Recommending Contacts in Social Networks Using Information Retrieval Models

Javier Sanz-Cruzado
Universidad Autónoma de Madrid
javier.sanz-cruzado@uam.es

Sofía M. Pepa
GMV Information Technologies and Services
sofiamarinapepa@gmail.com

Pablo Castells
Universidad Autónoma de Madrid
pablo.castells@uam.es

ABSTRACT

The fast expansion of online social networks has given rise to new challenges and opportunities for information retrieval and, as a particular area, recommender systems. A particularly compelling problem in this context is recommending contacts, that is, automatically predicting people that a given user may wish or benefit from connecting to in the network. This task has interesting particularities compared to more traditional recommendation domains, a salient one being that recommended items belong to the same space as the users they are recommended to. In this paper, we explore the connection between the contact recommendation and the information retrieval (IR) tasks. Specifically, we research the adaptation of IR models for recommending contacts in social networks. We report experiments over data downloaded from Twitter where we observe that IR models are competitive compared to state-of-the art contact recommendation methods.

KEYWORDS

contact recommendation; social networks; information retrieval; recommender systems; IR models

ACM Reference format:

J. Sanz-Cruzado S. M. Pepa and P. Castells. 2018. Recommending Contacts in Social Networks Using Information Retrieval Models. Proceedings of the 5th Spanish Conference on Information Retrieval (CERI 2018), *Zaragoza, Spain, June 2018*, Article no. 19, 8 pages.

1 INTRODUCCIÓN

La creación de aplicaciones de red social *online* como Facebook, LinkedIn o Twitter durante la década de los 2000 y su posterior expansión han dado lugar a nuevas perspectivas y desafíos para campos como la recuperación de información y, como un caso particular, para el campo de la recomendación. Uno de los problemas de interés en este ámbito es el de recomendar personas con las que establecer lazos de amistad o con las que interactuar. Motivada por la naturaleza social de estas redes y la afluencia masiva de usuarios que acceden diariamente a las mismas, la recomendación de contactos ha atraído en los últimos años el interés de la industria [8] y de la comunidad investigadora [3,10]. Las principales plataformas ofrecen servicios de recomendación de usuarios desde finales de la década pasada, con sistemas como '*Who-to-Follow*' en Twitter o '*People you may know*' en Facebook y LinkedIn.

La recomendación de contactos representa un caso muy particular de la tarea de recomendación [20], por dos razones: en primer lugar, se trata de un problema exclusivo del dominio de las redes sociales; en segundo, aunque normalmente los usuarios y los ítems a recomendar son objetos separados, en esta ocasión ítems y usuarios son el mismo conjunto. Estas particularidades han dado pie a una gran variedad de algoritmos de recomendación de usuarios, desarrollados en campos como la ciencia de redes (*network science*) [14], la recomendación clásica [10], el aprendizaje automático [15] o, en mucha menor medida, la recuperación de información [10].

En este trabajo nos centramos en esta última línea: investigamos la relación entre el problema de recomendar contactos en redes sociales y la recuperación de información orientada a la búsqueda de documentos de texto. Para ello, establecemos asociaciones entre los elementos que participan en las dos respectivas tareas, para adaptar los modelos clásicos de IR a la tarea de sugerir usuarios en una red social. En particular, exploramos la adaptación de los algoritmos de IR más comunes, en concreto el modelo vectorial [19], BM25 [18] y *query likelihood* [17]. Finalmente, comparamos empíricamente la efectividad de dichas adaptaciones con otros algoritmos de recomendación de contactos sobre diversas muestras de datos obtenidas de la red social Twitter, y constatamos que los modelos de IR son efectivos recomendando contactos.

Este trabajo se estructura como sigue: en la sección 2, resumimos brevemente el trabajo relacionado en las áreas relevantes para el problema que abordamos. En la sección 3, formalizamos la tarea de recomendación de contactos y la notación que emplearemos a lo largo del documento. A continuación, describimos la adaptación de algoritmos de IR a la tarea de sugerir usuarios (sección 4), y aplicamos esta idea desarrollando diferentes algoritmos basados en IR (sección 5). En la sección 6 presentamos nuestros experimentos y los resultados obtenidos. Finalmente, exponemos nuestras conclusiones y proponemos futuras líneas de investigación.

2 TRABAJO RELACIONADO

La recomendación de contactos tiene como objetivo identificar y sugerir usuarios que puedan ser de interés para otros en una red social. Las soluciones más comunes proceden de un problema conocido como la predicción de enlaces [14]: se trata de una tarea muy relacionada –en cierta medida equivalente– que, utilizando las propiedades topológicas de la red, busca identificar qué enlaces aparecerán en la red en el futuro. Además, se han desarrollado métodos específicos para recomendar usuarios, basados en paseos aleatorios [3,8], o contenido generado por el usuario [10].

CERI 18, June 26–27, 2018, Zaragoza, Spain

ACM ISBN 978-1-4503-6543-7/18/06…$15.00
https://doi.org/10.1145/3230599.3230619

En este trabajo, investigamos la ampliación de esta colección de algoritmos mediante la adaptación de modelos clásicos de IR a la tarea de recomendación de contactos. La conexión entre recomendación y recuperación de información se remonta a los inicios de los sistemas recomendación, y su relación con la tarea llamada "filtrado de información" [5]. Aunque gran parte de esta conexión se ha enfocado hacia la creación de algoritmos basados en contenido [1], también ha dado lugar al desarrollo de algoritmos basados en filtrado colaborativo [6,21].

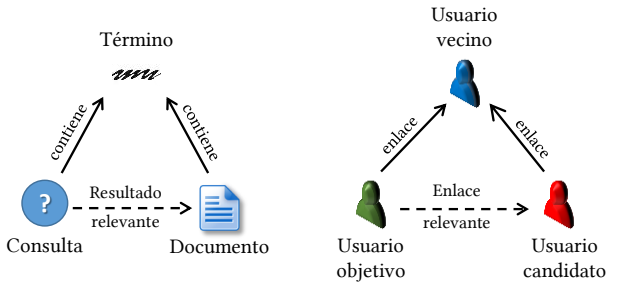
Una propuesta representativa y relevante para nuestro trabajo es la de Bellogín et al. [6], en la que se plantea una aproximación que permite aplicar cualquier método de ponderación de términos en IR al desarrollo de un algoritmo de recomendación basado en filtrado colaborativo. La propuesta de Bellogín et al. representa a usuarios e ítems en un espacio común, de forma que los usuarios (y sus preferencias) puedan jugar el papel de consultas y los ítems actúan como los documentos a recuperar. En nuestro trabajo buscamos una meta similar, dando un paso más allá: si Bellogín et al. pliegan tres espacios (términos, documentos, consultas) en dos (usuarios, ítems), nosotros los plegaremos los tres espacios en uno, como veremos.

