Expert Systems With Applications 180 (2021) 115028

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Session-based news recommendations using SimRank on multi-modal graphs

Panagiotis Symeonidis [c,*], Lidija Kirjackaja [b], Markus Zanker [a]

[a] Free University of Bolzano, Faculty of Computer Science, Bolzano 39100, Italy
[b] Vilnius Gediminas Technical University, Vilnius, Lithuania
[c] University of the Aegean, Samos, Greece

ARTICLE INFO

Keywords:
Graph-based news recommender systems
Time evolving graphs
Novel news article recommendations

ABSTRACT

Recommender systems are among the most widespread applications of artificial intelligence techniques. For instance, news recommender systems serve users in managing the overload of information they come across when accessing news portals. Obviously, in the news domain time-awareness of recommendation approaches are crucial. However, most of these approaches missed to consider user sessions, which group the items that a user interacted with. In this paper, we study the problem of session-based recommendations by running SimRank on time-evolving heterogeneous graphs. In particular, we construct a dynamic heterogeneous multi-partite graph and adjust SimRank to run on it by using different (i) sliding time window sizes, (ii) sub-graphs used for model learning and (iii) sequential article weighting strategies. We evaluate our algorithms on two real-life datasets, and we show that our method outperforms other state-of-the-art methods in terms of accuracy and diversity. The significance and impact of this work is important because it introduces to the research community of expert and intelligent systems, for the first time, a stream-based version of SimRank algorithm, which is able to run over time-evolving graphs.

1. Introduction

There exist many research works that strive to answer the question "what news article is a user going to click next given his profile". Content-based (CB) news recommendations build a user's profile based on the news articles that the user has read in the past, and recommend similar articles to him. Collaborative Filtering (CF) also helps in the exploration of news articles that other users similar to the target user have also read in the past. Recently, related work (Epure et al., 2017; Liu, Dolan, & Pedersen, 2010) has advanced the accuracy of news recommendations by considering the fact that individual and public preference evolves over time. For example, a user is a fan of basketball, but during the European football championship he may be also interested in news about football. However, although related work (Epure et al., 2017; Liu et al., 2010) considered the fact that user/public preferences over topic categories change over time, they missed to adequately exploit user sessions, which are short-time interactions of users with a system, and can reveal their very last and concrete intentions. In particular, these works take into account the time dimension to reveal users' preferences over time, but they miss to exploit adequately the information that is hidden inside user sessions, which include all the news articles that are viewed/read together by a user in a short time period (e.g. 30 min). (See Table 1).

In this paper, we exploit user sessions for providing personalised article recommendations in news industry domain. We build a multi-partite heterogeneous information network (i.e. user, article, category, etc.), that contains additionally a new type of an artificial node, called *session* node, which is associated with the co-click/co-view of two or more items in a short time period by a user. Then, we adjust SimRank to run on this dynamically time evolving graph, where we process it as a stream of data, and try to focus mostly on the users' last click behaviour, to reveal their last intentions. Thus, SimRank is used for searching relevance among items together with a sliding time window model, which captures the notion of recency of user's interest (i.e. short-term preference). In addition, we explore the effectiveness of different ways for building personalised recommendations for the "next item to be recommended" task.

The main contributions of this paper are summarized as follows:

* Corresponding author.
E-mail addresses: psymeon@aegean.gr (P. Symeonidis), mzanker@unibz.it (M. Zanker).

https://doi.org/10.1016/j.eswa.2021.115028
Received 29 November 2019; Received in revised form 30 March 2021; Accepted 9 April 2021
Available online 20 April 2021
0957-4174/

Table 1
SimRank similarity statements for different recommendation cases

Case	SimRank statement
Unipartite case	Two objects are similar if they are related to similar objects.
CF case	Two items are similar, if they are read by similar users, Two users are similar if they read similar items.
CB case	Two items are similar if they belong to similar categories, Two categories are similar if they have similar items
SB case	Two items are similar if they appear in similar sessions, Two sessions are similar if they consists of similar items.

- We adjust SimRank to run on a dynamic and time-evolving heterogeneous graph, which contains artificial **session** nodes.
- We identify important factors in news domain to improve article recommendations on two real-life datasets, in terms of their recommendation accuracy/diversity, under a **stream-based** evaluation protocol.
- We perform sensitivity analysis of SimRank to measure its effectiveness in recommendations when changing:
  1. the size of the training data by changing the size of a sliding time window, which disregards very old user-article interactions;
  2. the number of participating graph entities, by using different subgraphs for model learning;
  3. the way of capturing short user's interest by using different sequential article weighting strategies;

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 provides preliminaries for graphs and SimRank algorithm. Section 4 formulates the problem and describes our proposed algorithm. Experimental results are given in Section 5. Section 6 discusses some possible extensions of the proposed method. Finally, Section 7 concludes the paper.

2. Related work

In this paper, we focus on the graph-based models for the task of session-based news recommendation. One of the most popular algorithms of this category is proposed by Jeh and Widom (2002) and is denoted as SimRank. SimRank measures similarity of the structural context in which objects occur, based on their relationships with other objects. The measure can be used to measure similarities of nodes from one entity (e.g. similarities between users by considering their friends), and from two entities (e.g. similarities between users by considering items they interacted with).

For different recommendation approaches (e.g. Content-based (CB), Collaborative filtering (CF), Session-based (SB)), different general SimRank statements apply, though all corresponding to the underlying idea, that *two objects are similar if they are related to similar objects*. In Table 3 the SimRank statements for three CF approaches are presented.

A lot of research studies (Li et al., 2010; Lizorkin, Velikhov, Grinev, & Turdakov, 2008; Yu, Lin, & Zhang, 2013; Tao, Yu, & Li, 2014; Usman & Oseledets, 2015) tried to compute faster the SimRank results. For example, Yu et al. (2015) tried to find all partial pairs of SimRank and compute faster the similarity between these items. Another method is proposed by Kusumoto, Maehara, and Kawarabayashi (2014) who computes similarity scores between a query node and every other node.

Another graph-based algorithm or measuring similarity among entities is Random Walk with Restart (RWR) (Leskovec, Rajaraman, & Ullman, 2014). RWR tries to capture similarities among items or between users and items. The main different of RWR from PageRank is that the random walker re-starts his walk always from the initial node. Thus, RWR considers as more important the nearby nodes of the target node. To compute similarity in heterogeneous information networks, a graph-based algorithm, denoted as PathSim (Yizhou, Jiawei, Xifeng, & Tianyi, 2011), is proposed. PathSim runs on meta paths. A meta path is a sequence of different types of nodes and different types of links. Thus, each meta path carries a different semantic information. Finally, HeteSim (Shi, Kong, Huang, Philip, & Wu, 2014) is also a path-based measure. It is similar with SimRank algorihtm but it is path-constrained, which means that it uses only specific meta paths to measure similarity among nodes.

