Information Retrieval Journal (2019) 22:32–54
https://doi.org/10.1007/s10791-018-9335-0

SOCIAL MEDIA FOR PERSONALIZATION AND SEARCH

Influence me! Predicting links to influential users

Ariel Monteserin[1] · Marcelo G. Armentano[1]

Received: 27 October 2017 / Accepted: 27 June 2018 / Published online: 6 July 2018

Abstract
In addition to being in contact with friends, online social networks are commonly used as a source of information, suggestions and recommendations from members of the community. Whenever we accept a suggestion or perform any action because it was recommended by a "friend", we are being influenced by him/her. For this reason, it is useful for users seeking for interesting information to identify and connect to this kind of influential users. In this context, we propose an approach to predict links to influential users. Compared to approaches that identify general influential users in a network, our approach seeks to identify users who might have some kind of influence to individual (target) users. To carry out this goal, we adapted an influence maximization algorithm to find new influential users from the set of current influential users of the target user. Moreover, we compared the results obtained with different metrics for link prediction and analyzed in which context these metrics obtained better results.

Keywords Link prediction · Social influence · Social networks

1 Introduction

In recent years, predicting the future formation of links in a network has become a hot topic. For this reason, the task known as *Link Prediction* has arisen as an important research area in social network analysis. Several approaches have been proposed to tackle this problem. These approaches can be roughly grouped in content-based, topology-based, and learning-based approaches. However, most of these approaches focus on predicting links without taking into account the fact that there exist different kinds of nodes in a social network, for example, information sources, information seekers, idea starters, commentators, viewers, and influential, among others.

Influential nodes represent users that exert social influence on other users of the social network. Social influence occurs when one's actions are affected by others. For example, user *A* exerts influence on user *B* when *B* watches a movie because *A* recommended it previously. Thus, influential users can be seen as effective recommendation sources. Many applications exploit the concept of social influence. In the field of data mining, some applications include

✉ Marcelo G. Armentano
marcelo.armentano@isistan.unicen.edu.ar

[1] ISISTAN (CONICET/UNICEN), Tandil, Argentina

viral marketing (Wortman 2008; Monteserin and Armentano 2018), recommender systems (Ye et al. 2012), analysis of information diffusion in Facebook and Twitter (Bakshy et al. 2011), expert finding (Liu et al. 2013), decision support systems (Monteserin and Amandi 2015), analysis of scientist collaboration (Jiang et al. 2017) and ranking of feeds (Ienco et al. 2010), among others.

In this work, we propose an approach for influential links prediction (ILP): given a target user, our approach predicts links to users that could exert social influence on her/him. To do this, an influence maximization algorithm is used to determine a set of possible influential users from the set of current influential users of the target. This kind of algorithm tries to solve a key problem in the area of social networks analysis: the influence maximization problem. The influence maximization problem involves finding a set of users in a social network such that by targeting this set, the expected spread of influence in the network is maximized (Goyal et al. 2011; Kempe et al. 2003). In particular, we apply a data-based approach to social influence maximization, named *Credit Distribution* (*CD*) model (Goyal et al. 2011) that learns how influence flows in a network by directly leveraging available propagation traces. In this context, we claim that the set of nodes that also influence the nodes influenced by the set of current influential users of the target are potential new influential nodes that can be suggested as new connections to the target user. Thus, ILP searches for these nodes by using an adapted version of the *CD* model.

We claim that our approach is particularly useful for link prediction in scenarios with low homophily, where content-based approaches fail to capture similarities between nodes. Homophily is the tendency of individuals to associate and bond with similar others (Bonchi 2011). Homophily is usually taken into account by several Link Prediction approaches (for example, content-based algorithms, see Sect. 2.1). However, social influence is not the same as homophily (Aral et al. 2009; La Fond and Neville 2010). If social influence effects are present in a social network, nodes are likely to change their attributes to conform to their neighbors values. In contrast, if homophily effects are present in the network, individuals (nodes) are likely to link to other individuals with similar attribute values (Aral et al. 2009).

To validate our approach, we carried out a set of experiments in the movies domain (*Flixster*) and microblogging domain (*Twitter*). We compared the precision, recall, nDCG and AUC of ILP with respect to the main topological metrics studied in the literature for link prediction (common neighbors, Jaccard, Sørensen, and Adamic–Adar, among others Wang et al. 2015) and with respect to a learning-based approach. This comparison showed that our approach performed better that existent approaches. Moreover, we carried out a comparison of the performance of the topological metrics in two different configurations set up by varying the set of neighbors observed: only influential neighbors and all neighbors. This comparison showed that the best predictions were obtained when the topological metrics only observe the set of influential neighbors.

The article is organized as follows. Section 2 introduces some concepts related to link prediction and social influence maximization. Section 3 presents the approach to predict links to influential users. Section 4 shows the results obtained from the experiments. Finally, Sect. 5 presents our conclusions and future works.

2 Background

In this section, we describe two relevant fields, namely, *Link Prediction* and *Social Influence Maximization*. In the next section, we define the problem of link prediction and some of the main approaches to the problem. Next, we introduce the main concepts on social influence maximization, propagation models and algorithms.

2.1 Link prediction

Link prediction for social networks was formalized by Liben-Nowell and Kleinberg as the problem of predicting the edges that will be added to a given snapshot of a social network during the time period determined from time *t* to a future time *t'* (Liben-Nowell and Kleinberg 2003). Formally, given a snapshot of a network at time *t*, $G_t(V, E)$ where *V* is the set of nodes and *E* is the set of links, we seek to find the set of edges E' from all the $(|V| \cdot (|V| - 1)) - |E|$ possible links among nodes in *V* that will appear in the network at time t', $G_{t'}(V, E')$. It is worth noticing that the edges in the network can represent both connection and interactions between nodes.

Link prediction methods can be roughly grouped into three categories: content-based, topology-based, and learning-based.

Content-based algorithms assign to each pair of nodes *x* and *y* a similarity score *sim*(*x*, *y*) that is computed using the attributes of the nodes, such as the user profiles (Bhattacharyya et al. 2010), user-generated content (Armentano et al. 2013), documents information (Perlich et al. 2009), user interests (Anderson et al. 2012), etc. The similarity score for all pairs of non-connected nodes is computed and the edges with top *N* scores or, alternatively, with a score over a certain threshold, are predicted. These methods work under the assumption that users tend to relate with people who are similar to them in certain way. In other words, content-based algorithms assume that users in the network follow the principle of homophily.

Topology-based methods can be applied to any network, even if there is no information available about the nodes. These methods compute different metrics between pairs of nodes that are then used to rank the possible connections to be predicted, similarly to content-based methods. Metrics used by these methods can be divided into local (or neighbor-based) metrics, path-based metrics and random walk metrics. Liben-Nowell and Kleinberg 2003 presented different methods for link prediction based on node neighborhoods and on the ensemble of all paths. The simplest neighbor-based metric considers the number of common neighbors between two nodes (CN). Many metrics are based on CN and intend to normalize this metric with different criteria. For example, Jaccard Coefficient, use the total number of neighbors between the two nodes; Sørensen–Dice Index considers that lower degrees of nodes would have higher link likelihood, Hub promoted considers that the topological overlap is determined by the lower degree of nodes, while Hub Depressed determines the value by the higher degrees of nodes. Other approaches combine some of these topological metrics and their weighted versions (Armentano et al. 2012; Güneş et al. 2016). Among the path-based metrics, Local Path (Lü et al. 2009), Katz metric (Katz 1953) and Relation Strength Similarity (Chen et al. 2012) are commonly used for link prediction. Finally, random walk metrics use transition probabilities from a node to its neighbors to denote the destination of a random walker that departs from the current node. The random walk information can be used to measure the *distance* between any pair of nodes. Nodes are then sorted by shortest distance to select which edges to predict. Classical algorithms in this category are Hitting Time (Fouss et al. 2007), PageRank (Page et al. 1999) and it variants.

Finally, learning based methods approach link prediction as a binary classification problem. Each pair of nodes x and y is considered an instance that is described by a set of features (which are usually built from the metrics described previously) and a class label (+ if there exist an edge connecting x and y or − otherwise). Any classifier can then be used to predict the class for non-existent edges, such as decision trees (Scellato et al. 2011), naïve Bayes (Scellato et al. 2011), support vector machines (Li and Chen 2013), logistic regression (Chiang et al. 2011), frequent graph pattern mining (Pobiedina and Ichise 2016), and matrix alignment (Scripps et al. 2008). The main problem that has to be addressed when considering link prediction as a classification task is that the classes are inherently unbalanced for most networks, since the number of links that may appear represent a very small subset of the possible links that can be established in the network between any pair of nodes.

