291–1299.
- [Sab02] SABATER, J. *Trust and reputation for agent societies*. PhD thesis, Universitat Autònoma de Barcelona, 2002.
- [Sal89] SALTON, G. *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989.
- [SBL⁺04] STAAB, S., BHARGAVA, B., LESZEK, L., ROSENTHAL, A., WINSLETT, M., SLOMAN, M., DILLON, T. S., CHANG, E., HUSSAIN, F., NEJDL, W., ET AL. The pudding of trust. *IEEE Intelligent Systems* 19, 5 (2004), 74–88.
- [SC08] STANKEVICIUS, A. G., AND CAPOBIANCO, M. On cummulativity in the context of defeasible argumentation. In *XIV Congreso Argentino de Ciencias de la Computación, CACIC 2008* (2008).

- [SFHS07] SCHAFFER, J. B., FRANKOWSKI, D., HERLOCKER, J., AND SEN, S. *Collaborative filtering recommender systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, ch. 9, pp. 291–324.
- [SFR00] SCHILLO, M., FUNK, P., AND ROVATSOS, M. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence* 14, 8 (2000), 825–848.
- [SGC04] SIMARI, G. R., GARCÍA, A. J., AND CAPOBIANCO, M. Actions, planning and defeasible reasoning. In *10th International Workshop on Non-Monotonic Reasoning (NMR 2004)* (2004), pp. 377–384.
- [SGCS03] STOLZENBURG, F., GARCÍA, A. J., CHESÑEVAR, C. I., AND SIMARI, G. R. Computing generalized specificity. *Journal of Applied Non-Classical Logics* 13, 1 (2003), 87–113.
- [SH01] SOMLO, G., AND HOWE, A. E. Adaptive lightweight text filtering. In *Advances in Intelligent Data Analysis, 4th International Conference, IDA 2001, Cascais, Portugal, September 13-15, 2001, Proceedings* (2001), pp. 319–329.
- [Sha81] SHAW, M. E. *Group dynamics: The psychology of small group behavior*. McGraw-Hill Series in Psychology. McGraw-Hill, 1981.
- [Sha87] SHAPIRO, S. P. The social control of impersonal trust. *American Journal of Sociology* 93, 3 (1987), 623–658.
- [SHB05] SHANI, G., HECKERMAN, D., AND BRAFMAN, R. I. An MDP-based recommender system. *Journal of Machine Learning Research* 6 (December 2005), 1265–1295.
- [SHZK05] SONG, S., HWANG, K., ZHOU, R., AND KWOK, Y.-K. Trusted P2P transactions with fuzzy reputation aggregation. *IEEE Internet Computing* 9, 6 (2005), 24–34.
- [SJ03] SI, L., AND JIN, R. Flexible mixture model for collaborative filtering. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (2003), pp. 704–711.

- [SK09] SU, X., AND KHOSHGOFTAAR, T. M. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence 2009* (2009).
- [SK12] SHINDE, S. K., AND KULKARNI, U. Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems with Applications* 39, 1 (2012), 1381–1387.
- [SKKR00a] SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. T. Application of dimensionality reduction in recommender systems: A case study. In *WebKDD Workshop at the ACM SIGKDD* (2000).
- [SKKR00b] SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00), Minneapolis, MN, USA, October 17-20, 2000* (2000), pp. 158–167.
- [SKKR01] SARWAR, B. M., KARYPIS, G., KONSTAN, J. A., AND RIEDL, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001* (2001), pp. 285–295.
- [SKLB17] SEO, Y.-D., KIM, Y.-G., LEE, E., AND BAIK, D.-K. Personalized recommender system based on friendship strength in social network services. *Expert Systems with Applications* 69 (2017), 135–148.
- [SKR01] SCHAFFER, J. B., KONSTAN, J. A., AND RIEDL, J. E-commerce recommendation applications. *Data Mining and Knowledge Discovery* 5, 1-2 (2001), 115–153.
- [SKS⁺16] STEINBERGER, J., KUHNERT, B., SPEROTTO, A., BAIER, H., AND PRAS, A. In whom do we trust-sharing security events. In *IFIP International Conference on Autonomous Infrastructure, Management and Security* (2016), Springer, pp. 111–124.
- [SKY05] SUN, X., KONG, F., AND YE, S. A comparison of several algorithms for collaborative filtering in startup stage. In *Proceedings. 2005 IEEE Networking, Sensing and Control, 2005.* (2005), IEEE, pp. 25–28.