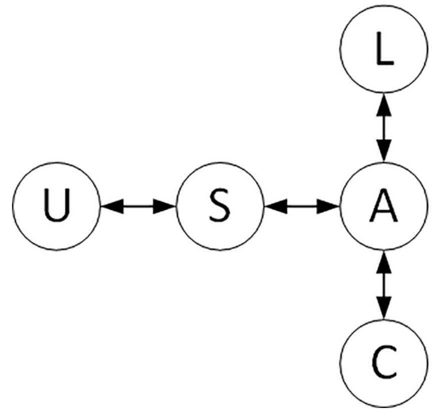

Fig. 1. Online News portal Network Schema.

In contrast to all the aforementioned methods which run mostly on static heterogeneous graphs, we combine SimRank together with a sliding time window model, which captures the notion of recency of user's interest (i.e. short-term preferences). That is, we use the idea of streaming graph data by using artificial session nodes, which are validated by a sliding time window. Session nodes and all their adjacent nodes that are not valid based on the time window are disregarded by our graph profile updater, as will be described later.

3. Preliminaries for graphs and SimRank

In this Section, we present some basic definitions for graphs and how SimRank is applied in different static heterogeneous graphs (i.e. unipartite, bipartite).

3.1. Heterogeneous information network

Definition 1. (*Information Network. (Yizhou et al., 2011)*) An information network is defined as a directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with an object type mapping function $\phi : \mathscr{V} \rightarrow \mathscr{Q}$ and a link type mapping function $\psi : \mathscr{E} \rightarrow \mathscr{R}$, where each object $v \in \mathscr{V}$ belongs to one particular object type $\phi(v) \in \mathscr{Q}$, and each link $e \in \mathscr{E}$ belong to a particular relation $\psi(e) \in \mathscr{R}$.

Differently from the traditional network definition, we explicitly distinguish object types and relationship types in the network. When the types of objects $|\mathscr{Q}| > 1$ or types of relations $|\mathscr{R}| > 1$, the network is called **heterogeneous information network**.

Example 1. A network for online news portal is a heterogeneous information network, containing objects from five types of entities $\mathscr{Q} = \{U, S, A, C, L\}$: users ($U$), sessions ($S$), articles ($A$), article categories ($C$), and article locations (L). Each user $u : \phi(u) = U$ has one or more links to the sessions $s : \phi(s) = S$, each session s has one unique user u associated with it and one or more articles $a : \phi(u) = A$ read by the user within the session. Finally, each article a can appear in one or more sessions s, belong to one news category $c : \phi(c) = C$ and be assigned to one geographic location $l : \phi(l) = L$.

3.2. Network schema

Definition 2. (*Network Schema. (Yizhou et al., 2011)*) The network schema is a meta template for a heterogeneous network $\mathscr{G} = (\mathscr{V}, \mathscr{E})$

with the object type mapping $\phi : \mathscr{V} \rightarrow \mathscr{O}$ and a link type mapping $\psi : \mathscr{E} \rightarrow \mathscr{R}$, which is a directed graph defined over object types $\mathscr{O}$, with edges as relations from $\mathscr{R}$, denoted as $T_G = (\mathscr{O}, \mathscr{R})$.

Network schema serves as a template for a network, and tells how many types of objects there are in the network and where the possible links exist. Network schema for online news portal is shown in Fig. 1.

3.3. SimRank and its extensions

In 2002, Jeh and Widom (2002) proposed a complementary approach in the problem of measuring similarity, applicable in any domain with entity-to-entity relationships. This measure is based on the idea that two entities are similar if they are referenced by similar entities. This general similarity measure is known as SimRank and it is based on a simple and intuitive graph-theoritic model.

In Eqs. (2) and (3), we can see how the SimRank algorithm is transformed for bipartite graphs. Let $s(a,b)$ denote the similarity between users a and b, and let s(c, d) denote the similarity between items c and d.

$$s(a,b) = \frac{C}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} s(O_i(a), O_j(b)) \tag{2}$$

$$s(c,d) = \frac{C}{|I(c)||I(d)|} \sum_{i=1}^{|I(c)|} \sum_{j=1}^{|I(d)|} s(I_i(c), I_j(d)) \tag{3}$$

Please notice that instead of implementing the two aforementioned formulas (i.e., Eqs. 2 and 3) and always checking which node type we are considering (users or items), it is enough to consider just the existence of a link: each link always connects only users with items, which means each link can be treated as outgoing for a user and as incoming for an item. This way we can express the aforementioned two formulas with only Eq. 4:

$$s(e,f) = \begin{cases} 1, & \text{if} e = f \\ \frac{C}{|N(e)||N(f)|} \sum_{i=1}^{|N(e)|} \sum_{j=1}^{|N(f)|} s(N_i(e), N_j(f)), & \text{if } e \neq f \text{and} (N(e) \neq \emptyset \text{or} N(f) \neq \emptyset) \\ 0, & \text{if } N(e) = \emptyset \text{or} N(f) = \emptyset, \end{cases} \tag{4}$$

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \tag{1}$$

In the above Equation, s(a,a) = 1 and $0 < C < 1$. C represents the degree of attenuation in similarity propagation.

Based on Eq. 1 to compute $s(a,b)$, we iterate over all in-neighbor pairs $(I_i(a), I_j(b))$ of (a,b), and sum up the similarity $s(I_i(a), I_j(b))$ of these pairs. Then we divide by the total number of in-neighbor pairs, $|I(a)||I(b)|$, to normalize. That is, the similarity between a and b is the average similarity between in-neighbors of a and in-neighbors of b. Eq. (1) is computed iteratively but it converges fast, in most cases usually after the fourth or fifth iteration.

The SimRank algorithm for bipartite user-item networks consists of two type of objects. That is, the similarity of items and the similarity of users are mutually-reinforcing notions:

– Users are similar if they purchase similar items.
– Items are similar if they are purchased by similar Users.

where $N(e), N(f)$ are the sets of neighbors of e and f respectively, and $C \in [0,1]$. That is, if a node is a user, its out-neighbors are the articles that he has interacted with, and if a node is an article, then its in-neighbors are the users who have interacted with it.

4. Our proposed method

Our recommender consists of two modules. The first is the (i) graph profile updater, and the second is the (ii) recommender that runs on top of the graph profile updater to deliver the top-N recommended items to each user.

The graph profile updater module reads instances from the stream of user sessions combining them with earlier recorded information on the participating entities (users, items, sessions). Thus, it assigns validity intervals to elements of the sessions stream S. Then, a sliding time window of size w states that the processing at a point in time t should consider all events not older than $t - w$. Therefore, the graph profile updater sets a validity interval $[t - w, t)$ in which it computes the similarity among items based on the valid whole graph or sub-graphs that derive from it. The pseudo-code of our graph profile update is depicted in Algorithm 1.