Recently, research in link prediction has also focused on dynamic networks (Rahman and Hasan 2016; Choudhury and Uddin 2017, 2018). This line of research considers that the behavior and characteristics of the nodes and the links among them change temporally. For these kind of networks, new set of metrics needs to be defined in order to measure the similarity between each pair of actors.

In this article, we focused on the topology structure of the network to locally find a set of candidate influential nodes of the target node. User actions on the network are used to determine the influence exerted on other users.

2.2 Social influence

Social influence occurs when a person's actions are affected by others. This effect can be seen in conformity, socialization, peer pressure, obedience, leadership, persuasion, sales, and marketing (Goyal 2013). Social influence is defined as the change in an individual's thoughts, feelings, attitudes, or behaviors that results from the interaction with another individual or a group (Rashotte 2006). Many applications exploit the social influence and the propagation of influence that users of a social network exert on other users has been widely studied in recent years.

A key problem in this area is the identification of influential users (Goyal et al. 2011). Kempe et al. (2003) formalized this problem as the influence maximization problem: *given a directed graph $G = (V, E, p)$, where nodes are users and edges are labeled with influence probabilities among users, the influence maximization problem looks for a set of seeds (users) that maximizes the expected spread of influence in the social network under a given propagation model.* A propagation model indicates how influence propagates through the network. Two propagation models were proposed by Kempe et al.: the *Independent Cascade* (IC) and the *Linear Threshold* (LT) models. In both models, each node can be either active or inactive at a given moment. Moreover, the tendency of each node to become active increases monotonically as more of its neighbors become active.

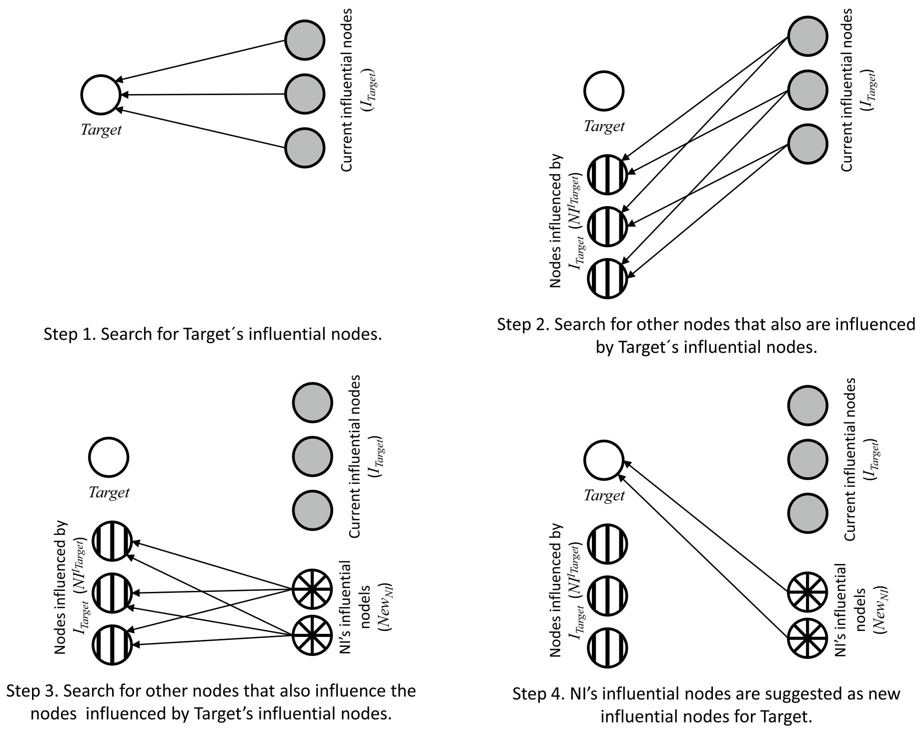
Given a propagation model m (for example, IC or LT) and an initial seed set $S \subseteq V$, the expected number of active nodes at the end of the process is the *expected (influence) spread*, denoted by $\sigma_m(S)$ (Goyal et al. 2011). Then, the influence maximization problem is defined as follows: *given a directed and edge-weighted social graph $G = (V, E, p)$ (where nodes are users and edges are labeled with influence probabilities among users), a* *propagation model m, and a number $k \leq |V|$, find a set $S \subseteq V$, $|S| = k$, such that $\sigma_m(S)$ is maximum*. Several approaches have been developed to solve this problem. Despite the fact that this problem is NP-hard under both the IC and LT propagation models, some characteristics of the function $\sigma_m(S)$ (monotonicity and submodularity, see Kempe et al. 2003 for further details) made it possible to develop a greedy algorithm to solve the problem.

One of the limitations of the *IC* and *LT* propagation models is that the edge-weighted social graph is assumed as input to the problem, without addressing the question of how the probabilities are obtained (Goyal et al. 2010). For this reason, Goyal et al. (2011) proposed the *Credit Distribution* (*CD*) model, which directly estimates influence spread by exploiting historical data. In this context, the influence maximization problem to be solved under the *CD* model is reformulated as follows: *given a directed social graph $G = (V, E)$, an action log $\mathbb{L}$, and a integer $k \leq |V|$, find a set $S \subseteq V$, $|S| = k$, such that $\sigma_{cd}(S)$ is maximum*. Under the *CD* model, $\sigma_{cd}(S)$ is defined as $\sigma_{cd}(S) = \sum_{u \in V} \kappa_{S,u}$, where $\kappa_{S,u}$ represents the total credit given to S for influencing u for all actions. To solve this problem, Goyal et al. developed an algorithm for influence maximization under the *CD* model. This algorithm initially scans the action log $\mathbb{L}$ to learn the influence probabilities in the social network, computing the influenceability scores for the users. An action log is a set of triples (u, a, t) which say user u performed action a at time t. Then, the seed set is selected under the *CD* model by using a greedy algorithm with *CELF* optimization (Goyal et al. 2011). It is worth noticing that if the timestamp in which an edge was created is available, the algorithm considers this information to find the seed set. This allows our approach to work with both static and dynamic networks. See Goyal et al. (2011) for further details on the algorithm implementation.

3 Link prediction of influential nodes

In this section, we present ILP, our approach to predict links to influential nodes. ILP is based on the social influence exerted among users in a social network. In this context, we claim that it is possible to predict links to influential nodes by observing which users also influence the set of users influenced by the current influential users of the target. Here, the target is the user to which the approach will recommend influential users (new links).

Figure 1 shows the 4 steps of the proposed approach. First, our approach searches for nodes that influence the target user (Fig. 1, Step 1). To do this, we reformulate the influence maximization problem under the *CD* model by adding the parameter T to the definition problem. Then, the influence maximization problem is defined as follows: *given a directed social graph $G = (V, E)$, a subset $T \subseteq V$, an action log $\mathbb{L}$, and a integer $k \leq |V|$, find a set $S \subseteq V$, $|S| = k$, such that $\sigma^T_{cd}(S) = \sum_{t \in T} \kappa_{S,t}$ is maximum*. In other words, we modify the problem definition to obtain a seed set by taking into account only the influence exerted on the nodes $t \in T$. Notice that when $T = V$ the problem becomes the traditional one. Thus, the first step of ILP is carried out by running the *CD* model with $T = \{Target\}$. The result of this step is the set I_{Target}, composed of the nodes that influence the target. Moreover, it is worth noticing that the greedy algorithm always returns a seed set S with k elements. However, it is possible that the last elements added to the seed set do not actually exert influence on T. This occurs whenever T is influenced only by l nodes and $l < k$. For this reason, we include a threshold *mininf*, and only keep in I_{Target} the nodes whose marginal gain exceed *mininf*, where the marginal gain of a node w is computed as $\sigma_m(S \cup \{w\}) - \sigma_m(S)$ (Goyal et al. 2011).


Target
Current influential nodes (I_{Target})
Nodes influenced by I_{Target} ($NI^{I_{Target}}$)
NI's influential nodels (New_{NI})

Step 1. Search for Target´s influential nodes.

Step 2. Search for other nodes that also are influenced by Target´s influential nodes.

Step 3. Search for other nodes that also influence the nodes influenced by Target's influential nodes.

Step 4. NI's influential nodes are suggested as new influential nodes for Target.