- [SL92] SIMARI, G. R., AND LOUI, R. P. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence* 53, 2-3 (1992), 125–157.
- [SL11] SHAMBOUR, Q., AND LU, J. A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services. *International Journal of Intelligent Systems* 26, 9 (2011), 814–843.
- [SL12] SHAMBOUR, Q., AND LU, J. A trust-semantic fusion-based recommendation approach for e-business applications. *Decision Support Systems* 54, 1 (2012), 768–780.
- [SLC⁺17] SUBRAMANIASWAMY, V., LOGESH, R., CHANDRASHEKHAR, M., CHALLA, A., AND VIJAYAKUMAR, V. A personalised movie recommendation system based on collaborative filtering. *International Journal of High Performance Computing and Networking* 10, 1-2 (2017), 54–63.
- [SM93] SHETH, B., AND MAES, P. Evolving agents for personalized information filtering. In *Proceedings of Ninth Conference on Artificial Intelligence for Applications* (1993), IEEE, pp. 345–352.
- [SM95] SHARDANAND, U., AND MAES, P. Social information filtering: Algorithms for automating “word of mouth”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1995), ACM Press/Addison-Wesley Publishing Co., pp. 210–217.
- [SMCS09] SAGUI, F. M., MAGUITMAN, A. G., CHESÑEVAR, C. I., AND SIMARI, G. R. Modeling news trust: A defeasible logic programming approach. *Iberoamerican Journal of Artificial Intelligence* 12, 40 (2009), 63–72.
- [Smy07] SMYTH, B. Case-based recommendation. In *The Adaptive Web*. Springer, 2007, pp. 342–376.
- [SN99] SOBOROFF, I., AND NICHOLAS, C. Combining content and collaboration in text filtering. In *Proceedings of the IJCAI* (1999), vol. 99, sn, pp. 86–91.

- [SNM09] SYMEONIDIS, P., NANOPOULOS, A., AND MANOLOPOULOS, Y. Mo-viExplain: A recommender system with explanations. *Recommender Systems 9* (2009), 317–320.
- [SNP13] SHERCHAN, W., NEPAL, S., AND PARIS, C. A survey of trust in social networks. *ACM Computing Surveys (CSUR)* 45, 4 (2013), 47.
- [SPL⁺16] SKLAR, E. I., PARSONS, S., LI, Z., SALVIT, J., PERUMAL, S., WALL, H., AND MANGELS, J. Evaluation of a trust-modulated argumentation-based interactive decision-making tool. *Autonomous Agents and Multi-Agent Systems* 30, 1 (2016), 136–173.
- [SPUP02] SCHEIN, A. I., POPESCU, A., UNGAR, L. H., AND PENNOCK, D. M. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2002), ACM, pp. 253–260.
- [SPX04] SONG, W., PHOHA, V. V., AND XU, X. The HMM-based model for evaluating recommender’s reputation. In *IEEE International Conference on E-Commerce Technology for Dynamic E-Business* (2004), IEEE, pp. 209–215.
- [SS⁺01a] SINHA, R. R., SWEARINGEN, K., ET AL. Comparing recommendations made by online systems and friends. In *DELOS Workshop: Personalization and Recommender Systems in Digital Libraries* (2001), vol. 106.
- [SS01b] SWEARINGEN, K., AND SINHA, R. Beyond algorithms: An HCI perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems* (2001), vol. 13, Citeseer, pp. 1–11.
- [SS02] SINHA, R. R., AND SWEARINGEN, K. The role of transparency in recommender systems. In *Extended abstracts of the 2002 Conference on Human Factors in Computing Systems, CHI 2002, Minneapolis, Minnesota, USA, April 20-25, 2002* (2002), pp. 830–831.
- [SS05] SABATER, J., AND SIERRA, C. Review on computational trust and reputation models. *Artificial intelligence review* 24, 1 (2005), 33–60.