Algorithm 1. Graph profile updater

```
Input  : stream of sessions S, graph database D, window length w in timepoints
Output: sessions S_act, users U_act, articles A_act, adjacent category and location nodes
S_act ← ∅
foreach timepoint t do
    S_t ← sessions arrived in (t − w, t]
    foreach session node s which belongs in S_t do
        Retrieve all its adjacent user and articles nodes from D
        foreach retrieved article node a do
            Retrieve all its adjacent category and location nodes from D
        end
    end
    Remove all session nodes that are inactive in S_t from S_act
end
```

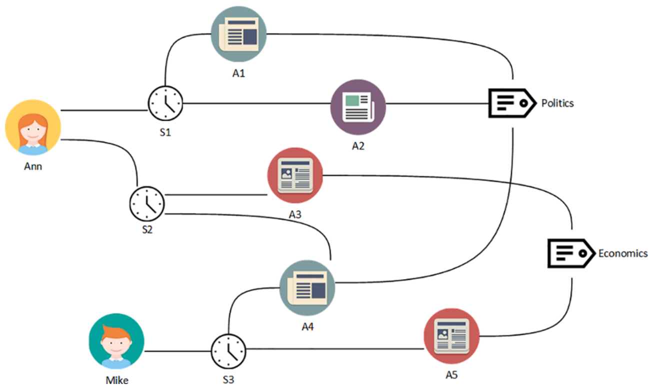

Fig. 2. Toy example of a heterogeneous graph network USAC.

Table 2
Different approaches to infer similarity among items and the corresponding sub-graphs of the USAC network.

Sub-graph	Approach
AC	Content-based (CB)
UA	Collaborative filtering (CF)
SA	Session-based (SB)
USA	CF + SB
SAC	SB + CB
USAC	CF + SB + CB

The recommender module returns an article-article similarity matrix that SimRank algorithm builds, when it runs on the updated graph. Based on this article-article similarity matrix, our recommender predicts the next item for a specific user u in a session in combination with the weighting strategies of the articles' sequence inside a session, which will be described later. The pseudo-code is given in Algorithm 2. Please notice that for the implementation of Stream-based SimRank, we use a special version proposed by Yu et al. (Yu et al., 2013), which speeds up the time complexity of SimRank from $O(k \times n^4)$ to $O(k \times d' \times n^2)$ time, where k is the number of algorithm's iterations, n is the number of nodes in the under consideration graph and d' is typically much smaller than the average in-degree of a graph.

4.1. SimRank for multi-modal and time-evolving graphs

In this Section, we will describe how SimRank can run in multi-modal and time-evolving graphs.

Example 2. Let's consider a simple toy example of a heterogeneous graph with a structure USAC (user-session-article-category), shown in Fig. 2. From the following graph structure we can extract different sub-graphs, corresponding to different recommendation approaches (see Table 2).

For different approaches of similarity search (e.g. Content-based, Collaborative filtering, Session-based), different general SimRank statements apply, though all corresponding to the underlying idea, that *two objects are similar if they are related to similar objects*. In Table 3 the SimRank statements for three discussed approaches are presented.

On all the different aforementioned sub-graphs of Table 2, we can iteratively apply SimRank function (Eq. 4) calculating article-article similarities, using damping factor $C = 0.8$ and with the convergence criteria $\varepsilon <= 10^{-4}$. For our running example of Fig. 2, the detailed computed results after applying SimRank on the whole USAC graph are shown in Fig. 3. However, for easier reading and comprehension, Table 4 shows only the resulting similarity scores per approach for article A1 with the four other articles.

In the following, we comment on the similarity score that is attained among articles after running SimRank on different sub-graphs:

Content-based (CB) similarity based on AC sub-graph: As it is shown in the second column of Table 4, only articles A2 and A4 belong to the same category, as article A1, thus they get similarity score of 0.8 (our damping factor C). Other articles are treated as absolutely

Table 3
SimRank similarity statements for different cases to infer similarity.

Case	SimRank statement
Base case	Two objects are similar if they are related to similar objects.
CF case	Two items are similar, if they are read by similar users, Two users are similar if they read similar items.
CB case	Two items are similar if they belong to similar categories, Two categories are similar if they unite similar items
SB case	Two items are similar if they appear in similar sessions, Two sessions are similar if they hold similar items.

Algorithm 2. Stream-based SimRank (G, C, k)

```
Input: The updated graph G, the damping factor C, the iteration number k
Output: SimRank score s(a, b), ∀ a, b ∈ G
foreach a ∈ G do              /* Initialization */
    foreach b ∈ G do
        if a == b then R(a, b) = 1
        else R(a, b) = 0
while (k > 0) do              /* Iteration */
    k ⟵ k − 1
    foreach a ∈ G do
        foreach b ∈ G do
            in ⟵ 0
            foreach i_a ∈ I(a) do
                foreach i_b ∈ I(b) do
                    in ⟵ in + R(i_a, i_b)
            R*(a, b) ⟵ w * (C*in)/(|I(a)||I(b)|)
    foreach a ∈ G do              /* Update */
        foreach b ∈ G do
            R(a, b) = R*(a, b)
return R(*,*)
```

	Ann	Mike	S1	S2	S3	A1	A2	A3	A4	A5	Politics	Economics
Ann	1.000	0.203	0.000	0.000	0.000	0.418	0.418	0.371	0.346	0.208	0.000	0.000
Mike	0.203	1.000	0.000	0.000	0.000	0.189	0.189	0.283	0.426	0.556	0.000	0.000
S1	0.000	0.000	1.000	0.324	0.191	0.000	0.000	0.000	0.000	0.000	0.447	0.176
S2	0.000	0.000	0.324	1.000	0.316	0.000	0.000	0.000	0.000	0.000	0.321	0.356
S3	0.000	0.000	0.191	0.316	1.000	0.000	0.000	0.000	0.000	0.000	0.281	0.391
A1	0.418	0.189	0.000	0.000	0.000	1.000	0.579	0.202	0.342	0.167	0.000	0.000
A2	0.418	0.189	0.000	0.000	0.000	0.579	1.000	0.202	0.342	0.167	0.000	0.000
A3	0.371	0.283	0.000	0.000	0.000	0.202	0.202	1.000	0.343	0.413	0.000	0.000
A4	0.346	0.426	0.000	0.000	0.000	0.342	0.342	0.343	1.000	0.338	0.000	0.000
A5	0.208	0.556	0.000	0.000	0.000	0.167	0.167	0.413	0.338	1.000	0.000	0.000
Politics	0.000	0.000	0.447	0.321	0.281	0.000	0.000	0.000	0.000	0.000	1.000	0.189
Economics	0.000	0.000	0.176	0.356	0.391	0.000	0.000	0.000	0.000	0.000	0.189	1.000

Fig. 3. Predicted Article-Article Similarity Scores of SimRank, which is applied on the whole USAC graph (combined similarity of CB + CF + SB).

Table 4
SimRank similarities of article A1 with other articles on different sub-graphs.

	CB	CF	SB	CF + SB	SB + CB	CF + SB + CB
A1-A2	0.8	0.8	0.8	0.8	0.608	0.579
A1-A3	0	0.8	0	0.281	0.171	0.202
A1-A4	0.8	0.573	0	0.198	0.343	0.342
A1-A5	0	0.347	0	0.115	0.171	0.167

dissimilar.