Fig. 1 Steps of the ILP approach

Secondly, ILP searches for nodes influenced by the set I_{Target} (Fig. 1, Step 2) and stores these nodes in the set $NI^{I_{Target}}$. Although this step is not defined as an influence maximization problem, ILP uses the concept of credit distribution to search for these nodes. Thus, we define $NI^{I_{Target}} = \{ni \in V \mid ni \neq Target \wedge \sum_{i \in I_{Target}} \kappa_{i,ni} > 0\}$. In other words, $NI^{I_{Target}}$ is composed of the nodes that gives credits to at least one node included in I_{Target} (excluding the target).

The third step consists in searching for a new set of nodes (New_{NI}) that also influence $NI^{I_{Target}}$ (Fig. 1, Step 3). To do this, we use the same influence maximization problem definition explained in Step 1 with $T = NI^{I_{Target}}$. In contrast with Step 1, New_{NI} must be filtered since the nodes of I_{Target} might have been also included in the seed set. Finally, in Step 4, ILP uses the set New_{NI} to recommend new influential links for target (Fig. 1, Step 4).

To illustrate our proposal, Fig. 2a shows an example of a simple directed social graph. This graph is composed of 9 nodes and 13 directed edges. The direction of the edge between a node *A* and a node *B* indicates that *A follows B*. Notice that influence flows contrary to the direction indicated by the edges. Table 1 shows the actions log used to learn the influence probabilities. This log has 3 columns: the id of the node that perform the action, the id of the action and the time when the action was performed. Following the steps presented above, with *Target* = 1, the first step results in a set $I_{Target} = \{3, 4\}$. Then, the second step of our approach searches for other nodes that are also influenced by nodes 3 and 4. Consequently, $NI^{I_{Target}} = \{5, 6\}$. Next,

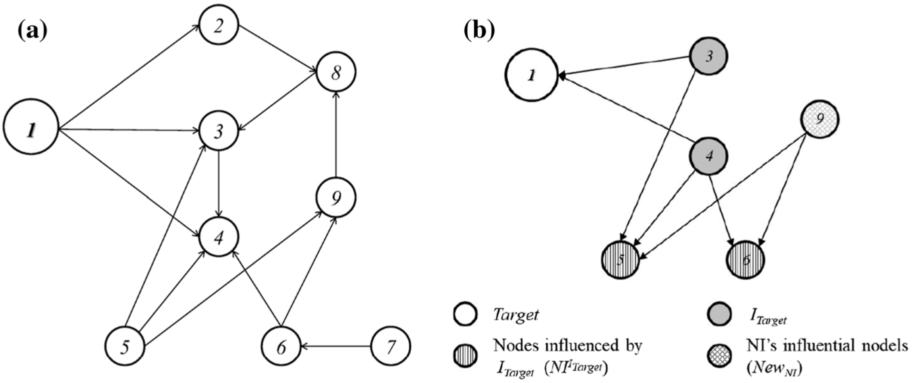

(a)
1
2
3
4
5
6
7
8
9
(b)
1
3
4
9
5
6
Target
I_{Target}
Nodes influenced by I_{Target} ($NI^{I_{Target}}$)
NI's influential nodels (New_{NI})

Fig. 2 Example of social and derived influence graph with node 1 as *target*

Table 1 Example of actions log

Node	Action	Time
3	1	1
1	1	2
5	1	2
4	2	3
1	2	4
5	2	5
3	3	6
1	3	7
4	4	7
1	4	8
6	4	8
9	5	9
5	5	10
6	5	10
7	5	11
9	6	12
5	6	13
6	6	13

ILP looks for nodes that exert influence on nodes 5 and 6. Thus, the third step returns the nodes 9, 3 and 4 ($New_{NI} = \{9, 3, 4\}$). However, since nodes 3 and 4 are included in I_{Target}, the final New_{NI} is 9. Finally, node 9 becomes a potential link to be recommended to node 1. Figure 2b shows the derived influence graph when the target is node 1. This graph shows the influence relationships among the nodes from the point of view of the target.

4 Experimental evaluation

4.1 Experimental settings

To evaluate our approach, we ran experiments comparing the performance of ILP with different well-known topological metrics for link prediction. We experimented on two well-known real-world dataset extracted from Flixster (Jamali and Ester 2010) and Twitter (De Domenico et al. 2013).

The Flixster dataset[1] is composed of 786,936 nodes; 7,058,819 directed edges; and 8,196,077 logged actions. Since Flixster[2] is one of the main players in the social movie rating businesses, each action represents a user rating a movie. Thus, if user v rates "Frozen", and later v's friend u does the same, we consider that the action of rating "Frozen" propagated from v to u (Goyal et al. 2011). Flixster dataset was chosen because it is a real-world dataset in which the similarity between linked users is low. To check this fact, we computed the average Pearson similarity ($\frac{\sum_{\forall u \rightarrow v} simPearson(u,v)}{|u \rightarrow v|}$ = 0.008) and the average GroupLens similarity ($\frac{\sum_{\forall u \rightarrow v} simGroupLens(u,v)}{|u \rightarrow v|}$ = 0.001) by using Eqs. 1 and 2, respectively.

$$simPearson(u,v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \mu_u)(r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \mu_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \mu_v)^2}} \tag{1}$$

$$simGroupLens(u,v) = \frac{\sum_{i \in I_u \cup I_v} (r_{ui} - \mu_u)(r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_u \cup I_v} (r_{ui} - \mu_u)^2} \sqrt{\sum_{i \in I_u \cup I_v} (r_{vi} - \mu_v)^2}} \tag{2}$$

Both similarity metrics measure the difference between the rating given by user u and user v to a given item i. Since different users may use different scales while rating items, the equation considers the difference between the actual rating of the user to the item and the average rating of the user to all items μ_u. The equation is then normalized so that the metric takes values between -1 (users rating items in an opposite manner) and 1 (users rating items in the same manner). Both metrics will be 0 for totally different users. The main difference between simPearson(u,v) and simGroupLens(u,v) is the set of items used to measure the similarity. Pearson correlation consider all items that were rated by both users ($I_u \bigcap I_v$). This causes that users with few rated items in common will have high similarities. If two users have rated many items and have only one item in common with the same rating, the metric will be 1, indicating that the users are similar while this might not be the case. For this reason, simGroupLens considers all the items rated by both users ($I_u \cup I_v$), with the normalized rating $r_{ui} - \mu_u = 0$ whenever u has not rated i. With this modification to the equation, if both users have rated exactly the same items, it remains the Pearson correlation. However, if one user has rated items that the other has not, those ratings drop out of the numerator (since they are multiplied by 0) but still contribute to the denominator.

It is important to consider this fact because a low similarity indicates low homophily, making unfavorable the application of content-based algorithms as we explain in Sect. 2.1.

[1] https://goo.gl/CxuBDa.

[2] http://www.flixster.com.

On the other hand, the dataset extracted from Twitter[3] was built after monitoring the spreading processes on Twitter before, during and after the announcement of the discovery of the Higgs boson. For this reason, we will refer to this dataset as *Higgs dataset*. The dataset is composed of the messages posted in Twitter about this discovery between 1st and 7th July 2012. To build the action log, we considered the tweets and retweets as actions. Thus, if a user v posts a tweet t, and later on a user u retweets this tweet, we consider t as an action propagated from v to u. Higgs dataset is composed of 456,626 nodes, 14,855,842, and 396,356 logged actions.

4.2 Procedure

We ran experiments by randomly selecting 1400 target users from each network. Then, for each target, we applied the first step of ILP and obtained the set I_{Target} with $k = 20$ and $mininf = 1.01$ (these values were also used to configure step 3 of ILP).[4] In Flixster, the average k of each I_{Target} processed was 7.2. For this reason, we applied a cross-validation technique with 4 folds. We decided to uses 4 folds since for a k-folds cross-validation we need that each user has at least k influential nodes to become a valid target, and the average amount of influential nodes of all nodes in the Flixster graph was 3.2. Thus, we discarded those users with less than 4 influential nodes during the target selection process. The same configuration was used for the Higgs dataset.