Algunos autores han conectado asimismo técnicas de IR con la tarea específica de recomendación de contactos. Por ejemplo, algunos algoritmos de predicción de enlaces, como los basados en Jaccard [12, 19], que se han utilizado para sugerir usuarios, tienen sus raíces en IR. Más recientemente, Hannon et al. [10] propusieron un sistema que permite aplicar el modelo vectorial [19] para recomendar usuarios en Twitter, con métodos basados tanto en contenido como en filtrado colaborativo. Nuestro trabajo busca ampliar, generalizar y sistematizar este punto de vista para adaptar cualquier algoritmo del estado del arte de la recuperación de información.

3 PRELIMINARES

En primer lugar, definimos la tarea de recomendación de contactos y la notación que utilizaremos a lo largo de este documento. Dada una red social, es posible representar su estructura como un grafo $\mathcal{G} = \langle \mathcal{U}, E \rangle$, donde $\mathcal{U}$ representa el conjunto de usuarios de la red social, y $E \subset \mathcal{U}_*^2$ representa las relaciones existentes entre usuarios, como puedan ser relaciones de amistad, interacciones entre ellos, etc., donde $\mathcal{U}_*^2 = \{(u, v) \in \mathcal{U}^2 | u \neq v\}$ representa el conjunto de pares de usuarios distintos. Para cada usuario $u \in \mathcal{U}$ definimos su vecindario (el conjunto de usuarios con los que ha establecido relaciones) como $\Gamma(u)$. En el caso de redes dirigidas, distinguiremos entre tres posibles vecindarios: el vecindario entrante (es decir, aquellos usuarios que crean relaciones hacia el usuario u) como $\Gamma_{\text{in}}(u)$, el vecindario saliente (aquellos usuarios hacia los que el usuario u establece enlaces) como $\Gamma_{\text{out}}(u)$, y la unión de ambas como $\Gamma_{\text{und}}(u)$.

Entonces, para un usuario individual u, la tarea de recomendar contactos consiste en hallar un subconjunto de los usuarios hacia los que u no tiene ningún enlace, $\hat{\Gamma}_{\text{out}}(u) \subset \mathcal{U} \setminus \Gamma_{\text{out}}(u)$, que puedan ser de su interés. Para ello, se plantea la recomendación como un problema de ranking, en el que buscamos hallar un número fijo n de usuarios que maximicen una determinada función de ranking $f_u : \mathcal{U} \setminus \Gamma_{\text{out}}(u) \to \mathbb{R}$.



(a) Tres espacios en la recuperación de texto

(b) Espacio único en la recomendación de contactos

Figura 1. Relación entre elementos de la tarea de IR (a) y la de recomendación de contactos (b).

4 RECOMENDACIÓN DE CONTACTOS MEDIANTE MODELOS DE IR

La recomendación es uno de los múltiples problemas que estudia la recuperación de información, al igual que la búsqueda basada en texto, que representa el problema más clásico en dicho campo. Si bien la recomendación y la búsqueda se han estudiado como problemas separados, es posible establecer analogías y equivalencias entre ambas tareas. Por ejemplo, es posible definir la tarea de recomendación como una tarea de búsqueda en la que la consulta no existe (al menos, de forma explícita), y, en su lugar, se utilizan registros de la actividad pasada del usuario sobre el sistema.

4.1 Formulación unificada

Para formular la recomendación de contactos como una tarea de IR, establecemos equivalencias entre los elementos que aparecen en la tarea de recomendación (usuarios y las interacciones entre ellos) con los espacios asociados a la tarea de búsqueda textual (consultas, documentos y términos). En anteriores adaptaciones de modelos de IR para tareas de recomendación, habitualmente se han proyectado los tres espacios de IR en dos: el conjunto de usuarios y el conjunto de ítems [6]. En el caso particular de la recomendación de usuarios, ambos espacios son el mismo. Por tanto, para adaptar los modelos de IR para la dicha tarea, se proyectan en una única dimensión: el conjunto de usuarios en la red social, jugando tres papeles diferentes, tal como ilustramos en la Figura 1.

En primer lugar, asociamos los documentos con los usuarios candidatos de la recomendación, dado que ambos elementos representan un papel similar en sus respectivas tareas: son los elementos que han de ser identificados y recuperados para satisfacer una determinada necesidad. En cuanto a la necesidad de información, en IR tiene una expresión explícita representada por la consulta, mientras que en la tarea de recomendación la necesidad es implícita. Puesto que en los sistemas de recomendación se asume que los gustos presentes de los usuarios se manifiestan en sus elecciones y preferencias pasadas, cabe tomar éstas como una representación de la necesidad de información. Dichas preferencias se modelan en forma de un perfil para cada usuario en la red, que puede asimilarse al papel de consulta. Como este perfil no es más que una representación del usuario al que se le van a sugerir posibles enlaces, el ele-

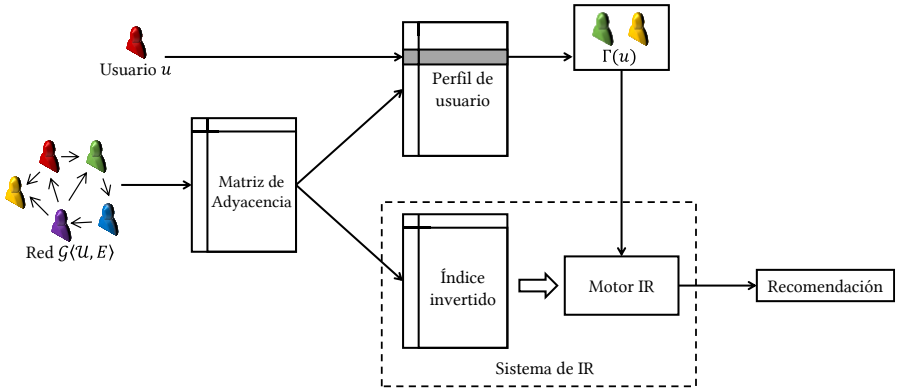


Figura 2. Adaptación de modelos de IR para la recomendación de contactos.

mento que identificamos finalmente con la consulta es el usuario que va a recibir la recomendación, conocido como usuario objetivo.