Collaborative filtering (CF) similarity based on UA sub-graph: The bipartite graph user-article (UA) is derived from the original graph by joining users with articles directly, skipping the sessions grouping. As shown in Table 4, an observation here is that even though articles A1 and A5 are not read by the same user, they get a similarity score of 0.347, because they are read by *similar* users, which in their turn become similar because they have read the same article A4.

Session-based (SB) similarity using the SA sub-graph: Only article A2 appears in the same session with A1, thus all other articles are treated as dissimilar. Please notice that even though articles A3 and A5 do not belong to the same session, they get a similarity bigger than 0 (as shown in Fig. 3), because the sessions, that they belong to, are similar (i.e., both hold article A4).

Combining CF with SB similarity based on USA sub-graph: As shown in Table 4, the pair A1-A2 has the same similarity for both CB and CF approaches, as they both are read by one user (Ann) within the same session. A1-A3 get a lower similarity, as they are read by the same user but within different sessions (note, that the time interval between those sessions could be 1 month or 1 year, and so those articles would very unlikely be similar).

Combining SB with CB similarity based on SAC sub-graph: As it is shown in Table 4, the pair A1-A2 again attains the biggest similarity, because of belonging to the same session and category. A4 gets lower similarity, as only belongs to the same category, but different session. Articles A3 and A5 get similarity to A1 bigger than 0, even while belonging to different categories and different sessions. The reason is, that they both co-appear in a session together with A4, which is similar to A1, which makes sessions S1, S2 and S3 also similar, and finally, their articles A3 and A5 also similar to A1.

Combining CF with SB and CB using the USAC sub-graph:. This combination captures all available information and can be considered as being able to capture better the notion of similarity among entities of the real-world. The applied idea is very similar to the previous SB + CB case, with the only difference, that the pair A1–A3 wins against A1–A5, because of being read by the same user Ann (sessions S1-S2 become more similar than S1–S3).

In the experimental section, we will run experiments with different recommendation strategies and their combinations that will run on different sub-graphs of the whole graph, so that we can identify the most effective one.

4.2. Weighting strategies of sequential articles

In this section, we compare four different ways of capturing the importance of each article in a user session to infer his short-term reading goals. User preferences might change within even a short interaction with a system depending on the already seen items. Thus, we want to identify how we should weight the articles of the session to best reflect user's current intention, preference and attention over items. Here are the four approaches to be compared:

1. $\mathbf{W_{LA}}$ - *only the last viewed article matters.* The recommendation is made by finding the most relevant articles to the *currently viewed one.* This approach provides non-personalized recommendations, as only relies on one item. This approach is most commonly used in modern e-commerce systems, such as *Amazon*, *eBay*, etc., where the recommended items are divided into groups like: "This item was frequently viewed together with …", "People who viewed this item also viewed …", etc.
2. $\mathbf{W_{M}}$ - *all articles of the session are equally important.* The simple *mean* of relevance vectors of all articles revealed from the session is calculated, and these articles are recommended, that have the highest average similarity score (i.e., are the most relevant inside the user session).
3. $\mathbf{W_{S}}$ - *the closer the article to the end of the session, the more important it is.* We assume that recommended articles should be more relevant to the most recent session articles and less relevant to the oldest ones. We propose to use a Sigmoid-based weight assignment function. The general Sigmoid function is stated as:

$$S(x) = \frac{1}{1+e^{-x}} \tag{5}$$

As long as Sigmoid lays in the interval $\approx [-5;5]$, where $S(-5) \approx 0$, $S(5) \approx 1$, we propose to assign weights of the articles in a session using the following equation:

$$w(a) = S(-5 + \frac{10 * n(a)}{N}), \tag{6}$$

where $n(a)$ is the position of the article inside a session, and N is the total number of articles inside a session. The weights are then normalized by dividing each by the sum of all, so that all weights sum up to one.
4. $\mathbf{W_{T}}$ - *the longer is the reading duration of an article, the more important it is.* A large amount of articles within user sessions are read for less than 20 s, usually denoting either unintended user actions or a lack of user interest in the article. Therefore, we propose to make the weight of an article to be depended on its timeview. We use the following piecewise logarithmic function to bring closer articles that were read for long time enough and give very small weights to not interesting articles:

$$w(a) = \begin{cases} log_{10}(t(a)+1), & \text{if } t(a) < t_{breakpoint}; \\ ln(t(a)+1), & \text{if } t(a) >= t_{breakpoint}, \end{cases} \tag{7}$$

where $t(a)$ is the time duration of the article a within the session in seconds, and $t_{breakpoint}$ is a time duration under which we treat the article views as less important. In our experiments we use $t_{breakpoint} = 20$. Normalisation of weights is performed afterwards.

In the next section, we will evaluate the performance of SimRank, as we change the sequential article weighting strategies (W_{LA}, W_M, W_S, W_T), to identify the most effective one.

5. Experimental evaluation

5.1. Log analysis of real-life datasets

In this Section, we will describe the basic characteristics and statistics of two real-life data sets (acquired from two news providers' - an Italian and a German- that operate in the region of Alto Adige in Italy). These datasets will be later used to evaluate the effectiveness of our method against state-of-the-art graph-based and other methods (i.e., SimRank, PathSim, HeteSim, Session-knn, etc.).

5.1.1. Italian news provider data set

For the Italian news provider, the data set accommodates 14367 interactions/events/views on 2081 articles of 10421 unique users in one year (i.e. from 1st April 2016 to 30th March 2017). This means that the average number of views/clicks per article is 6.9, which will affect the prediction of all models due to sparsity. The interactions of each session are logged with the following information: the user session's identifier, the interaction's time stamp and duration, the article's textual content.

Based on the articles' text, we trained a Latent Dirichlet Allocation (LDA) (Blei, 2012) model and found five topic categories in which an article may belong to. Thus, we have classified news articles into five topic categories, $C = \{c_1, c_2, \ldots, c_n\}$, including, "Politics", "Local News", "Education", "Justice", and "Sports". Please notice that the distribution of articles on the 5 found categories is more biased for articles about Politics and Local News as shown in Fig. 4a. But there is also important interest of the crowd about the other 3 topics.

5.1.2. German news provider data set

For the German news provider, the data set accommodates 5536 interactions/events/views on 468 articles of 3626 unique users in one year. This means that the average number of views/clicks per articles is 11.8, which is double than the one for the Italian news provider. Thus, for this data set, we expect that all prediction models will perform better since it is denser. We have also classified news articles into five topic categories, including again "Justice", "Education", "Politics", "Local News", and "Sports". As shown in Fig. 4b, the distribution of articles on the 5 found categories is very biased for articles about Justice and less about Education, and the crowd shows very small interest about the other 3 topics. This fact may affect the performance of the prediction models in terms of diversity of their recommendations. Another important observation extracted from Fig. 4 is that different public communities inside one geographic region can have very different interests, and this should also be considered when building and diversifying recommendations.