The cross-validation process consisted in picking a target, hiding each fold F_i, one at a time, and running the rest of the steps of ILP on the remaining 3 folds (i.e. $I_{Target} - F_i$). Notice that the cross-validation process was carried out by considering I_{Target}, which is the set of influential nodes, since our goal is to recommend links to influential users. Finally, we compared the results obtained, New_{NI}, with the hidden fold and computed precision and recall measures using Eqs. 3 and 4 with $New = New_{NI}$, respectively. Moreover, we computed the Normalized Discounted Cumulative Gain (nDCG) measure (Wang et al. 2013) using Eq. 5. In Eq. 5, DCG is computed using Eq. 6, where $rel_i = 1$ if the link in the position i of New was in the hidden fold and $rel_i = 0$ if not. In addition, $IDCG$ (ideal DCG) represents the DCG measure for the perfect ranking. NDCG is a normalized measure of ranking quality. The premise of nDCG is that relevant links appearing lower in a result ranking should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result.

$$precision_{New} = \frac{\left|\left\{x \mid x \in F_i \wedge x \in New\right\}\right|}{|New|} \tag{3}$$

$$recall_{New} \frac{\left|\left\{x \mid x \in F_i \wedge x \in New\right\}\right|}{\left|F_i\right|} \tag{4}$$

$$nDCG_{New} = \frac{DCG_{New}}{IDCG} \tag{5}$$

[3] https://snap.stanford.edu/data/higgs-twitter.html

[4] Notice that if some element added to the seed set does not influence T, its marginal gain will be 1.00.

$$DCG_{New} = \sum_{i=1}^{|New|} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{6}$$

Furthermore, we ran predictions by using the following state-of-the art topological metrics for link prediction (Wang et al. 2015), where $\Gamma(x)$ is the set of neighbors of node x, and $|\Gamma(x)|$ is the number of neighbors of nodes x.

- *Common Neighbor* (*CN*) this metric is one of the most widespread measurement used in link prediction due to its simplicity. CN is defined as the number of nodes that two nodes, x and y, have a direct interaction with (Eq. 7).

$$CN(x,y) = |\Gamma(x) \cap \Gamma(y)| \tag{7}$$

- *Jaccard Coefficient* (*JC*) this coefficient normalizes the size of common neighbors with the total number of neighbor that x and y have (Eq. 8).

$$JC(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{8}$$

- *Sørensen Index* (*SI*) besides taking into account the size of the common neighbors, it also points out that lower degrees of nodes would have higher link likelihood (Eq. 9).

$$SI(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|} \tag{9}$$

- *Salton Cosine Similarity* (*SC*) this metric is a common cosine metric for measuring the similarity between two nodes (Eq. 10).

$$SC(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| \cdot |\Gamma(y)|}} \tag{10}$$

- *Hub Promoted* (*HP*) it defines the topological overlap of nodes x and y. The HP value is determined by the lower degree of nodes (Eq. 11).

$$HP(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{min(|\Gamma(x)|, |\Gamma(y)|)} \tag{11}$$

- *Hub Depressed)* (*HD*) this metric is similar to HP, but the value is determined by the higher degrees of nodes (Eq. 12)

$$HD(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{max(|\Gamma(x)|, |\Gamma(y)|)} \tag{12}$$

- *Leicht–Holme–Nerman* (*LHN*) this metric assigns high similarity to node pairs that have many common neighbors compared not to the possible maximum, but to the expected number of such neighbors (Eq. 13).

$$LHN(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| \cdot |\Gamma(y)|} \tag{13}$$

- *Adamic–Adar Coefficient* (*AA*) this coefficient was initially proposed for computing similarity between two web pages. In this metric, common neighbors that have fewer neighbors are weighted more heavily (Eq. 14)

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|} \tag{14}$$

- *Preferential Attachment* (*PA*) it indicates that new links will be more likely to connect higher-degree nodes than lower ones (Eq. 15).

$$PA(x,y) = |\Gamma(x)| \cdot |\Gamma(y)| \tag{15}$$

- *Resource Allocation* (*RA*) RA metric is similar to AA. Both metrics suppress the contribution of the high degree common neighbors. However, RA metric punishes the high degree common neighbors more heavily than AA (Eq. 16).

$$RA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|} \tag{16}$$

Additionally, we ran predictions using a learning-based approach (*LB*). This approach consisted in a Logistic Regression-based classifier (using *Weka*[5] framework). For each target T and fold i, we trained a classifier whose input consisted in instances of the form $\{CN(T,y)$, $JC(T,y)$, $SI(T,y)$, $SC(T,y)$, $HP(T,y)$, $HD(T,y)$, $LHN(T,y)$, $AA(T,y)$, $PA(T,y)$, $RA(T,y)$, $class\}$ where $y \in ((\Gamma(T) - F_i) \cup NL_{training} \subset \{nl \mid nl \notin \Gamma(T)\})$ (NL was randomly selected), $\left|\Gamma(T) - F_i\right| = |NL|$ (to keep the training set balanced), and *class* was *LINK* if $y \in (\Gamma(T) - F_i)$ or *NO-LINK* if $y \in NL$. Once the classifier was trained, we tested it by using a set of instances for which $y \in F_i \cup (\{nl \mid nl \notin \Gamma(T)\} - NL_{training})$. Due to time execution limitations, we reduced the number of no-links in the testing set to 20,000 (randomly selected). Notice that this reduction benefited the performance of the learning-based approach.

Moreover, for each baseline $b \in \{CN, JC, SI, SC, HP, HD, LHN, AA, PA, RA, LB\}$, we built two rankings, according to the value of the metric or the likelihood associated to each testing instance of belonging to *the LINK* class in a descendant order:

- Ranking New_I^b with $\Gamma(x) = I_{Target} - F_i$, that is, using the same set of nodes used by our approach.
- Ranking New_{All}^b with $\Gamma(x) = \Gamma(Target) - F_i$, that is, using all the neighbors of Target without the hidden nodes.

From each ranking, we took the 20 top ranked nodes as recommendations to the target users. Finally, we also computed precision, recall and nDCG for each ranking using Eqs. 3, 4 and 5 with $New = New_I^b$ and $New = New_{All}^b$.

In addition, we compute the area under the receiver operating characteristic curve (AUC), since this is a standard metric used to quantify the accuracy of different link prediction methods (Ding et al. 2016; Lü and Zhou 2011; Dai et al. 2017). This metric can be interpreted as the probability that a randomly chosen missing link is given a higher score than a randomly chosen nonexistent link (Lü and Zhou 2011). Among n independent comparisons, if there are n' occurrences of missing links having a higher score and n''

[5] https://www.cs.waikato.ac.nz/ml/weka/.

occurrences of missing links and nonexistent link having the same score, we define the accuracy as: $AUC = (n' + 0.5n'')/n$. Then, if all the scores are generated from an independent and identical distribution, the accuracy should be about 0.5. Therefore, the degree to which the accuracy exceeds 0.5 indicates how much better the algorithm performs than pure chance (Lü and Zhou 2011).

4.3 Results

4.3.1 Flixster dataset

Table 2 shows the evolution of precision, recall, nDCG and AUC measures as the number of predictions increases (from 5 to 20). As we can see, ILP improved the classic measures in all the scenarios. The best precision was obtained by ILP with 5 predictions (3.2%) whereas the best recall was obtained with 20 predictions (15.5%). We found a significant improvement of the precision and recall of ILP with respect to the best topological metrics (CN, JC and AA): 28 and 20.15%, respectively ($p < 0.05$). The learning-based approach did not improve the metrics obtained for ILP, even though the number of no-links in the testing set was reduced.

On the other hand, we can observe that the topological metrics obtained better results generating New_I than New_{All}. For example, the best precision for New_I was obtained using CN, JC and AA metrics (2.5%). In contrast, for New_{All}, the best precision was 1.7%. This represents a significant difference of 47.06% between New_I and New_{All} ($p < 0.05$). Something similar occurred with the recall: for New_I, the best value was 12.9% using AA with 20 predictions, whereas the best value for New_{All} was 11.7% using also AA (difference of 10.25% with $p < 0.05$).

Regarding nDCG, ILP obtained a value of 0.097 in contrast to 0.084 obtained by LB. Moreover, the best nDCG value for the topological metrics was 0.082 and was also obtained for New_I using CN, whereas the best value for New_{All} was 0.06 but using AA (a significant difference of 36%). It is worth noticing that contrary to what happens with the topological metrics, the learning-based approach presented a worse performance using New_I^{LB} than New_{All}^{LB}. This is because the training set is reduced, since $|I_{Target}| < |\Gamma(Target)|$, negatively affecting the performance of the classifier.

Figure 3 shows a comparison of the AUC measure obtained by ILP and the AUC measures obtained by the baselines building New_I. The X-axis represents the number of recommendations (links predicted) and Y-axis represents the AUC measure. As we can see, ILP obtained a better AUC measure when the number of recommendation is less than or equal to 100. With more than 100 recommendations, the AUC obtained with ILP increased slightly and was overcome by some of the topological metrics (CN, JC, AA and RA). This happens because the total number of recommendations of ILP is less than the total number of recommendation that the topological metrics are able to recommend. In fact, only an average of 40.63 new connections found with ILP have a marginal gain higher than 1.01. However, we think that this is not a limitation of ILP due to the fact that when recommending new connections for users of a social network, recommendations lists tend to be short (frequently less than 20) in order to help users to focus on the most relevant results.