- [Sto02] STOLLE, D. Trusting strangers—the concept of generalized trust in perspective. *Österreichische Zeitschrift für Politikwissenschaft* 31, 4 (2002), 397–412.
- [SW14] SCHERER, S., AND WIMMER, M. A. Conceptualising trust in e-participation contexts. In *International Conference on Electronic Participation* (2014), Springer, pp. 64–77.
- [SXL⁺17] SU, K., XIAO, B., LIU, B., ZHANG, H., AND ZHANG, Z. TAP: A personalized trust-aware QoS prediction approach for web service recommendation. *Knowledge-Based Systems* 115 (2017), 55–65.
- [Szt99] SZTOMPKA, P. *Trust: A sociological theory*. Cambridge University Press, 1999.
- [TA12] TAVAKOLIFARD, M., AND ALMEROOTH, K. C. Social computing: An intersection of recommender systems, trust/reputation systems, and social networks. *IEEE Network* 26, 4 (2012), 53–58.
- [TC00] TRAN, T., AND COHEN, R. Hybrid recommender systems for electronic commerce. In *Proc. Knowledge-Based Electronic Markets, Papers from the AAI Workshop, Technical Report WS-00-04, AAI Press* (2000).
- [TGGS15] TEZE, J. C. L., GOTTIFREDI, S., GARCÍA, A. J., AND SIMARI, G. R. Improving argumentation-based recommender systems through context-adaptable selection criteria. *Expert Systems with Applications* 42, 21 (2015), 8243–8258.
- [TH12] TSAI, C.-F., AND HUNG, C. Cluster ensembles in collaborative filtering recommendation. *Applied Soft Computing* 12, 4 (2012), 1417–1425.
- [THA⁺97] TERVEEN, L., HILL, W., AMENTO, B., McDONALD, D., AND CRETER, J. PHOAKS: A system for sharing recommendations. *Communications of the ACM* 40, 3 (1997), 59–62.
- [THL13] TANG, J., HU, X., AND LIU, H. Social recommendation: A review. *Social Network Analysis and Mining* 3, 4 (2013), 1113–1133.

- [TKG07] TAGHIPOUR, N., KARDAN, A., AND GHIDARY, S. S. Usage-based web recommendations: a reinforcement learning approach. In *ACM Recommender Systems 2007* (2007), pp. 113–120.
- [TLU06] TOIVONEN, S., LENZINI, G., AND UUSITALO, I. Context-aware trust evaluation functions for dynamic reconfigurable systems. In *Proceedings of the Workshop on Models of Trust for the Web (MTW 2006), held in Conjunction with the 15th International World Wide Web Conference* (2006), vol. 190.
- [TM07] TINTAREV, N., AND MASTHOFF, J. A survey of explanations in recommender systems. In *Proceedings of the 23rd International Conference on Data Engineering Workshops, ICDE 2007, 15-20 April 2007, Istanbul, Turkey* (2007), pp. 801–810.
- [TM11] TINTAREV, N., AND MASTHOFF, J. Designing and evaluating explanations for recommender systems. In *Recommender Systems Handbook*. Springer, 2011, pp. 479–510.
- [TM12] TINTAREV, N., AND MASTHOFF, J. Evaluating the effectiveness of explanations for recommender systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 399–439.
- [TNM18] TARUS, J. K., NIU, Z., AND MUSTAFA, G. Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review* 50, 1 (2018), 21–48.
- [TNY17] TARUS, J. K., NIU, Z., AND YOUSIF, A. A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems* 72 (2017), 37–48.
- [Tou58] TOULMIN, S. E. *The uses of argument*. Cambridge University Press, 1958.
- [Tri85] TRIVERS, R. *Social evolution*. No. 156 T7. California: Cummings, 1985.

- [Twa16] TWARDOWSKI, B. Modelling contextual information in session-aware recommender systems with neural networks. In *Proceedings of the 10th ACM Conference on Recommender Systems* (2016), ACM, pp. 273–276.
- [Tyl01] TYLER, T. R. Why do people rely on others? Social identity and social aspects of trust. *Trust in Society, Thousand Oaks* (2001), 285–306.
- [UF98] UNGAR, L. H., AND FOSTER, D. P. Clustering methods for collaborative filtering. In *AAAI Workshop on Recommendation Systems* (1998), vol. 1, pp. 114–129.
- [VBGVDT11] VILLATA, S., BOELLA, G., GABBAY, D. M., AND VAN DER TORRE, L. Arguing about the trustworthiness of the information sources. In *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (Berlin, Heidelberg, 2011), ECSQARU’11, Springer-Verlag, pp. 74–85.
- [VBGVDT13] VILLATA, S., BOELLA, G., GABBAY, D. M., AND VAN DER TORRE, L. A socio-cognitive model of trust using argumentation theory. *International Journal of Approximate Reasoning* 54, 4 (2013), 541–559.
- [VCDCT08] VICTOR, P., CORNELIS, C., DE COCK, M., AND TEREDESAI, A. M. Key figure impact in trust-enhanced recommender systems. *AI Communications* 21, 2-3 (2008), 127–143.
- [VCVHV12] VERBIEST, N., CORNELIS, C., VICTOR, P., AND HERRERA-VIEDMA, E. Trust and distrust aggregation enhanced with path length incorporation. *Fuzzy Sets and Systems* 202 (2012), 61 – 74. Theme: Aggregation Functions.
- [Ver03a] VERHEIJ, B. Artificial argument assistants for defeasible argumentation. *Artificial Intelligence* 150, 1-2 (2003), 291–324.
- [Ver03b] VERHEIJ, B. DefLog: On the logical interpretation of prima facie justified assumptions. *Journal of Logic and Computation* 13, 3 (2003), 319–346.