Detailed general statistics of both datasets are summarized in Table 5, where the cleaning procedure lies in removing the sessions that contain only one article, as no predictions can be tested on such sessions, and no article co-occurence patterns can be identified. Please notice that most of the users have only a small number of sessions (i.e., 1.23 or 1.17 sessions per user in the Italian and the German news provider, respectively).

5.2. Prequential evaluation protocol

In this Section, we present our evaluation protocol, which is in the same direction, with the one introduced by Jannach, Lerche, and Jugovac (2017) and Ludewig and Jannach (2018) for predicting the next item inside a session, known also as prequential evaluation in stream mining (Quadrana, Cremonesi, & Jannach, 2018; Vinagre, Jorge, & Gama, 2014).

As shown in Fig. 5, in prequential evaluation, future articles are first predicted by the model, so that the quality of the model is evaluated; then articles with their true labels are used for model learning, which means that approaches adapt to the user's every next click. As also shown in Fig. 5, results are obtained when applying a sliding-window protocol, where we split the data into several slices of equal size. An important parameter of this protocol is the sliding time window size of the training data. If this sliding time window is too large the system is not sensitive to changes (concept drifts). if it is too small there is not enough data to build a model predicting the next items in a session. Please notice also in Fig. 5 that parameter v determines up to how many

Table 5
General statistics of datasets.

	Italian News Provider		German News Provider	
	Before cleaning	After cleaning	Before cleaning	After cleaning
Users	10 421	918	3 626	600
Articles	2 081	1 097	468	169
Sessions	11 686	1 126	4 078	704
Views	14 367	3130	5 536	2 161
Avg # views per session	1.23	2.78	1.36	3.07
Avg # views per user	1.38	3.41	1.53	3.6
Avg # sessions per user	1.12	1.23	1.2	1.17
Avg # sessions per article	5.61	2.53	9.94	6.68
Avg # views per article	6.9	2.85	11.8	12.78

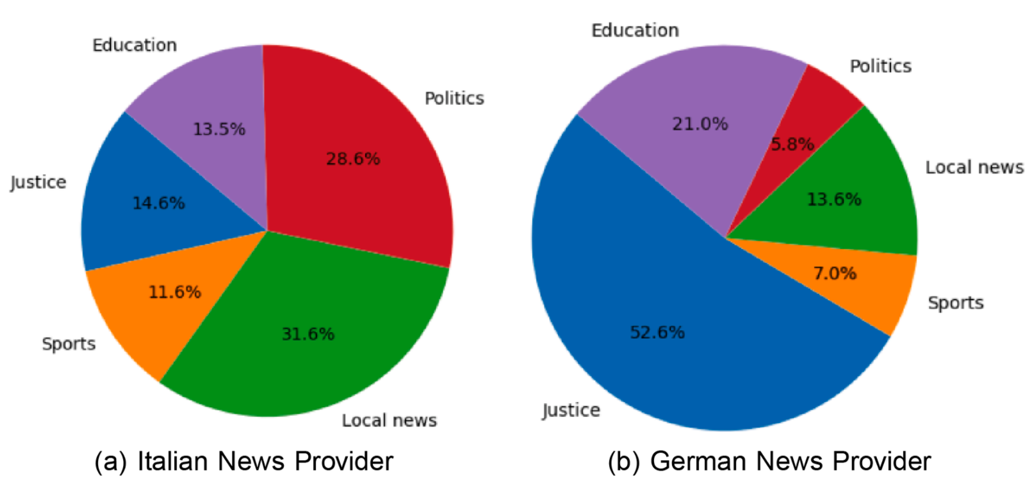

Fig. 4. News category preferences by users of two news articles providers.

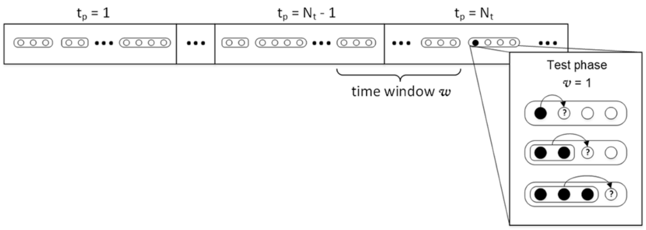

Fig. 5. Identifying user session short-term intentions.

items/views of the currently evaluated session are revealed each time. This parameter helps us to assess how many clicks it takes for an adaptive recommendation strategy to guess the user's intention.

5.3. Evaluation metrics

5.3.1. Normalized discounted cumulative gain (nDCG)

nDCG is a fine-grained version of precision, that takes also into account the position of a correct item in the recommendation list (it is particularly important in RS as lower positioned items may be overlooked by users). Eq. 8 is proposed to be used for prequential evaluation of RSs in 2017 by Song et al. (2017) as a simplified version of a standard nDCG metric.

$$nDCG = \frac{1}{log_2(1 + rank_{pos})}, \quad (8)$$

where $rank_{pos}$ denotes the rank of a positive item, i.e. the position of correctly predicted item. The lower is placed a relevant item in the recommendation list, the lower is the nDCG score.

5.3.2. Intra-list diversity (ILD)

Diversity is a metric that captures how different to each other are the items within the recommendations list (intra-list diversity). One of the most frequently considered diversity metrics, first proposed to use in the context of recommendation by Smyth et al. (2001), is the so-called *average intra-list distance* or just *intra-list diversity*, ILD (Castells, Hurley, & Vargas, 2015). ILD of a set of recommended items is defined as the average pairwise distance of the items in the set:

$$ILD = \frac{1}{|R|(|R| - 1)} \sum_{i \in R} \sum_{j \in R} d(i,j), \quad (9)$$

where R is a recommendation list, and $d(i,j)$ is a distance measure between items i and j. Distance measure $d(i,j)$ is a configurable element of the metric and can be defined as a complement of any well-known similarity measure between two items. In our experiments we use a simplified version of content-based item distance measure: $d(i,j) = 1$, if items i, j belong to different categories, and $d(i,j) = 0$, if to the same. This way we want to measure how diverse is the recommendation list in terms of news categories.

5.4. Sensitivity analysis of SimRank

In this Section, we evaluate the performance of SimRank in terms of nDCG, and ILD evaluation metrics vs. the number of recommendations made, when we change:

1. the time window size $w = [0.5, 1, 2, 3, 4]$;
2. the sub-graph used for model learning (SA, USA, SAL, etc.);
3. the sequential article weighting strategy (W_{LA}, W_M, W_S, W_T) as discussed in Section 4.2;

We provide top-5 recommendations for predicting each next article.

5.4.1. Changing the time window size w

In this Section, we analyse the performance of SimRank, as we change the sliding time window size w in both data sets (Italian and German news providers) by setting the time period splits $N_t = 365$. We run SimRank on the whole USACL graph, and weight articles in a session using W_M approach (i.e., all articles of the session are considered as equal important), and the number of revealed articles from the session is $v = 1$. The performance results are shown in Table 6.