Por último, es necesario definir qué elementos van a jugar el papel de los términos. En el problema general de recomendar ítems, este punto es el que representa una mayor dificultad: en dicho caso, usuarios e ítems existen en espacios diferentes, por lo que una correspondencia para los términos válida para los usuarios podría no ser adecuada para los ítems, y viceversa [6]. En el caso de la recomendación de contactos, este problema desaparece, puesto que no hay diferencia entre usuarios e ítems: cualquier espacio de representación (haciendo el papel de términos) que funcione para los usuarios objetivo funciona automáticamente para los ítems de esta recomendación, es decir, los usuarios candidatos.

Las posibilidades para definir un equivalente para el papel de los términos son innumerables, y dan lugar a algoritmos muy diferentes. Por ejemplo, si quisiéramos definir métodos de recomendación de contactos basados en contenido, una posibilidad pasa por utilizar textos asociados, de una u otra manera, a los usuarios en la plataforma de red social (por ejemplo, mensajes o documentos publicados o preferidos por los usuarios) [10]. Para enfocarnos, en nuestro caso, a algoritmos de filtrado colaborativo, los "términos" con los que se describen los perfiles de usuario se basan en la interacción de éstos con los ítems, esto es, con otros usuarios de la red social. Identificamos así la relación término-documentos clásica de IR con las interacciones usuario-usuario en la red social.

Una vez establecidas las asociaciones entre espacios, es posible aplicar algoritmos de IR para sugerir usuarios en redes sociales, como el que se muestra en la Figura 2: dado el grafo de la red social, extraemos las relaciones sociales utilizando su matriz de adyacencia A (una representación matricial del grafo, en la que $A_{uv} = 1$ si existe un enlace entre los usuarios u y v, y $A_{uv} = 0$ en otro caso). Mediante dichos enlaces, construimos dos elementos: por un lado, un índice invertido que permite recuperar a los usuarios candidatos, y, por otro, una estructura que permita obtener los vecindarios de los usuarios objetivo. En el caso del índice, sus claves son los identificadores de los usuarios de la red, y, para un usuario v, su lista de postings se corresponde con la lista de usuarios que lo incluyen en su vecindario. Entonces, utilizando este índice y la consulta, es posible utilizar cualquier motor de IR para recomendar usuarios en redes sociales.

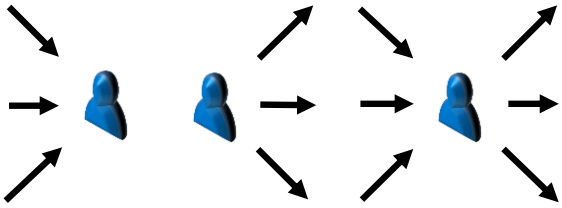
4.2 Selección de vecindarios

En función del tipo de red sobre el que se aplique la aproximación anterior, se presenta una cuestión adicional a establecer, debida a la representación de los usuarios en función de sus vecinos. En redes dirigidas como Twitter o Instagram, no existe una definición unívoca de vecindario de un usuario, sino que existen tres, como se ilustra en la Figura 3: el vecindario entrante $\Gamma_{\text{in}}(u)$ (usuarios que crean enlaces hacia u), el vecindario saliente $\Gamma_{\text{out}}(u)$ (usuarios hacia los que u crea enlaces), y la unión de ambos $\Gamma_{\text{und}}(u) = \Gamma_{\text{out}}(u) \cup \Gamma_{\text{in}}(u)$, es decir, los vecinos de u ignorando la dirección de los enlaces.

Cualquiera de las tres opciones es válida en nuestra adaptación de modelos. Dado que en nuestra aproximación el índice invertido y los perfiles de usuario son construidos de manera independiente, es posible definir, además, los usuarios objetivo y candidato en función de diferentes vecindarios. Puesto que en ambos casos seguimos utilizando los mismos elementos para representar consultas y documentos, sigue siendo posible equiparar las diferentes representaciones. Determinar qué vecindario caracteriza mejor a los usuarios candidato y objetivo en una red social es un problema interesante por sí mismo [10], y afecta a muchos otros algoritmos empleados para sugerir usuarios, como Adamic-Adar [14] que utilizan también los vecindarios de ambos usuarios en sus funciones de ranking. Por ello, exploraremos este problema en la sección de experimentos.

5 ADAPTACIÓN DE MODELOS DE IR

Considerando la aproximación descrita en la sección anterior, cabe considerar formulaciones alternativas a las originales para los diferentes modelos de IR, específicas y optimizadas para recomendar contactos. En esta sección, mostramos en detalle cómo se realiza dicha reformulación para dos algoritmos del estado del arte de IR: BIR y BM25 [18]. Además, se proporcionan las fórmulas para aplicar el modelo vectorial [19], el método basado en modelos de lenguaje *query likelihood* [17] y la similitud de Jaccard [12,19] a la tarea de recomendar usuarios. A lo largo de esta sección, denotaremos como $\Gamma^q(u)$ y $\Gamma^d(v)$ respectivamente el vecindario seleccionado para la consulta y para los documentos.



(a) Entrante Γ_{in} (b) Saliente Γ_{out} (c) No dirigido Γ_{und}

Figura 3. Posibles selecciones de vecindarios.

5.1 Binary Independent Retrieval

El modelo conocido como BIR (*binary independent retrieval*) [18] es el más simple de los modelos probabilísticos de IR. Bajo la suposición de que la frecuencia de los términos de un documento sigue una distribución de Bernoulli, BIR utiliza como función de ranking la probabilidad de relevancia del documento d para la consulta correspondiente q. Su formulación es la siguiente:

$$f_q(d) = \sum_{w \in d \cap q} \text{RSJ}(w)$$

donde $\text{RSJ}(w)$ representa la fórmula de Robertson-Spärck-Jones [18], que se define como:

$$\text{RSJ}(w) = \log \frac{|R_w|(|D| - |D_w| - |R| - |R_w|)}{(|R| - |R_w|)(|D_w| - |R_w|)} \quad (1)$$

donde R es el conjunto de documentos relevantes para la consulta, R_w es el conjunto de documentos relevantes que contienen el término w, D es la colección y D_w es el conjunto de documentos que contienen a w. En la mayoría de las ocasiones, no existe información a priori que permita conocer qué documentos son relevantes para la consulta, por lo que la aplicación directa de esta fórmula es imposible. Sin embargo, considerando que típicamente solo una pequeña fracción de los documentos disponibles es relevante para una consulta, es posible utilizar la siguiente aproximación:

$$\text{RSJ}(w) = \log \frac{|D| - |D_w| + 0.5}{|D_w| + 0.5}$$

Para adaptar este algoritmo a la tarea de recomendación de contactos, en primer lugar, sustituimos la consulta y el documento por los vecindarios seleccionados para los usuarios objetivo y candidato (respectivamente). Además, es necesario definir los valores $|D|$ y $|D_w|$. Puesto que los usuarios candidatos juegan el papel de documentos, D se define simplemente como el conjunto de usuarios en la red social, $\mathcal{U}$. Dado que los términos también se identifican con los usuarios, definimos D_w como el conjunto de usuarios candidatos en cuyos vecindarios aparece el usuario w. Este conjunto se corresponde con el vecindario "inverso" de w, es decir, el vecindario de dicho usuario en la red seleccionando la dirección opuesta a la seleccionada para los usuarios candidatos (por ejemplo, si tomamos el vecindario saliente para representar a los posibles nodos a recomendar, utilizaríamos el vecindario entrante de w). Denotando dicho vecindario como $\Gamma^d_{\text{inv}}(w)$, y utilizando todo lo anterior, la adaptación de BIR para sugerir contactos es:

$$f_u(v) = \sum_{w \in \Gamma^q(u) \cap \Gamma^d(v)} \text{RSJ}(w)$$

$$\text{RSJ}(w) = \log \frac{|\mathcal{U}| - |\Gamma^d_{\text{inv}}(w)| + 0.5}{|\Gamma^d_{\text{inv}}(w)| + 0.5}$$

5.2 BM25

BM25 es uno de los modelos probabilísticos más conocidos y eficaces en IR [18]. Sigue los mismos principios que el algoritmo BIR, pero considera que la frecuencia de las palabras en los documentos sigue una distribución de Poisson, en lugar de una de Bernoulli. Su función de ranking es la siguiente:

$$f_q(d) = \sum_{w \in d \cap q} \frac{(k+1)\text{freq}(w,d)\text{RSJ}(w)}{k(1 - b + b|d|/\text{avg}_{d'}(|d'|)) + \text{freq}(w,d)}$$

donde $\text{freq}(w,d)$ denota la frecuencia del término w en el documento d, $|d|$ es la longitud de documento, $\text{RSJ}(w)$ está definida en la ecuación 1, y $k \in [0,\infty)$ y $b \in [0,1]$ son dos parámetros libres por configurar. El parámetro k controla el efecto de la frecuencia de los términos en el documento, y el parámetro b la influencia de la longitud del documento. A diferencia del algoritmo anterior, BM25 tiene pues en consideración la frecuencia de las palabras en los documentos, así como la longitud de los mismos.

De nuevo, aplicando las relaciones entre espacios y estas fórmulas, es posible representar este algoritmo como un modelo para la recomendación de personas. Además de intercambiar el documento y la consulta por los perfiles de los usuarios candidato y objetivo, es necesario definir tanto la longitud de los documentos como la frecuencia de los términos. Dado que trabajamos con grafos sin pesos, un usuario aparece en un vecindario, como máximo, una vez, por lo que la frecuencia se puede definir como:

$$\text{freq}(w,v) = 1_{\Gamma^d(v)}(w) = \begin{cases} 1 & \text{si } w \in \Gamma^d(v) \\ 0 & \text{en otro caso} \end{cases}$$

El caso de la longitud de documento es más complejo. De acuerdo con la formulación original del algoritmo, existen varias posibilidades para medir la longitud de los documentos (número de palabras diferentes, suma de las frecuencias,...). En nuestro caso, seleccionamos el número de vecinos del usuario candidato, utilizando cualquiera de las posibles definiciones de vecindario (entrante, saliente o unión de ambos). Aunque la selección más inmediata para dicho vecindario pasa por utilizar el mismo definido anteriormente, en nuestra adaptación, permitimos el uso de cualquiera de ellos, siendo esta selección de vecindario un nuevo parámetro libre que ha de ser optimizado. Representamos este vecindario como $\Gamma^l(v)$.

Teniendo en cuenta todo lo anterior, definimos la función de ranking para BM25 como:

$$f_u(v) = \sum_{w \in \Gamma^q(u) \cap \Gamma^d(v)} \frac{(k+1)\text{RSJ}(w)}{k(1 - b + b|\Gamma^l(v)|/\text{avg}_{v'}(|\Gamma^l(v')|) + 1}$$

Es fácil observar que, si el parámetro b es idénticamente 0, el algoritmo es equivalente a BIR.

Tabla 1. Adaptación de modelos de IR para la recomendación de contactos.

Modelo	Función de ranking para IR $f_q(d)$	Función de ranking para recomendar contactos $f_u(v)$
VSM	$\sum_{w\in q\cap \mathrm{d}} q_w d_w / \sqrt{\sum_{\mathrm{w}\in \mathrm{d}} d_w^2}$	$\sum_{w\in \Gamma^{\mathrm{q}}(u)\cap\Gamma^{\mathrm{d}}(v)} u_w v_w / \sqrt{\sum_{\mathrm{w}\in\Gamma^{\mathrm{d}}(v)} v_w^2}$
	Esquema de pesos: tf $d_w = (1+\log \mathrm{freq}(\mathrm{w},\mathrm{d}))\cdot \mathbb{1}_{\mathrm{d}}(w)$; tf-idf $d_w = \mathrm{tf}(w,d)\log\left(1+\frac{\lvert D\rvert}{1+\lvert D_w\rvert}\right)$	Esquema de pesos: tf $u_w = \mathbb{1}_{\Gamma(u)}(w)$; tf-idf $u_w = \mathbb{1}_{\Gamma(u)}(w)\log\left(1+\frac{\lvert U\rvert}{1+\lvert\Gamma_{\mathrm{inv}}(w)\rvert}\right)$
BIR	$\sum_{w\in q\cap \mathrm{d}} \mathrm{RSJ}(w)$ $\quad \mathrm{RSJ}(w) = \log\frac{\lvert D\rvert-\lvert D_w\rvert-0.5}{\lvert D_w\rvert-0.5}$	$\sum_{w\in\Gamma^{\mathrm{q}}(u)\cap\Gamma^{d}(v)} \mathrm{RSJ}(w)$ $\quad \mathrm{RSJ}(w) = \log\frac{\lvert U\rvert-\lvert\Gamma^{\mathrm{d}}_{\mathrm{inv}}(w)\rvert-0.5}{\lvert\Gamma^{\mathrm{d}}_{\mathrm{inv}}(w)\rvert-0.5}$
BM25	$\sum_{w\in q\cap d}\frac{(k+1)\,\mathrm{freq}(w,d)\,\mathrm{RSJ}(w)}{k\left(1-b+b\lvert d\rvert/\mathrm{avg}_{d'}\lvert \mathrm{d}'\rvert\right)+\mathrm{freq}(w,d)}$	$\sum_{w\in\Gamma^{\mathrm{q}}(u)\cap\Gamma^{\mathrm{d}}(v)}\frac{(k+1)\,\mathrm{RSJ}(w)}{k\left(1-b+b\lvert\Gamma^{l}(v)\rvert/\mathrm{avg}_{v'}\lvert\Gamma^{l}(v')\rvert\right)+1}$
Extreme BM25	$\sum_{w\in q\cap d}\frac{\mathrm{freq}(w,d)\,\mathrm{RSJ}(w)}{1-b+b\lvert d\rvert/\mathrm{avg}_{d'}\lvert d'\rvert}$	$\sum_{w\in\Gamma^{\mathrm{q}}(u)\cap\Gamma^{\mathrm{d}}(v)}\frac{\mathrm{RSJ}(w)}{1-b+b\lvert\Gamma^{l}(v)\rvert/\mathrm{avg}_{v'}\lvert\Gamma^{l}(v')\rvert}$
QLJM	$\sum_{w\in q}\log\left(\frac{(1-\lambda)}{\lvert d\rvert}\mathrm{freq}(w,d)+\lambda\frac{\lvert d\rvert}{\sum_{d'\in D}\lvert d'\rvert}\right)$	$\sum_{w\in\Gamma^{\mathrm{q}}(u)}\log\left(\frac{(1-\lambda)}{\lvert\Gamma^{\mathrm{d}}(v)\rvert}\mathbb{1}_{\Gamma^{\mathrm{d}}(v)}(w)+\lambda\frac{\lvert\Gamma^{\mathrm{d}}_{\mathrm{inv}}(w)\rvert}{\lvert E\rvert}\right)$
Jaccard	$\lvert d\cap q\rvert/\lvert d\cup q\rvert$	$\lvert\Gamma^{\mathrm{q}}(u)\cap\Gamma^{\mathrm{d}}(v)\rvert/\lvert\Gamma^{\mathrm{q}}(u)\cup\Gamma^{\mathrm{d}}(v)\rvert$