- [Vil05] VILJANEN, L. Towards an ontology of trust. In *International Conference on Trust, Privacy and Security in Digital Business* (2005), Springer, pp. 175–184.
- [VM07] VOZALIS, M. G., AND MARGARITIS, K. G. Using SVD and demographic data for the enhancement of generalized collaborative filtering. *Information Sciences* 177, 15 (2007), 3017–3037.
- [VMVS00] VAN METEREN, R., AND VAN SOMEREN, M. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop* (2000), pp. 47–56.
- [Vre91] VREESWIJK, G. The feasibility of defeat in defeasible reasoning. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991.* (1991), pp. 526–534.
- [Wat05] WATSON, M. L. Can there be just one trust? A cross-disciplinary identification of trust definitions and measurement. *The Institute for Public Relations* (2005), 1–25.
- [WBCC09] WARDEH, M., BENCH-CAPON, T., AND COENEN, F. Padua: A protocol for argumentation dialogue using association rules. *Artificial Intelligence and Law* 17, 3 (2009), 183–215.
- [WBS08] WALTER, F., BATTISTON, S., AND SCHWEITZER, F. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems* 16 (2008), 57–74. 10.1007/s10458-007-9021-x.
- [WBS09] WALTER, F. E., BATTISTON, S., AND SCHWEITZER, F. Personalised and dynamic trust in social networks. In *Proceedings of the third ACM Conference on Recommender Systems* (2009), ACM, pp. 197–204.
- [WCFHV17] WU, J., CHICLANA, F., FUJITA, H., AND HERRERA-VIEDMA, E. A visual interaction consensus model for social network group decision

- making with trust propagation. *Knowledge-Based Systems 122* (2017), 39–50.
- [WCL14] WU, M.-L., CHANG, C.-H., AND LIU, R.-Z. Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. *Expert Systems with Applications 41*, 6 (2014), 2754–2761.
- [WHC⁺17] WEI, J., HE, J., CHEN, K., ZHOU, Y., AND TANG, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Systems with Applications 69* (2017), 29–39.
- [WL92] WOO, T. Y. C., AND LAM, S. S. Authentication for distributed systems. *IEEE Computer*, 1 (1992), 39–52.
- [WLL15] WANG, Y., LI, L., AND LIU, G. Social context-aware trust inference for trust enhancement in social network based recommendations on service providers. *World Wide Web 18*, 1 (2015), 159–184.
- [WPB01] WEBB, G. I., PAZZANI, M. J., AND BILLSUS, D. Machine learning for user modeling. *User Modeling and User-Adapted Interaction 11*, 1-2 (March 2001), 19–29.
- [WRIJ05] WISHART, R., ROBINSON, R., INDULSKA, J., AND JØSANG, A. SuperstringRep: Reputation-enhanced service discovery. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science—Volume 38* (2005), Australian Computer Society, Inc., pp. 49–57.
- [WSWW17] WU, D., SI, S., WU, S., AND WANG, R. Dynamic trust relationships aware data privacy protection in mobile crowd-sensing. *IEEE Internet of Things Journal 5*, 4 (2017), 2958–2970.
- [WV07] WANG, Y., AND VASSILEVA, J. A review on trust and reputation for web service selection. In *ICDCSW'07. 27th International Conference on Distributed Computing Systems Workshops* (2007), IEEE.
- [WvEH16] WYNER, A. Z., VAN ENGERS, T. M., AND HUNTER, A. Working on the argument pipeline: Through flow issues between natural language