Figure 4 compares the AUC measure obtained by ILP with state of the art topological metrics and the learning-based approach generating New_{All}. As occurred with New_I, ILP showed a better AUC value when the number of links predicted was lower than 20. In this case, the AUC measure of AA and RA metrics mildly overcome ILP when $n = 100$. In

Table 2 Comparison of precision (P), recall (R), NDCG (N) and AUC for Flixster dataset

N	5								10							
Rank	I				All				I				All			
	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC
ILP	0.032	0.089	0.075	0.543	–	–	–	–	0.024	0.129	0.09	0.566	–	–	–	–
CN	0.025	0.075	0.066	0.538	0.017	0.047	0.039	0.521	0.017	0.095	0.074	0.548	0.013	0.071	0.048	0.535
JC	0.025	0.073	0.065	0.537	0.017	0.047	0.039	0.521	0.017	0.094	0.072	0.547	0.013	0.071	0.048	0.535
SI	0.003	0.007	0.006	0.503	0.013	0.033	0.027	0.514	0.002	0.011	0.007	0.505	0.011	0.059	0.036	0.527
SC	0.002	0.005	0.004	0.503	0.013	0.035	0.027	0.513	0.002	0.01	0.006	0.505	0.01	0.056	0.035	0.524
HP	0.008	0.031	0.024	0.518	0.001	0.002	0.002	0.501	0.007	0.049	0.03	0.527	0.001	0.006	0.003	0.503
HD	0.004	0.009	0.007	0.504	0.012	0.031	0.025	0.513	0.003	0.011	0.008	0.505	0.011	0.055	0.034	0.526
LHN	0.001	0.003	0.002	0.501	0.001	0.002	0.001	0.5	0.001	0.004	0.003	0.502	0.001	0.003	0.002	0.501
AA	0.025	0.073	0.062	0.54	0.017	0.048	0.039	0.522	0.018	0.098	0.071	0.552	0.013	0.078	0.049	0.537
PA	0.001	0.001	0.001	0.5	0.001	0.001	0.001	0.5	0.002	0.009	0.004	0.503	0.002	0.009	0.004	0.503
RA	0.02	0.06	0.051	0.534	0.015	0.044	0.033	0.522	0.015	0.084	0.06	0.545	0.013	0.073	0.043	0.536
LB	0.009	0.028	0.024	0.514	0.025	0.064	0.054	0.532	0.007	0.044	0.029	0.522	0.02	0.099	0.067	0.55
N	15								20							
Rank	I				All				I				All			
	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC
ILP	0.019	0.151	0.096	0.579	–	–	–	–	0.015	0.155	0.097	0.589	–	–	–	–
CN	0.014	0.113	0.079	0.56	0.011	0.093	0.054	0.549	0.012	0.127	0.082	0.567	0.01	0.109	0.058	0.559
JC	0.013	0.112	0.077	0.559	0.011	0.093	0.054	0.549	0.012	0.125	0.081	0.566	0.01	0.109	0.058	0.559
SI	0.002	0.014	0.008	0.507	0.01	0.076	0.041	0.536	0.002	0.018	0.009	0.509	0.009	0.091	0.045	0.545
SC	0.002	0.015	0.007	0.509	0.009	0.071	0.039	0.533	0.002	0.02	0.009	0.512	0.008	0.081	0.042	0.539
HP	0.006	0.063	0.034	0.535	0.001	0.01	0.004	0.505	0.005	0.073	0.037	0.542	0.001	0.016	0.005	0.508
HD	0.002	0.014	0.009	0.507	0.009	0.073	0.039	0.535	0.002	0.016	0.009	0.508	0.008	0.085	0.042	0.543

Table 2 (continued)

N	15								20							
Rank	I				All				I				All			
	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC
LHN	0.001	0.007	0.003	0.504	0.001	0.005	0.002	0.502	0.001	0.01	0.004	0.505	0.001	0.008	0.003	0.504
AA	0.015	0.117	0.077	0.561	0.012	0.099	0.055	0.552	0.012	0.129	0.08	0.57	0.011	0.117	0.06	0.564
PA	0.001	0.01	0.004	0.504	0.001	0.01	0.004	0.504	0.001	0.01	0.004	0.504	0.001	0.01	0.004	0.504
RA	0.012	0.104	0.066	0.555	0.011	0.092	0.049	0.548	0.01	0.116	0.069	0.561	0.01	0.11	0.053	0.56
LB	0.007	0.06	0.034	0.53	0.018	0.131	0.076	0.565	0.007	0.076	0.038	0.538	0.017	0.163	0.084	0.582

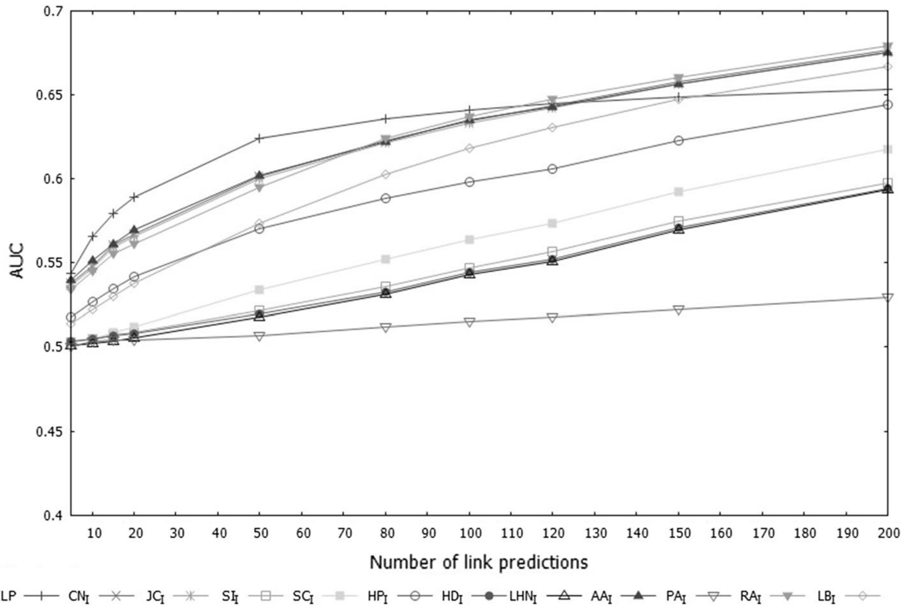


Fig. 3 Comparison of AUC measure between ILP and baselines generating New_I (Flixster dataset)

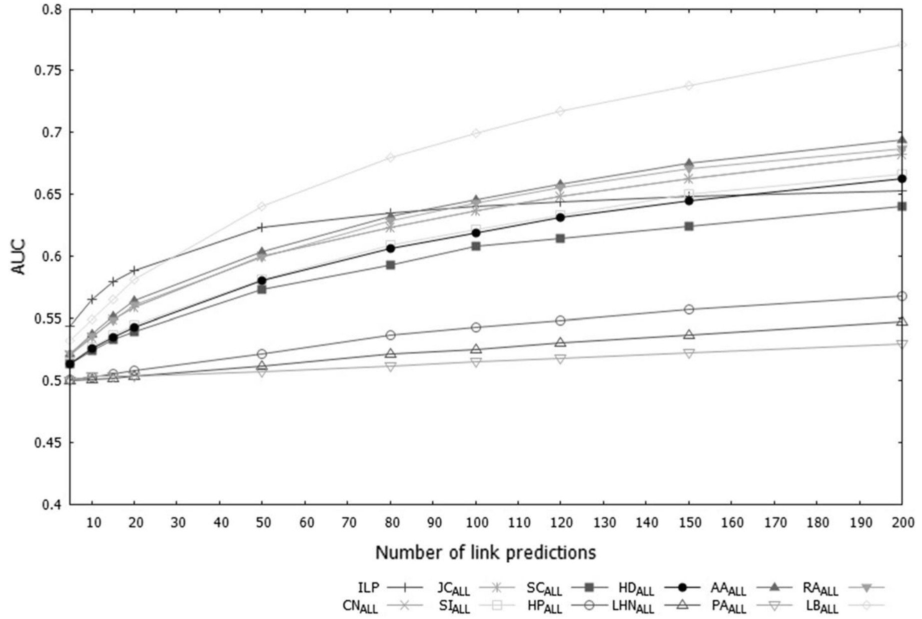


Fig. 4 Comparison of AUC measure between ILP and baselines generating New_{All} (Flixster dataset)