As we can see from Table 6, the window size that provides the highest nDCG is $w = 1$ day for the Italian news provider dataset, and $w = 2$ days for the German news provider. It is impressive the fact that nDCG for German news provider is 2 times better than the nDCG for the Italian news provider in all measurements. The reason is that news articles in German news providers web site are viewed twice as much comparing to the news articles on the Italian news provider web site, as it can be shown by the last row of Table 5. Finally, as expected, ILD is getting higher as we increase the size of w, since more news articles can be considered for recommendation. However, we have to underline the fact that all models attain better ILD for the Italian news provider than the German. The main reason is that German crowd shows more interest about 2 only topics, whereas the Italian speaking population shows more distributed interest about the 5 topic categories, as can be seen in Fig. 4a. In further experiments we use $w = 1$ and $w = 2$ for the Italian and the German news provider, respectively.

5.4.2. Changing the network structure G

In this Section, we evaluate the performance of SimRank when we run it on different sub-graphs that can be derived from the whole USACL graph. Please notice that the acquired results are not always stable and slightly fluctuate accidentally with each program run and thus we report results which are averaged after performing 30 runs. The performance results of the SimRank models based on different sub-graphs are provided in Table 7 in the descending order of nDCG.

For the Italian news provider dataset, as shown in Table 7, the two best results of SimRank in terms of nDCG are acquired using very similar sub-graphs SACL, and UACL. That is, there is almost no difference between the impact of session (S) and user (U) entities. This effect is easily explainable, as long as during the dataset analysis in Section 5.1 we

Table 6
SimRank evaluation with different window sizes *w*.

w size	Italian Provider		German Provider	
	nDCG	ILD	nDCG	ILD
0.5 days	0.09	0.42	0.44	0.06
1 day	**0.23**	0.68	0.46	0.11
2 days	0.15	0.44	**0.51**	0.18
3 days	0.11	0.57	0.48	0.17
4 days	0.08	**0.69**	0.47	**0.20**

Table 7
Models comparison.

sub-graph	Italian Provider		sub-graph	German Provider	
	nDCG	ILD		nDCG	ILD
SACL	**0.24**	0.69	SA	**0.53**	**0.17**
UACL	0.24	0.68	SAL	0.52	0.17
USAC	0.23	0.68	SAC	0.52	0.16
USACL	0.23	0.68	USACL	0.51	0.16
SAC	0.23	0.69	UA	0.51	0.16
UAC	0.23	0.68	AL	0.51	0.15
AL	0.23	**0.74**	AC	0.50	0.15
UAL	0.22	0.73	UAC	0.49	0.15
SAL	0.22	**0.74**	ACL	0.48	0.15
AC	0.2	0.69	UAL	0.48	0.15
UA	0.2	0.72	UACL	0.47	0.15
USAL	0.19	0.73	USA	0.47	0.15
USA	0.19	0.72	SACL	0.47	0.15
SA	0.19	0.72	USAL	0.47	0.15

figured out, that most of the users have only a small number of sessions (i.e., 1.23 or 1.17 sessions per user in the Italian and the German news provider, respectively), and therefore S≈U. Moreover, sub-graphs containing both location (L) and news category (C) entities perform very well, which means that the category of an article and the location that it refers to is a very important factor for correct predictions. Moreover, from the top cells of Table 7 we verify that collaborative filtering based approaches (UACL, USACL, USAC) are performing better than content-based ones (AL, AC). This is also as expected, because the recommendation accuracy performance of the content-based approaches in our case is very close to random, since they randomly select 5 articles from the same category or location (or both) of the articles of the current user session to form a recommendation list, without any additional ranking criteria.

For the German news provider, sub-graph SA attains the best accuracy and diversity score. This means that session-based recommendation is adequate in providing good recommendations. When we combine with it either location (L) or news category (C) entities, then the performance of SAL and SAC slightly drops. Again, collaborative filtering (i.e., USACL, UA) performs better than content-based filtering (i.e. AL, AC). Please notice that in both datasets the exploitation of sessions leads to better prediction performance. Finally, please notice from Table 7 that the diversity (ILD) of recommended items from more accurate models is not higher, which is in accordance with the commonly accepted claim that in recommender systems accuracy and diversity metrics may contradict to each other, and needs further investigation.

5.4.3. Changing the weighting strategy of sequential articles in a session

In this Section, we perform an experiment to clarify if there is a noticeable difference between the four article weighting approaches described in Section 4.2. For the training we use all the available data, whereas for the test phase we perform two additional steps:

1. *Test sessions cleaning*: we remove the articles that do not appear in the training set (no relevance vectors exist for them). This way the comparison is more accurate: for example, there are no sessions of length 3, in which only 1 article has appeared in the training and, thus, only 1 relevance vector is considered in all four approaches;
2. *Establishing parameter* $v = 2$: we start making recommendations after revealing two consequent user view actions/items (from already filtered sessions). The reason for this is that we want to objectively compare the influence of different article weighting approaches, while the prediction for the first article in a session should be the same for all approaches.

For the Italian news provider, we run the SimRank on the SACL graph, because, as discovered in the previous experiment, it provides the best overall nDCG. For the German news provider, we run the SimRank on the SA graph. For both data sets, as shown in Table 8, the two best performing strategies are W_{LA} and W_S, which are both biased towards the last article in the session. The 3rd strategy in terms of accuracy is W_M, and the least accurate is W_T. However, the actual time that an article is read (i.s., W_T strategy) does not really give us better prediction results and maybe it should be considered together with W_S and needs further investigation. Henceforth, we use the W_{LA} (i.e., only the last article in the session matters) strategy to treat the more recent clicked items in a session as more important ones, since it attains also high effectiveness in terms of diversity.

Table 8
Models comparison.

Weighting strategy	Italian Provider		German Provider	
	nDCG	ILD	nDCG	ILD
W_{LA}	**0.26**	**0.69**	**0.65**	**0.16**
W_S	0.24	0.69	0.53	0.16
W_M	0.23	0.68	0.48	0.14
W_T	0.23	0.68	0.44	0.14

5.5. Comparison with other methods

In this Section, we compare our approach against well-known graph-based methods (i.e., SimRank, PathSim, HeteSim) and other state-of-the-art methods (i.e., IBCF, Session-knn, GRU4REC, Cat-TPM).

Based on the sensitivity analysis of SimRank that we performed in Section 5.4, we will continue our experimentation with two SimRank models for the two data sets: (i) based on SACL sub-graph as the most accurate for the Italian news provider, and (ii) based on SA graph as the most accurate for the German news provider. Moreover, we compare our method with the following comparison partners:

(i) **Most Popular Recent Items (POP):** POP recommends the top-N most clicked articles of the active time window.

(ii) **Item-based Collaborative Filtering (IBCF) (Das, Datar, Garg, and Rajaram, 2007):** Based on IBCF, two items are considered similar, if they are selected by similar users. In Das et al. (2007), IBCF considers the co-visitation count of news articles, which counts the number of times an item was co-visited (clicked before of after) with another item.