- ge argument, instantiated arguments, and argumentation frameworks. *Argument & Computation* 7, 1 (2016), 69–89.
- [XL04] XIONG, L., AND LIU, L. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering* 16, 7 (2004), 843–857.
- [Yam90] YAMAMOTO, Y. A morality based on trust: Some reflections on japanese morality. *Philosophy East and West* 40, 4 (1990), 451–469.
- [YH13] YANG, W.-S., AND HWANG, S.-Y. iTravel: A recommender system in mobile peer-to-peer environment. *Journal of Systems and Software* 86, 1 (2013), 12–20.
- [YK00] YANIV, I., AND KLEINBERGER, E. Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational Behavior and Human Decision Processes* 83, 2 (2000), 260–281.
- [YKSN18] YADAV, S., KUMAR, V., SINHA, S., AND NAGPAL, S. Trust aware recommender system using swarm intelligence. *Journal of Computational Science* 28 (2018), 180–192.
- [YLLL16] YANG, B., LEI, Y., LIU, J., AND LI, W. Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2016), 1633–1647.
- [YMS11] YAJIMA, T., MATSUMOTO, A., AND SHIGENO, H. PTrust: Provisional value based trust for reputation aggregation in peer-to-peer networks. In *2011 1st International Symposium on Access Spaces (ISAS)* (2011), IEEE, pp. 180–185.
- [YS00] YU, B., AND SINGH, M. P. A social mechanism of reputation management in electronic communities. In *International Workshop on Cooperative Information Agents* (2000), Springer, pp. 154–165.
- [YS02] YU, B., AND SINGH, M. P. An evidential model of distributed reputation management. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1* (2002), ACM, pp. 294–301.

- [YSS04] YU, B., SINGH, M. P., AND SYCARA, K. Developing trust in large-scale peer-to-peer systems. In *IEEE First Symposium on Multi-Agent Security and Survivability, 2004* (2004), IEEE, pp. 1–10.
- [YST⁺04] YU, K., SCHWAIGHOFER, A., TRESP, V., XU, X., AND KRIEGEL, H.-P. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering* 16, 1 (2004), 56–69.
- [YST⁺13] YING, J.-C., SHI, B.-N., TSENG, V. S., TSAI, H.-W., CHENG, K. H., AND LIN, S.-C. Preference-aware community detection for item recommendation. In *2013 Conference on Technologies and Applications of Artificial Intelligence* (2013), IEEE, pp. 49–54.
- [YWZ⁺16] YANG, Z., WU, B., ZHENG, K., WANG, X., AND LEI, L. A survey of collaborative filtering-based recommender systems for mobile Internet applications. *IEEE Access* 4 (2016), 3273–3287.
- [YXT⁺02] YU, K., XU, X., TAO, J., ESTER, M., AND KRIEGEL, H.-P. Instance selection techniques for memory-based collaborative filtering. In *Proceedings of the 2002 SIAM International Conference on Data Mining* (2002), SIAM, pp. 59–74.
- [YXYG15] YU, Z., XU, H., YANG, Z., AND GUO, B. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Transactions on Human-Machine Systems* 46, 1 (2015), 151–158.
- [ZC01] ZHANG, Y., AND CALLAN, J. Maximum likelihood estimation for filtering thresholds. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2001), ACM, pp. 294–302.
- [ZCM02] ZHANG, Y., CALLAN, J., AND MINKA, T. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2002), ACM, pp. 81–88.

- [ZF07] ZHANG, Y., AND FANG, Y. A fine-grained reputation system for reliable service selection in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems* 18, 8 (2007), 1134–1145.
- [ZH07] ZHOU, R., AND HWANG, K. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel & Distributed Systems*, 4 (2007), 460–473.
- [ZH08] ZHANG, M., AND HURLEY, N. Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM conference on Recommender systems* (2008), ACM, pp. 123–130.
- [Zha13] ZHANG, L. The definition of novelty in recommendation system. *Journal of Engineering Science & Technology Review* 6, 3 (2013).
- [ZKL⁺10] ZHOU, T., KUSCSIK, Z., LIU, J.-G., MEDO, M., WAKELING, J. R., AND ZHANG, Y.-C. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences* 107, 10 (2010), 4511–4515.
- [ZM00] ZACHARIA, G., AND MAES, P. Trust management through reputation mechanisms. *Applied Artificial Intelligence* 14, 9 (2000), 881–907.
- [ZMKL05] ZIEGLER, C.-N., MCNEE, S. M., KONSTAN, J. A., AND LAUSEN, G. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web* (2005), ACM, pp. 22–32.
- [ZMP98] ZAHEER, A., MCEVILY, B., AND PERRONE, V. Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance. *Organization science* 9, 2 (1998), 141–159.