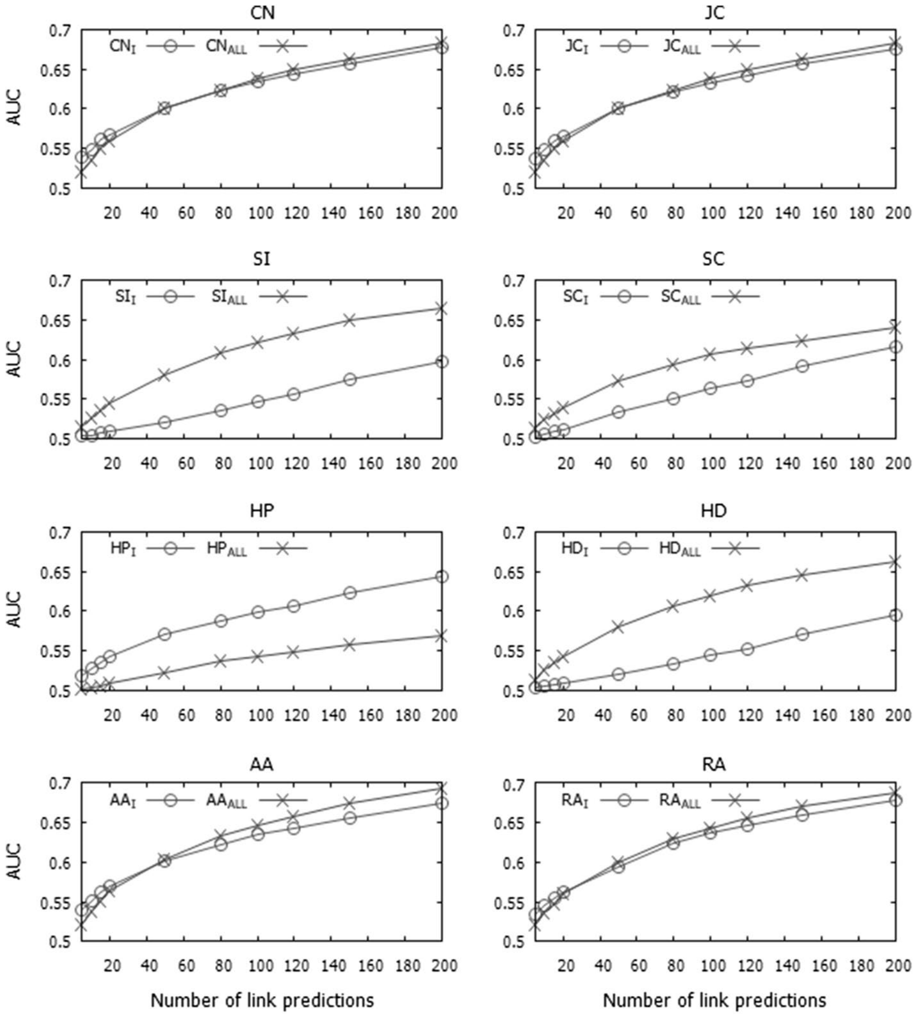


Fig. 5 Comparison of AUC measure between topological metrics using New_I and New_{All} (Flixster dataset)

contrast, the learning-based approach overcame ILP results for more than 20 predictions. However, it is important to remark that due to execution time limitation the testing set used to test LB was significantly reduced. This reduction clearly improved the performance of the LB approach. For this reason, we think that the AUC will be worse with the full testing set.

Finally, Fig. 5 shows a comparison between topological metrics building New_I and New_{All}. As we can see, the approach using New_I obtained better results than using New_{All} for metrics CN, JC, HP, AA and RA when $n \leq 20$. However, when $n > 20$, the AUC measure obtained using New_{All} outperformed the AUC measure obtained using New_I. As with ILP, this happens because the maximum number of link predictions that the metrics can recommend for New_I is less than for New_{All}. In turn, this is because $I_{Target} - F_i$ (used to generate New_I) is less than $\Gamma(Target) - F_i$ (used to generate New_{All}). Nevertheless, New_I for HP metric was not affected since the HP value is determined by the lower degree of nodes (Wang et al. 2015). In contrast, HD obtained better results using New_{All}, since HD value is determined by the higher degrees of nodes. The rest of the metrics showed a worse performance generating New_I.

4.3.2 Higgs dataset

The results obtained with the Higgs dataset were more promising than those obtained with Flixster. Table 3 shows precision, recall, nDCG and AUC measures for 5, 10, 15 and 20 predictions. Similar to Flixster dataset, ILP obtained the best performance. The best precision was 6.9% obtained with ILP in 5 predictions (an improvement of 97.14% compared to the result obtained by New_I^{LB} also in 5 predictions), while the best recall was 35.8% obtained with ILP in 20 predictions (an improvement of 47.93% compared to the result obtained by New_I^{LB} also in 20 predictions).

Comparing the results obtained by the topological metrics with New_I and New_{All}, we observed the same patterns that using Flixster. That is, the best results was obtained with New_I. However, contrary to what happens with Flixster dataset, the learning-based approach presented better performance also generating New_I than New_{All}. We think that this is also related to the best results obtained by ILP with Higgs dataset.

Figures 6 and 7 compare the AUC measure obtained by ILP with the baseline approaches generating New_I and New_{All}, respectively. These figures also show that the best results were obtained with ILP, particularly, when the number of predictions was lower than 150. As with the experiments with Flixster, the AUC obtained by ILP grew rapidly with few predictions, but then slowed its growth.

Finally, Fig. 8 also shows a comparison of AUC metric obtained by topological metrics generating New_I and New_{All}. These results were similar to those obtained with Flixster dataset. However, we can observe a more significant difference between the AUCs when the curve generated with New_I overcame the curve generated with New_{All}, such is the case of CN, JC, AA and RA.

5 Conclusions and future work

In short, we highlight two main contributions of this work. First, we presented a new approach to predict links to influential users. Additionally, we presented an experimental analysis that shows that topological metrics have a better performance predicting links to influential users when they are applied over the set of current influential users of the target. We found that with ILP we can improve the link prediction performance with respect to classical topological metrics and learning-based approaches. On the other hand, one of the limitations of our approach is that the target must have influential users in his/her current neighborhood to be able to receive recommendations of other influential links. Moreover, as shown by the AUC metric, ILP predicts a limited number of new links. For this reason, our approach showed a better performance than topological metrics when the number of recommendations is lower than ~ 100 recommendations, but when this number increases our approach is overcome by classical approaches. We also observed some differences between the datasets used for the experimentation. We think that these differences are related to the role that social influence plays in each social network. Thus, if the social influence is high among users in the social network, our approach will obtain better results.

Table 3 Comparison of precision (P), recall (R), NDCG (N) and AUC for Higgs dataset

N	5								10							
Rank	I				All				I				All			
	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC	P	R	N	AUC
ILP	0.069	0.276	0.234	0.638	–	–	–	–	0.04	0.314	0.248	0.657	–	–	–	–
CN	0.02	0.077	0.062	0.539	0.008	0.031	0.023	0.516	0.014	0.108	0.072	0.554	0.008	0.058	0.033	0.529
JC	0.018	0.07	0.055	0.535	0.008	0.031	0.023	0.516	0.013	0.097	0.064	0.548	0.008	0.058	0.033	0.529
SI	0.006	0.021	0.015	0.51	0.006	0.025	0.02	0.512	0.005	0.031	0.019	0.516	0.007	0.049	0.029	0.525
SC	0.006	0.022	0.017	0.511	0.007	0.024	0.02	0.512	0.004	0.032	0.021	0.516	0.006	0.04	0.026	0.52
HP	0.009	0.035	0.025	0.517	0.001	0.003	0.002	0.501	0.008	0.065	0.035	0.533	0.001	0.008	0.004	0.504
HD	0.005	0.018	0.014	0.509	0.008	0.028	0.021	0.514	0.004	0.028	0.018	0.514	0.006	0.044	0.027	0.522
LHN	0.004	0.018	0.013	0.509	0.001	0.003	0.003	0.501	0.004	0.026	0.016	0.513	0	0.003	0.003	0.502
AA	0.014	0.054	0.044	0.527	0.007	0.027	0.021	0.514	0.013	0.102	0.061	0.551	0.007	0.054	0.031	0.527
PA	0	0	0	0.5	0	0	0	0.5	0	0	0	0.5	0	0	0	0.5
RA	0.014	0.057	0.047	0.528	0.005	0.017	0.014	0.508	0.01	0.072	0.053	0.536	0.004	0.024	0.017	0.512
LB	0.035	0.118	0.09	0.559	0.02	0.064	0.05	0.532	0.027	0.183	0.112	0.591	0.014	0.091	0.059	0.545