5.3 Extreme BM25

Como método novedoso basado en IR, proponemos un algoritmo basado en BM25. Denominamos este algoritmo como *extreme* BM25, puesto que se trata de una versión extrema de la fórmula anterior, en la que el parámetro k tiende a infinito. Se define como:

$$f_u(v) = \sum_{w\in\Gamma^{q}(u)\cap\Gamma^{d}(v)} \frac{\mathrm{RSJ}(w)}{1-b+b\lvert\Gamma^{l}(v)\rvert/\mathrm{avg}_{v'}(\lvert\Gamma^{l}(v')\rvert)}$$

Esta formulación tiene la ventaja de disponer de un parámetro libre menos, lo que simplifica su configuración.

5.4 Otros modelos de IR

Además de los algoritmos presentados hasta ahora, mostramos en la Tabla 1 la adaptación de otros modelos de recuperación de información para recomendar usuarios. Utilizando las técnicas empleadas para BIR y BM25, hemos adaptado el modelo vectorial (VSM) [19] utilizando dos esquemas diferentes para los pesos: a) sólo tf, y b) tf-idf. En la Tabla 1 se muestran los detalles para ambos. Aparte, adaptamos el modelo de lenguaje conocido como *query likelihood* (QL) [17] utilizando el suavizado de Jelinek-Mercer [13]. Finalmente, mostramos como quedaría la similitud de Jaccard [12,19] que se emplea tradicionalmente en IR para la comparación de documentos, y es uno de los métodos más conocidos de predicción de enlaces [14]. Se incluyen en dicha tabla los algoritmos BIR, BM25 y extreme BM25 por compleción.

6 EXPERIMENTOS

Con el objetivo de probar empíricamente la efectividad de los algoritmos basados en IR en la tarea de recomendar usuarios en redes sociales, proponemos un experimento de evaluación *offline* sobre datos de la red social Twitter. En esta sección, mostramos la configuración de dicho experimento, así como los resultados obtenidos para el mismo.

6.1 Conjuntos de datos

En nuestro experimento, comparamos el rendimiento de los algoritmos de recomendación sobre dos muestras diferentes de la red social Twitter. Para cada muestra, consideramos dos grafos: un grafo explícito de seguimiento (*follows*), donde $(u,v)\in E$ si u sigue a v en la red social; y un grafo dinámico, obtenido a partir de las interacciones entre usuarios, en el que $(u,v)\in E$ si u ha retweeteado un tweet creado por el usuario v, o bien ha mencionado o respondido a v en un tweet.

Partiendo de un usuario semilla, utilizamos la API REST de Twitter para obtener nuestros grafos. En primer lugar, obtenemos el grafo de interacciones, utilizando un procedimiento similar al conocido como muestreo de bola de nieve (en inglés, *snowball sampling*) [9], que permite aprovechar las capacidades de dicha API: para cada usuario explorado, obtenemos un conjunto de tweets, y extraemos de los mismos los enlaces de interacción que crea el usuario. El conjunto de personas con los que interacciona en dicho conjunto constituye el vecindario saliente de dicho usuario. Probamos dos opciones diferentes para la selección del conjunto de tweets:

1. El conjunto de tweets publicados por el usuario en un intervalo de tiempo determinado (en nuestros experimentos, 1 mes, desde el 19 de junio al 19 de julio de 2015).
2. Los n tweets más recientes de un usuario publicados previamente a una fecha dada (en nuestro caso, la fecha límite escogida fue el 2 de agosto de 2015, y $n = 200$).

En el resto de esta sección, nos referiremos al conjunto de datos recuperado mediante la primera opción como "1 mes", y al otro como "200 tweets".

Una vez recuperado un usuario, se procede a visitar el resto de los usuarios utilizando una cola FIFO (*first in, first out*). El procedimiento continuará hasta que hayamos visitado un determinado número de usuarios (10,000 en nuestros experimentos). Las muestras estarán formadas por el conjunto de usuarios visitados, y los

Tabla 2. Descripción de los datos del experimento

		#Enlaces	
Grafo	#Usuarios	Entrenamiento	Test
Interacciones 1 mes	9,528	170,425	57,846
Follows 1 mes	9,861	630,504	13,766
Interacciones 200 tweets	9,985	137,850	21,598
Follows 200 Tweets	9,964	427,568	46,760

enlaces serán las interacciones que hayamos detectado entre ellos. Una vez concluya la construcción de dicho grafo, es posible obtener el grafo de *follows*, de nuevo utilizando la API REST de Twitter. Esta red captura todas las relaciones de seguimiento entre los usuarios recuperados mediante la descarga anterior.