(iii) **PathSim (Yizhou et al., 2011):** PathSim captures the nodes' *visibility* in the network, bringing closer the nodes that share similar visibility, differently than SimRank, which is biased towards more popular items in the network. Based on PathSim and given a symmetric meta path P, PathSim between two objects of the same type x and y is as follows:

$$s(x,y) = \frac{2 * |p_{xdzigrarr;y} : p_{xdzigrarr;y} \in P|}{|p_{xdzigrarr;x} : p_{xdzigrarr;x} \in P| + |p_{ydzigrarr;y} : p_{ydzigrarr;y} \in P|}, \quad (10)$$

where $p_{xdzigrarr;y}$ is a path instance between x and y, $p_{xdzigrarr;x}$ is that between x and x, and $p_{ydzigrarr;y}$ is that between y and y.

(iv) **HeteSim (Shi et al., 2014):** This is a path-based measure, which can measure the relatedness of same-typed or different-typed objects in heterogeneous graphs. HeteSim is based on the similar idea of SimRank but it is path-constrained, which means that it uses also meta paths for inferring relevance similarity among different entities of a graph.

(v) **Session-knn (Jannach et al., 2017):** Session-knn method takes the set of user actions in the current session, e.g. two view events for certain items, and then in a first step determines the k most similar past sessions in the training data. Then, given the current session s, the set of k nearest neighbors N_s, and a function $sim(s1, s2)$ that returns a similarity score for two sessions $s1$ and $s2$, the score of a recommendable item i is:

$$score_{KNN}(i,s) = \sum_{n \in N_s} sim(s,n) \times 1_n(i), \quad (11)$$

where $1_n(i) = 1$ if n contains i and 0 otherwise. The similarity measure used by Jannach et al. (2017) in experiments is cosine similarity, as it was found out that the best results are achieved when encoding sessions as binary vectors of the item space.

(vi) **News Category Transition Probability Matrix (Cat-TPM) (Liu et al., 2010):** Based on Cat-TPM, when a user selects two articles in a row, a transition from a category of the first article to a category of the second article is recorded. Cat-TPM combines the content-based with the collaborative filtering methods to generate the personalized Google news recommendations (Liu et al., 2010).

(vii) **GRU4REC (Hidasi, Karatzoglou, Baltrunas, and Tikk, 2015):** GRU4REC is a neural network-based recommender system that

Table 9
Models comparison.

Model	Italian Provider		German Provider	
	nDCG	ILD	nDCG	ILD
SimRank	**0.26**	**0.69**	**0.65**	**0.16**
PathSim	0.23	0.66	0.54	0.14
HeteSim	0.23	0.62	0.49	0.14
POP	0.23	**0.69**	0.48	0.14
Cat −TPM	0.20	0.66	0.47	0.14
Session −kNN	0.18	0.61	0.46	0.14
GRU4REC	0.16	0.57	0.43	0.13
IBCF	0.14	0.50	0.38	0.10

uses a Gated Recurrent Unit (GRU), which learns when and how much to update the hidden state of the GRU model.

The parameters we used to evaluate the performance of the comparison partners are identical to those reported in the original papers and for our data sets were tuned so as to get the best results for these methods with the best found window sizes.

Table 9 reports the average nDCG of the under comparison algorithms by setting the sliding time window size $w = 1$ day for the Italian and $w = 2$ day for the German news provider.

As shown in the first row of Table 9, our proposed approach has the best overall average nDCG in both datasets. The reason is that SimRank sums up the co-visitation probability of two articles after all possible steps by taking under consideration of the SACL and SA network structure, respectively for the Italian and the German data set. Thus, SimRank is able to adequately capture the short-term preferences (i.e., S stands for sessions) of individuals and their news categories and locations of interest (i.e., C stands for articles Category and L stands for Location). PathSim penalises the similarity of articles that are popular, without a particular reason and thus, accuracy drops. HeteSim also has lower effectiveness, because it requires to transform odd-length paths into even-length paths, by incorporating artificial nodes. Cat −TPM fails to provide accurate predictions because it is more suitable for long term predictions, since it is mainly based on news categories similarity. Session −kNN and IBCF are not so effective because they do not consider the intra-session similarity of items. That is, items that belong in the same session should be also considered as more similar. GRU4REC is outperformed by Session −kNN as also reported in (Jugovac, Jannach, & Karimi, 2018), and it seems that it is not adequate for capturing the notion of similarity among items or users in radically changing environments (i.e. news recommendations). Finally, it worths underlying the fact that POP is very effective in term of accuracy, as also reported in (Ludmann, 2017) for the news recommendation domain.

6. Discussion

In this Section, we discuss some of the possible advancements that could be applied to our proposed method.

As far as the scalability of our SimRank similarity measure $sim(a,b)$, it can be considered as the probability that two random surfers, starting from node a and node b, will meet at a middle node on the path that connects them. Based on this assumption, we can easily deal with very big data since we can perform random walks (i.e. a kind of sampling over all paths that connect the nodes), which can be controlled by how soft or hard the convergence criterion is set. Of course, a harder termination/convergence criterion may result in shorter random walks, which may lead to higher scalability but then our recommendations may become less accurate.

Our experiments have been performed with an off-line evaluation protocol. That is, we had to pre-process the data of sessions such that we have at least two items in each session to be able to evaluate the next item to be recommended. Thus, if we consider the data sample as representative for the actual data, someone could argue that the evaluation is not faithful to the real case scenario. To overcome this issue, we would suggest that also an online evaluation of the methods would give a better insight of our findings.

As far as the diversity is concerned, someone could argue that POP algorithm, has equal ILD with our SimRank as shown in Table 9. We have to highlight the fact that there is no method in the comparison partners, which has higher ILD with the highest precision simultaneously together. That is, our *SimRank* method is able to provide the most accurate recommendations with minimal loss in terms of ILD.

In this paper, we focus on users' satisfaction, which can be measured by the precision/recall or the Click Through Rate (CTR) metric. Of course, high click-rates might not reflect users' satisfaction under all circumstances. For instance, users who have to click many recommendations to find articles which they find exciting boost the click rate while becoming dissatisfied. In summary, real systems experience a conflict between objectives (i.e, user satisfaction versus item sales promotion) and probably no single objective could be further improved without hurting the other one.

7. Conclusion

In this paper, we proposed a stream-based version of SimRank that runs on a time-evolving graph. We combine it with a sliding time window model and several weighting strategies to reveal the user's short term intention. We run experiments on two real-life data sets and our approach in all cases outperforms graph-based and other state-of-the-art session-based recommendation methods. In the same direction with Symeonidis et al. (2019), we use side information (i.e., auxiliary information) to leverage the quality of our news recommendations. We also follow a similar strategy with the work of Symeonidis, Nanopoulos, Papadopoulos, and Manolopoulos (2008), to provide both effective and efficient news recommendations. Moreover, in contrast to Zheng, Li, Hong, and Li (2013), who considered the individual user behavior and user group behavior, we consider also the temporal behavior of users. Finally, in contrast to Li, Zheng, Yang, and Li (2014), who integrated the long-term and short-term reading preferences of users, we also take under consideration the information from the user sessions, which means that our method uses more detailed and fine-grained granularity of the time dimension.