N	15								20							
Rank	I				All				I				All			
	P	R	N	AUC	P	R	N	P	P	R	N	AUC	P	R	N	AUC
ILP	0.029	0.344	0.256	0.672	–	–	–	–	0.023	0.358	0.26	0.679	–	–	–	–
CN	0.013	0.142	0.082	0.571	0.006	0.064	0.034	0.532	0.01	0.156	0.085	0.578	0.006	0.078	0.038	0.539
JC	0.012	0.127	0.073	0.563	0.006	0.064	0.034	0.532	0.01	0.141	0.076	0.57	0.006	0.078	0.038	0.539
SI	0.004	0.038	0.02	0.519	0.005	0.058	0.031	0.529	0.004	0.047	0.023	0.523	0.005	0.069	0.034	0.535
SC	0.004	0.041	0.023	0.52	0.005	0.057	0.03	0.528	0.005	0.062	0.028	0.531	0.005	0.067	0.033	0.533
HP	0.008	0.09	0.042	0.545	0.001	0.009	0.004	0.505	0.007	0.11	0.047	0.555	0.001	0.011	0.005	0.506
HD	0.003	0.034	0.019	0.517	0.005	0.053	0.029	0.527	0.003	0.039	0.021	0.52	0.005	0.063	0.032	0.531

Table 3 (continued)

N	15								20							
Rank	I				All				I				All			
	P	R	N	AUC	P	R	N	P	P	R	N	AUC	P	R	N	AUC
LHN	0.003	0.032	0.018	0.516	0.001	0.006	0.004	0.503	0.002	0.034	0.018	0.517	0.001	0.006	0.004	0.503
AA	0.011	0.126	0.068	0.563	0.006	0.066	0.034	0.533	0.011	0.161	0.076	0.581	0.005	0.075	0.036	0.537
PA	0	0.001	0	0.501	0	0.001	0	0.501	0	0.001	0	0.501	0	0.001	0	0.501
RA	0.009	0.101	0.061	0.551	0.003	0.032	0.019	0.516	0.008	0.12	0.066	0.56	0.003	0.037	0.02	0.519
LB	0.021	0.216	0.122	0.608	0.012	0.117	0.066	0.558	0.017	0.242	0.128	0.621	0.01	0.137	0.071	0.568

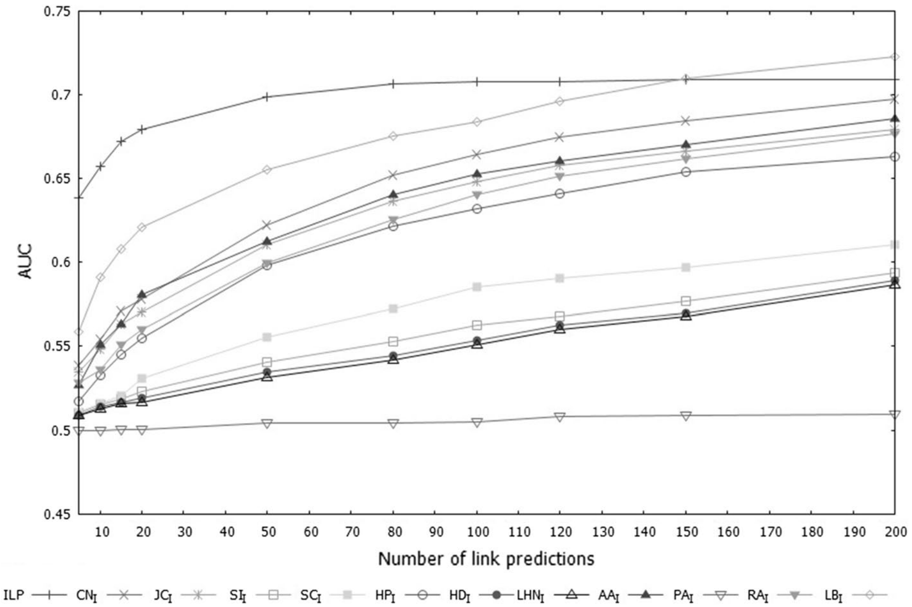


Fig. 6 Comparison of AUC measure between ILP and baselines generating New_I (Higgs dataset)

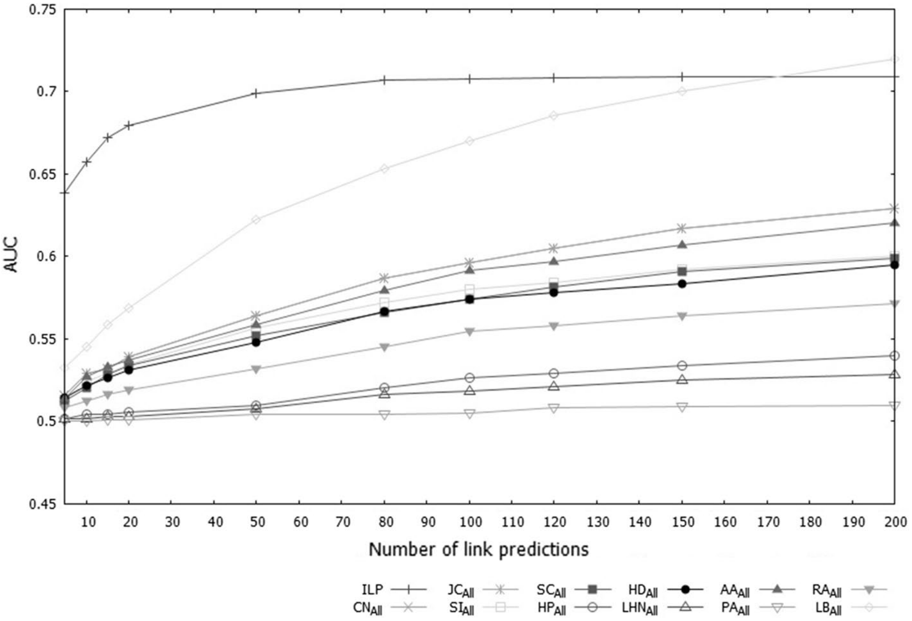


Fig. 7 Comparison of AUC measure between ILP and baselines generating New_{All} (Higgs dataset)

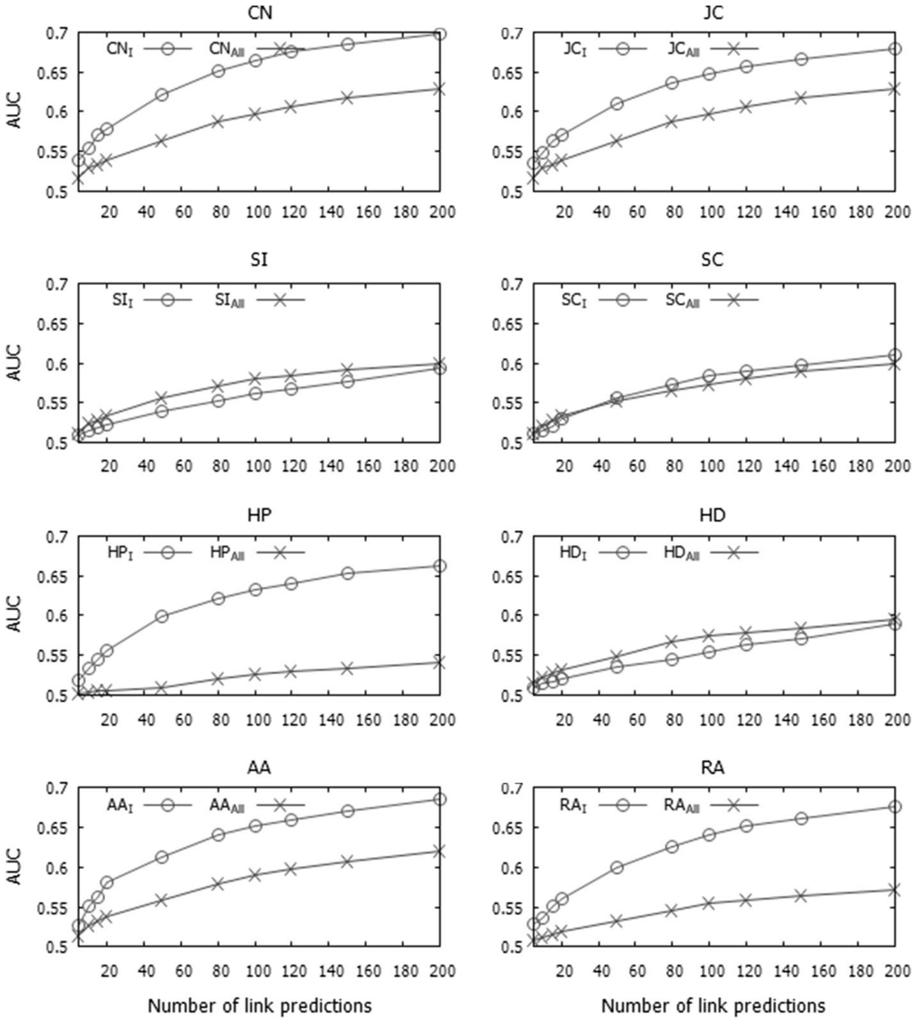


Fig. 8 Comparison of AUC measure between topological metrics using New_I and New_{All} (Higgs dataset)

Future work will focus on enriching ILP with content-based information in order to recommend links to influential users in specific topics or domains.