6.2 Metodología de evaluación

Para evaluar los algoritmos, dividimos las redes sociales en un grafo de entrenamiento y otro de test, mediante un corte temporal: todos los enlaces creados antes de la fecha indicada forman el conjunto de entrenamiento, y todos los posteriores a la misma el conjunto de test. En el caso de la red de interacción de "1 mes", tomamos las interacciones definidas en los tweets publicados durante las tres primeras semanas (hasta el 12 de julio) como entrenamiento, y el resto de interacciones como conjunto de test.

En el conjunto de "200 tweets", el grafo de entrenamiento comprende las interacciones recogidas en el primer 80% de los tweets. Si un enlace aparece en ambos conjuntos es eliminado de la red de test. Para las redes de seguimiento explícito, puesto que la API de Twitter no proporciona información temporal sobre la creación de enlaces, se realizó una segunda descarga de los enlaces 4 meses más tarde, de tal forma que la primera descarga forma el grafo de entrenamiento, y los nuevos enlaces que se obtuvieron en la segunda descarga el conjunto de test. En la Tabla 2 se muestran el número de nodos y de enlaces para cada uno de los grafos.

A partir de dichas particiones, es posible aplicar métricas de IR como precisión, *recall* o nDCG [4]. Para ello, basta con considerar que un usuario v es relevante para el usuario u si el enlace (u, v) aparece en el grafo de entrenamiento. En otro caso, v será considerado "no relevante" para u. Si un usuario no tiene datos de test (es decir, no ha creado ningún nuevo enlace en el período de test), lo excluimos del cálculo de la métrica, ya que ningún algoritmo será capaz de obtener un acierto mayor que cero (y por tanto ese usuario no ayuda a comparar algoritmos).

Una consideración adicional a la hora de generar las recomendaciones es que eliminamos del conjunto de usuarios candidatos a aquellos usuarios que ya dispongan de un enlace hacia el nodo objetivo – es decir, no recomendamos enlaces recíprocos. La razón es la siguiente: Twitter notifica a los usuarios sobre cualquier acción que alguien lleve a cabo en la red que les afecte, incluyendo cada vez que otro usuario les sigue o interactúa con ellos – por lo que, en la práctica, Twitter ya ha "recomendado" a dichos usuarios". Por ello, sería redundante incluirlos en nuestras recomendaciones, y, más importante, podría introducir un sesgo en la recomendación, dado que los enlaces recíprocos tendrían mayores posibilidades de aparecer en el conjunto de test.

Tabla 3. Parámetros óptimos para los diferentes algoritmos y conjuntos de datos. La selección óptima de direccionalidad para Γ^q y Γ^d para cada algoritmo se puede observar en la Figura 4. En Adamic-Adar, indicamos Γ^l para representar la dirección en la selección de los vecinos comunes entre el usuario objetivo y el candidato (ver [20]).

	1 mes		200 tweets	
Algoritmo	Inter.	Follows	Inter.	Follows
BM25	$k = 1$ $b = 0.1$ $\Gamma^l = \Gamma_{\text{out}}$	$k = 1{,}000$ $b = 0.999$ $\Gamma^l = \Gamma_{\text{out}}$	$k = 1$ $b = 0.3$ $\Gamma^l = \Gamma_{\text{out}}$	$k = 1$ $b = 0.1$ $\Gamma^l = \Gamma_{\text{und}}$
Extreme BM25	$b = 0.1$ $\Gamma^l = \Gamma_{\text{out}}$	$b = 0.999$ $\Gamma^l = \Gamma_{\text{out}}$	$b = 0.1$ $\Gamma^l = \Gamma_{\text{out}}$	$b = 0.1$ $\Gamma^l = \Gamma_{\text{out}}$
QL	$\lambda = 0.1$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.1$
Money	$\alpha = 0.99$	$\alpha = 0.1$	$\alpha = 0.99$	$\alpha = 0.99$
Adamic-Adar	$\Gamma^l = \Gamma_{\text{und}}$	$\Gamma^l = \Gamma_{\text{in}}$	$\Gamma^l = \Gamma_{\text{und}}$	$\Gamma^l = \Gamma_{\text{und}}$
PageRank pers.	$r = 0.4$	$r = 0.5$	$r = 0.4$	$r = 0.8$
MF	$k = 260$ $\alpha = 40$ $\lambda = 150$	$k = 20$ $\alpha = 40$ $\lambda = 150$	$k = 300$ $\alpha = 40$ $\lambda = 150$	$k = 300$ $\alpha = 40$ $\lambda = 150$
UB-kNN	$k = 120$	$k = 80$	$k = 100$	$k = 40$
IB-kNN	$k = 300$	$k = 240$	$k = 290$	$k = 300$

6.3 Algoritmos de recomendación

Finalmente, además de los modelos de IR, para nuestros experimentos hemos seleccionado un conjunto de algoritmos de recomendación de contactos que se encuentran entre los más representativos y/o efectivos en la literatura. Estos incluyen Adamic-Adar, el mayor número de vecinos comunes (MCN), una versión personalizada de PageRank [14], kNN basado en usuarios e ítems [1,16] y el algoritmo de factorización de matrices propuesto por Hu et al. [11]. Además, aplicamos una versión del algoritmo Money [8] propuesto por Twitter, en la que, por simplicidad, incluimos a todos los usuarios de la red en el círculo de confianza definido en dicho trabajo. Optimizamos los diferentes parámetros de todos los algoritmos aplicando una *grid search* orientada a maximizar el acierto de los algoritmos (P@10). Mostramos los valores de los parámetros en la Tabla 3.

6.4 Resultados

En la Tabla 4 mostramos los valores de P@10 para los diferentes algoritmos evaluados. En dicha tabla, podemos observar una comparativa de algoritmos similar entre tres de las cuatro redes analizadas: los grafos de interacción para "1 mes" y "200 tweets", y el grafo de *follows* del segundo conjunto de datos. En dichas redes, la popularidad obtiene resultados bajos, y destacan, en general, las aproximaciones propias de la recomendación clásica, el algoritmo Money, Adamic-Adar y los modelos probabilísticos de IR como BIR, BM25 o *extreme* BM25. En el grafo restante, la popularidad predomina sobre la gran mayoría de algoritmos, con la excepción de los algoritmos BM25 y *extreme* BM25.

Tabla 4. Valores de P@10 para los diferentes algoritmos. Cada columna representa un grafo diferente, y, en negrita, se marca el mejor algoritmo para cada conjunto. En cada columna, aplicamos una escala de color que va desde blanco (peor algoritmo) hasta azul (mejor algoritmo).