As future work, we want to increase the effectiveness of our stream-based SimRank version, to infer similarity among entities that come from more edge types of the graph. In particular, SimRank takes under consideration only the incoming links of the graph nodes. To further increase the effectiveness of our method, we can take under consideration also the outgoing links of the graph nodes. For example, Zhao, Han, and Sun (2009) proposed P-Rank (Penetrating Rank), an enriched version of SimRank. The authors wanted to test "how similar two entities are within an information network". With P-Rank, they could effectively compute the structural similarities of entities in such networks. The difference between SimRank and P-Rank is that the latter is taking into consideration both in and out-link relationships, whereas the first was calculating the similarity based only on in-link edges. Recently, Symeonidis et al. (2019) proposed the OmniRank algorithm, which extends the SimRank and P-Rank algorithms and runs on a heterogeneous graph (i.e., heterogeneous graph) to provide omni-directional recommendations. A heterogeneous graph, consists of k-disjoint sets of nodes (i.e. users, articles, tags, etc.), incorporating also edges among nodes of the same entity set. In contrast to SimRank and P-Rank algorithms, OmniRank can capture the notion of similarity from additional edges that may connect the nodes (article-article, tag-tag, and user-user) of a heterogeneous graph. Thus, SimRank and P-Rank become just simplified special cases of OmniRank and can be easily derived from it. Finally, we want to run our stream-based SimRank version, to infer similarity among entities from Health industry. For the health industry domain, in a network schema for patients' treatment, we would have a graph that consists of Patients (P), who undergo a Treatment (T) using Medicines (M) to target Diseases (D) that may have harmful side Effects (E). We strongly believe that our proposed method could capture the notion of similarities of the complex inter-relationships of the health graph to leverage the quality of drug predictions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM, 55*(4), 77–84.

Castells, P., Hurley, N., Vargas, S. (2015). Novelty and diversity in recommender systems.

Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web* (pp. 271–280). ACM.

Epure, E. V., Kille, B., Ingvaldsen, J. E., Deneckere, R., Salinesi, C., & Albayrak, S. (2017). Recommending personalized news in short user sessions. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 121–129). ACM.

Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. CoRR abs/1511.06939 (2015), http://arxiv.org/abs/1511.06939.

Jannach, D., & Ludewig, M. (2017). When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 306–310). RecSys '17, ACM, New York, NY, USA.

Jannach, D., Lerche, L., & Jugovac, M. (2017). Adaptation and evaluation of recommendations for short-term shopping goals. In *Proceedings of the ninth ACM Conference on Recommender Systems. RecSys '15*. New York, NY, USA.

Jeh, G., Widom, J. (2002) Simrank: a measure of structural-context similarity. In: Proceedings 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2002). pp. 538–543. Edmonton, Canada.

Jugovac, M., Jannach, D., & Karimi, M. (2018). Streamingrec: a framework for benchmarking stream-based news recommenders. In *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 269–273). ACM.

Kusumoto, M., Maehara, T., & Kawarabayashi, K. (2014). Scalable similarity search for simrank. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 325–336). ACM.

Leskovec, J., Rajaraman, A., & Ullman, J. (2014). *Mining of Massive Datasets*.

Li, C., Han, J., G.He, Jin, X., Sun, Y., Yu, Y., Wu, T.: Fast computation of simrank for static and dynamic information networks. Proc. of the International Conference on Extending Database Technology (EDBT 2010) pp. 465–476 (2010).

Li, L., Zheng, L., Yang, F., & Li, T. (2014). Modeling and broadening temporal user interest in personalized news recommendation. *Expert Systems with Applications, 41*(7), 3168–3177.

Liu, J., Dolan, P., & Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces* (pp. 31–40). ACM.

Lizorkin, D., Velikhov, P., Grinev, M., & Turdakov, D. (2008). Accuracy estimate and optimization techniques for simrank computation. *Proceedings of the VLDB Endowment, 1*(1), 422–433.

Ludewig, M., Jannach, D. (2018) Evaluation of session-based recommendation algorithms. arXiv preprint arXiv:1803.09587.

Ludmann, C. (2017) Recommending news articles in the clef news recommendation evaluation lab with the data stream management system odysseus. In: 8th International Conference of the CLEF Initiative, Dublin, Ireland. CEUR Workshop Proceedings.

Quadrana, M., Cremonesi, P., Jannach, D. (2018) Sequence-aware recommender systems. arXiv preprint arXiv:1802.08452.

Shi, C., Kong, X., Huang, Y., Philip, S. Y., & Wu, B. (2014). Hetesim: A general framework for relevance measure in heterogeneous networks. *IEEE Transactions on Knowledge and Data Engineering, 26*(10), 2479–2492.

Smyth, B., & McClave, P. (2001). Similarity vs. diversity. In *Proceedings of the 4th international conference on Case-Based Reasoning: Case-Based Reasoning Research and Development* (pp. 347–361). London, UK.

Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., & Tang, J. (2017). *Session-based social recommendation via dynamic graph attention networks*. ACM.

Symeonidis, P., & Chairistanidis, S. (2019). Omnirank: learning to recommend based on omni-traversal of heterogeneous graphs. *Social Network Analysis and Mining, 9*(1), 1–13.

Symeonidis, P., & Malakoudis, D. (2019). Multi-modal matrix factorization with side information for recommending massive open online courses. *Expert Systems With Applications, 118*, 261–271.

Symeonidis, P., Nanopoulos, A., Papadopoulos, A. N., & Manolopoulos, Y. (2008). Collaborative recommender systems: Combining effectiveness and efficiency. *Expert Systems with Applications, 34*(4), 2995–3013.

Tao, W., Yu, M., & Li, G. (2014). Efficient top-k simrank-based similarity join. *Proceedings of the VLDB Endowment, 8*(3), 317–328.

Usman, B., Oseledets, I. (2015) Tensor simrank for heterogeneous information networks. arXiv preprint arXiv:1502.06818.

Vinagre, J., Jorge, A., Gama, J. (2014) Evaluation of recommender systems in streaming environments.

Yizhou, S., Jiawei, H., Xifeng, Y., Philip S., Y., Tianyi, W. (2011) Pathsim: Meta path-based top-k similarity search in heterogenuos information networks. In: Proceedings of the VLDB Endowment (VLDB'2011). Seattle, Washigton.

Yu, W., Lin, X., & Zhang, W. (2013). Towards efficient simrank computation on large networks. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on* (pp. 601–612). IEEE.

Yu, W., & McCann, J. (2015). Efficient partial-pairs simrank search on large networks. *Proceedings of the VLDB Endowment, 8*(5), 569–580.

Zhao, P., Han, J., & Sun, Y. (2009). P-rank: a comprehensive structural similarity measure over information networks. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management* (pp. 553–562).

Zheng, L., Li, L., Hong, W., & Li, T. (2013). Penetrate: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications, 40*(6), 2127–2136. https://www.sciencedirect.com/science/article/pii/S0957417412011530.