Acknowledgements This research was partially supported by ANPCyT through PICT Project No. 2014-2750.

References

Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2012). Effects of user similarity in social media. In *Proceedings of the fifth ACM international conference on web search and data mining, WSDM '12* (pp. 703–712). New York, NY: ACM.

Aral, S., Muchnik, L., & Sundararajan, A. (2009). Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, *106*(51), 21544–21549.

Armentano, M. G., Godoy, D., & Amandi, A. (2012). Topology-based recommendation of users in micro-blogging communities. *Journal of Computer Science and Technology*, *27*(3), 624–634.

Armentano, M. G., Godoy, D., & Amandi, A. A. (2013). Followee recommendation based on text analysis of micro-blogging activity. *Information Systems*, *38*(8), 1116–1127.

Bakshy, E., Hofman, J. M., Mason, W. A., & Watts, D. J. (2011). Everyone's an influencer: Quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on web search and data mining, WSDM '11* (pp. 65–74). New York, NY: ACM.

Bhattacharyya, P., Garg, A., & Wu, S. F. (2010). Analysis of user keyword similarity in online social networks. *Social Network Analysis and Mining*, *1*(3), 143–158.

Bonchi, F. (2011). Influence propagation in social networks: A data mining perspective. In *2011 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (WI-IAT)* (Vol. 1, p. 2).

Chen, H.-H., Gou, L., Zhang, X. L., & Giles, C. L. (2012). Discovering missing links in networks using vertex similarity measures. In *Proceedings of the 27th annual ACM symposium on applied computing* (pp. 138–143).

Chiang, K.-Y., Natarajan, N., Tewari, A., & Dhillon, I. S. (2011). Exploiting longer cycles for link prediction in signed networks. In *Proceedings of the 20th ACM international conference on information and knowledge management* (pp. 1157–1162).

Choudhury, N., & Uddin, S. (2017). Evolution similarity for dynamic link prediction in longitudinal networks. In B. Gonçalves, R. Menezes, R. Sinatra, & V. Zlatic (Eds.), *Complex Networks VIII* (pp. 109–118). Cham: Springer.

Choudhury, N., & Uddin, S. (2018). Evolutionary community mining for link prediction in dynamic networks. In C. Cherifi, H. Cherifi, M. Karsai, & M. Musolesi (Eds.), *Complex Networks and Their Applications VI* (pp. 127–138). Cham: Springer.

Dai, C., Chen, L., & Li, B. (2017). Network link prediction based on direct optimization of area under curve. *Applied Intelligence*, *46*(2), 427–437.

De Domenico, M., Lima, A., Mougel, P., & Musolesi, M. (2013). The anatomy of a scientific rumor. *Scientific Reports*, *3*, 02980.

Ding, J., Jiao, L., Jianshe, W., & Liu, F. (2016). Prediction of missing links based on community relevance and ruler inference. *Knowledge-Based Systems*, *98*, 200–215.

Fouss, F., Pirotte, A., Renders, J.-M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *19*(3), 355–369.

Goyal, A. (2013). *Social influence and its applications: An algorithmic and data mining study*. Ph.D. thesis, University of British Columbia.

Goyal, A., Bonchi, F., & Lakshmanan, L. V. S. (2010). Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on web search and data mining, WSDM '10* (pp. 241–250). New York, NY: ACM.

Goyal, A., Bonchi, F., & Lakshmanan, L. V. S. (2011). A data-based approach to social influence maximization. *PVLDB*, *5*(1), 73–84.

Goyal, A., Lu, W., & Lakshmanan, L. V. S. (2011). Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In *20th international world wide web conference on proceedings of WWW 2011* (pp. 47–48).

Güneş, İ., Gündüz-Öğüdücü, Ş., & Çataltepe, Z. (2016). Link prediction using time series of neighborhood-based node similarity scores. *Data Mining and Knowledge Discovery*, *30*(1), 147–180.

Ienco, D., Bonchi, F., & Castillo, C. (2010). The meme ranking problem: Maximizing microblogging virality. In *2010 IEEE international conference on data mining workshops (ICDMW)* (pp. 328–335).

Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on recommender systems, RecSys '10* (pp. 135–142). New York, NY: ACM.

Jiang, J., Shi, P., An, B., Jianyong, Y., & Wang, C. (2017). Measuring the social influences of scientist groups based on multiple types of collaboration relations. *Information Processing & Management*, *53*(1), 1–20.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, *18*(1), 39–43.

Kempe, D., Kleinberg, J., & Tardos, É.(2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '03* (pp. 137–146). New York, NY: ACM.

La Fond, T., & Neville, J. (2010). Randomization tests for distinguishing social influence and homophily effects. In *Proceedings of the 19th international conference on world wide web, WWW '10* (pp. 601–610). New York, NY: ACM.

Liben-Nowell, D., & Kleinberg, J. (2003). The link prediction problem for social networks. In *Proceedings of the 12th international conference on information and knowledge management (CIKM).*

Li, X., & Chen, H. (2013). Recommendation as link prediction in bipartite graphs. *Decision Support Systems*, *54*(2), 880–890.

Liu, D., Wang, L., Zheng, J., Ning, K., & Zhang, L.-J. (2013). Influence analysis based expert finding model and its applications in enterprise social network. In *2013 IEEE international conference on services computing* (Vol. 0, pp. 368–375).

Lü, L., Jin, C.-H., & Zhou, T. (2009). Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, *80*(4), 046122.

Lü, L., & Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A*, *390*(6), 11501170.

Monteserin, A., & Amandi, A. (2015). Whom should I persuade during a negotiation? An approach based on social influence maximization. *Decision Support Systems*, *77*, 1–20.

Monteserin, A., & Armentano, M. G. (2018). Influence-based approach to market basket analysis. *Information Systems* (in press).

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web*. Technical report, Stanford InfoLab.

Perlich, C., Swirszcz, G., & Lawrence, R. (2009). *Content-based link prediction for patent marketing*. Technical report, IBM Research Division.

Pobiedina, N., & Ichise, R. (2016). Citation count prediction as a link prediction problem. *Applied Intelligence*, *44*(2), 252–268.

Rahman, M., & Hasan, M. A. (2016). Link prediction in dynamic networks using graphle. In P. Frasconi, N. Landwehr, G. Manco, & J. Vreeken (Eds.), *Machine Learning and Knowledge Discovery in Databases* (pp. 394–409). Cham: Springer.

Rashotte, L. (2007). Social influence. In G. Ritzer (Ed.), *Blackwell encyclopedia of sociology* (Vol. IX, pp. 4426–4429). London: Blackwell.

Scellato, S., Noulas, A., & Mascolo, C. (2011). Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp 1046–1054). ACM.

Scripps, J., Tan, P.-N., Chen, F., & Esfahanian, A.-H. (2008). A matrix alignment approach for link prediction. In *ICPR* (pp. 1–4).

Wang, Y., Wang, L., Li, Y., He, D., Liu, T.-Y., & Chen, W. (2013). A theoretical analysis of NDCG type ranking measures. CoRR arXiv:1304.6480.

Wang, P., BaoWen, X., YuRong, W., & Zhou, X. Y. (2015). Link prediction in social networks: The state-of-the-art. *Science China Information Sciences*, *58*(1), 1–38.

Wortman, J. (2008). *Viral marketing and the diffusion of trends on social networks*. Technical Report No. MS-CIS-08-19, University of Pennsylvania Department of Computer and Information Science.

Ye, M., Liu, X., & Lee, W.-C. (2012). Exploring social influence for recommendation: A generative model approach. In *Proceedings of the 35th international ACM SIGIR conference on research and development in information retrieval, SIGIR '12* (pp. 671–680). New York, NY: ACM.