		1 mes		200 tweets	
	Algoritmo	Inter.	Follows	Inter.	Follows
IR	BM25	0.0623	**0.0110**	**0.0546**	0.0468
IR	Extreme BM25	0.0605	**0.0110**	0.0542	0.0460
IR	BIR	0.0675	0.0024	0.0535	0.0462
IR	QL	0.0580	0.0012	0.0476	0.0473
IR	Jaccard	0.0226	0.0006	0.0304	0.0343
IR	VSM tf	0.0186	0.0007	0.0253	0.0337
IR	VSM tf-idf	0.0185	0.0007	0.0268	0.0334
Recom. contactos	Money	0.0772	0.0022	0.0477	0.0435
Recom. contactos	Adamic-Adar	0.0676	0.0026	0.0533	0.0470
Recom. contactos	MCN	0.0638	0.0024	0.0507	0.0456
Recom. contactos	PageRank pers.	0.0598	0.0032	0.0336	0.0351
CF	MF	**0.0837**	0.0019	0.0542	**0.0541**
CF	UB-kNN	0.0810	0.0012	0.0479	0.0504
CF	IB-kNN	0.0738	0.0005	0.0361	0.0475
Otros	Popularity	0.0313	0.0045	0.0225	0.0168
Otros	Random	0.0007	0.0000	0.0004	0.0006

Cabe, por tanto, destacar la competitividad de los métodos probabilísticos de IR, que introducen al menos una de sus variantes (ya sea BIR, BM25 o *extreme* BM25) entre los 6 mejores algoritmos (de 16), e incluso son los mejores en dos de las redes sobre las que hemos realizado los experimentos: la red de *follows* para el conjunto de "1 mes", y la red de interacción para el conjunto de "200 tweets". De entre todas las variantes, BM25 es la más destacada, siendo la mejor en tres de las cuatro redes. En cuanto al resto, *query likelihood* obtiene valores de precisión ligeramente por debajo (con la excepción del grafo de *follows* de 200 tweets). Y mucho más lejos aparecen los modelos más clásicos, como la similitud de Jaccard o ambas variantes del modelo vectorial que muestran resultados similares a la popularidad (e incluso por debajo).

Respecto al resto de algoritmos, los algoritmos basados en vecindario y especialmente, factorización de matrices son muy efectivos para recomendar contactos (a excepción de la comparativa en la red de *follows* de "1 mes"). Adamic-Adar y MCN proporcionan valores de precisión similares a BIR y la versión personalizada de PageRank se queda, en general, por debajo de ellos.

Selección de vecindario. Además de los algoritmos de IR, otros algoritmos de los estudiados en nuestra comparativa utilizan como base la intersección entre los vecindarios de los usuarios objetivo y candidato. Este es el caso de MCN y Adamic-Adar. Un problema interesante es determinar qué vecindario representa mejor a los usuarios objetivos y a los usuarios candidatos. Para ello, probamos cómo varía P@10 en función de que vecindario seleccionamos para ambos usuarios (empleando, en el resto de hiperparámetros, la configuración escogida para la mejor versión del algoritmo). Esta comparativa se muestra en la Figura 4 para los diferentes grafos estudiados. En dichas figuras, las etiquetas más exteriores del eje x se refieren a la selección de vecindario para el usuario objetivo, y las más interiores a la selección del usuario candidato.

Si observamos la selección de vecindarios por separado, vemos que, en general, se repite el mismo esquema en los diferentes grafos que estudiamos: en el caso de los usuarios candidatos, la peor elección pasa por escoger su vecindario saliente: en todos los casos, seleccionar este vecindario produce los peores resultados. Por otro lado, los mejores resultados se producen en la mayoría de las ocasiones cuando se selecciona su vecindario entrante. En el caso del usuario objetivo, de nuevo, la peor elección es utilizar su vecindario saliente, pero, entre las dos opciones restantes, no queda claro cuál es la alternativa (el vecindario entrante o el vecindario no dirigido) que caracteriza mejor a dicho usuario.

7 CONCLUSIONES

Aunque inicialmente se estudiaron de forma separada, la búsqueda de texto y la recomendación son tareas muy relacionadas. Esta relación se ha explorado anteriormente de forma general hacia la adaptación de modelos de IR para la recomendación de ítems [6,21]. En el presente trabajo particularizamos este paso a la recomendación de contactos en redes sociales. Nuestra investigación encuentra que la adaptación de modelos de IR da lugar igualmente a soluciones empíricamente efectivas, y de hecho más simples, en cierta medida, que las previamente establecidas para el caso general de la recomendación de ítems.

Varios modelos de IR se han mostrado competitivos en términos de acierto en nuestros experimentos, en comparación con una selección de algoritmos de recomendación de contactos del estado del arte, tras aplicarlos sobre datos de red social extraídos de Twitter. Este hecho es especialmente notorio en el caso del modelo probabilístico BM25 [18] y sus variantes. Además, la formulación de estos algoritmos ha permitido analizar qué vecindarios caracterizan mejor a los usuarios objetivo y candidato a la hora de recomendar contactos: el vecindario entrante en el caso de los usuarios candidatos, y el vecindario entrante o no dirigido para los usuarios objetivo.

Para concluir, puesto que hemos observado diferencias en los resultados para diferentes redes, nos planteamos analizar el efecto que puede tener la forma de obtener las muestras de grafos con respecto a la efectividad de los diversos algoritmos. Además, nos planteamos el estudio de la recomendación de contactos desde una perspectiva diferente: el análisis de nuevas dimensiones de evaluación alternativas al acierto, tales como la novedad y la diversidad de la recomendación [7], o los posibles efectos que pueden tener estos modelos sobre la evolución de la red [2].

AGRADECIMIENTOS

Este trabajo está financiado por el Ministerio de Economía y Competitividad (TIN2016-80630-P).

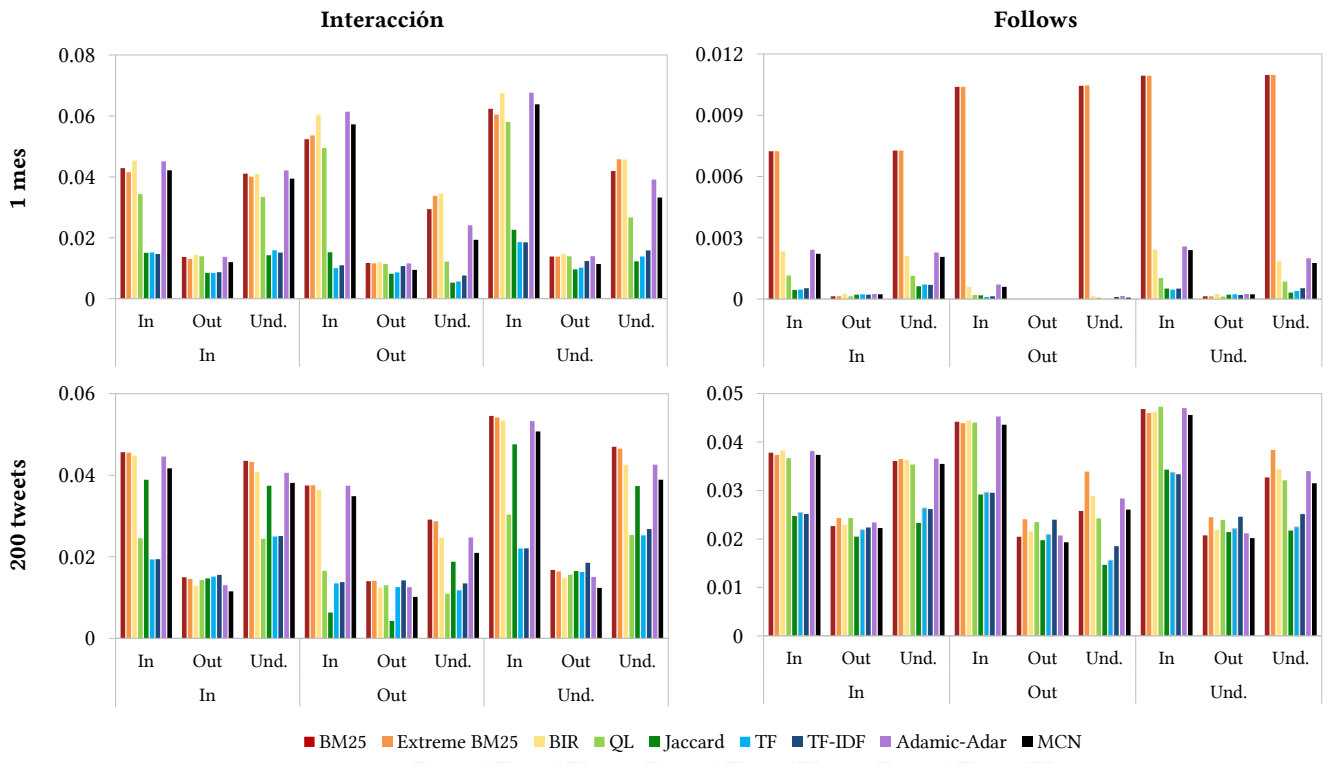


Figura 4. Resultados del experimento de direccionalidad. En todas las gráficas, el eje Y representa el valor de P@10. En el eje X, las etiquetas más exteriores se refieren al vecindario escogido para el usuario objetivo (In para el vecindario entrante, Out para el saliente y Und para el no dirigido), y las etiquetas más interiores, al vecindario para el usuario candidato.

REFERENCIAS

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generatio of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (April 2005), 734-749.

[2] Luca M. Aiello and Nicola Barbieri. 2017. Evolution of Ego-networks in Social Media with Link Recommendations. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining* (WSDM 2017). ACM, New York, NY, USA, 111-120.

[3] Lars Backstrom and Jure Leskovec. 2011. Supervised random walks: predictin g and recommending links in social networks. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining* (WSDM 2011). ACM, New York, NY, USA, 634-644.

[4] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 2011. *Modern Information Retrieval: The Concepts and Technology Behind Search* (2nd ed.). Addison-Wesley Publishing Company, USA.

[5] Nicholas J. Belkin and W. Bruce Croft. 1992. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM* 35, 12 (December 1992), 29-38.

[6] Alejandro Bellogín, Jung Wang and Pablo Castells. 2013. Bridging Memory-Based Collaborative Filtering and Text Retrieval. *Information Retrieval* 16, 6 (December 2013), 697-724.

[7] Pablo Castells, Neil Hurley and Saúl Vargas. 2015. Novelty and Diversity in Recommender Systems. In F. Ricci, L. Rokach, B. Shapira (Eds.), *Recommender Systems Handbook* (2nd ed.). Springer, New York, NY, USA,881-918.

[8] Ashish Goel, Pankaj Gupta, John Sirois, Dong Wang, Aneesh Sharma and Siva Gurumurthy. 2015. The who-to-follow system at Twitter: Strategy, algorithms and revenue impact. *Interfaces* 45, 1 (February 2015), 98-107.

[9] Leo A. Goodman. 1961. Snowball Sampling. *Annals of Mathematical Statistics* 31, 1 (March 1961), 148-170.

[10] John Hannon, Mike Bennet and Barry Smith. 2010. Recommending Twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the 4th ACM Conference on Recommender Systems* (RecSys 2010). ACM, New York, NY, USA, 199-206.

[11] Yifan Hu, Yehuda Koren and Chris Volinsky. 2008. Collaborative Filtering for Impicit Feedback Datasets. In *Proc. of the 8th IEEE Int. Conference on Data Mining* (ICDM 2008). IEEE Computer Society, Washington, DC, USA, 15-19.

[12] Paul Jaccard. 1901. Étude comparative de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, 142 (January 1901), 547-579.

[13] Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*. North-Holland, Amsterdam, The Netherlands.

[14] David Liben-Nowell and Jon Kleinberg. 2003. The link prediction problem for social networks. In P*roc. of the 12th International Conference on Information and Knowledge Management* (CIKM 2003). ACM, New York, NY, USA, 556-559.

[15] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem and Mohammed J. Zaki. 2006. Link Prediction Using Supervised Learning. In *Proceedings of the SDM 06 Workshop on Link Analysis, Counterterrorism and Security.*

[16] Xia Ning, Christian Desrosiers and George Karypis. 2015. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In F. Ricci, L. Rokach, B. Shapira (Eds.), *Recommender Systems Handbook* (2nd ed.). Springer, New York, NY, USA, 37-77.

[17] Jay M. Ponte and Bruce W. Croft. 1998. A language modelling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Search and Development in Information Retrieval* (SIGIR 1998). ACM, New York, NY, USA, 275-281.

[18] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (April 2009), 333-389.

[19] Gerald Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA.

[20] Javier Sanz-Cruzado and Pablo Castells. 2018. Contact Recommendations in Social Networks. In S. Berkovsky, I. Cantador, D. Tikk (Eds.), *Collaborative Recommendations: Algorithms, Practical Challenges and Applications.* World Scientific Publishing, Singapore.

[21] Daniel Valcarce. 2015. Exploring Statistical Language Models for Recommender Systems. In *Proceedings of the 9th ACM Conference on Recommender Systems* (RecSys 2015). ACM, New York, NY, USA, 375-378.