Friendship Recommendations in Online Social Networks

Giuliana Carullo*, Aniello Castiglione†‡, Alfredo De Santis§
Department of Computer Science, University of Salerno
Via Giovanni Paolo II, 132 I-84084 - Fisciano (SA), Italy
g.carullo3@studenti.unisa.it, castiglione@ieee.org†, castiglione@acm.org‡, ads@unisa.it§

Abstract—Recommendation systems are popular both commercially and in the research community. For example, Online Social Networks (OSNs), like Twitter, are of special attention since a lot of connection are established between users without any previous knowledge. This highlights one of the key features of a lot of OSNs: the creation of relationships between users. Therefore, it is important to find new ways to provide interesting friendships suggestions.

This work is the first step of an in-depth study whose goal is to find the right trade-offs between the number of factors explored in current state-of-the-art research. In particular, our contribution is an approach based on both Hubs And Authorities algorithm and similarity measures. The first one lets us leverage triadic closures while the second one takes into account homophily. Even if the interplay between similarity and social ties is still an open issue in the analysis of OSNs, we lean towards the idea that it really counts.

In order to support this hypothesis, a preliminary evaluation is performed on an implementation of the presented algorithm on datasets from Twitter. The deriving results show promising perspectives in terms of both effectiveness and scalability. These encouraging results are driving our future research efforts.

Index Terms—Recommendation Systems; Hubs And Authorities; HITS; Online Social Networks; Similarity; Twitter; Real-world Sensing Data; Friendship Prediction.

I. INTRODUCTION

In the last few years we assisted to a continuous growth of OSNs, both in terms of numbers and size. Moreover, Social Networks Services (SNSs) such as Facebook, Twitter and Flickr are striving to expand their popularity and importance. Some of them provide a service to recommend friends, even though the method is based on as-yet-undisclosed algorithms. However, the human factors behind how a user gets in touch with others follow complex mechanisms. Hence, in order to propose an effective friendship recommendation system, there is the need to identify the factors that impact on the creation of relationships between users. We claim that friends-of-friends (FOF) relationships and similarity strongly impact the way people interact. In particular, similarity has been detected to be a strong impacting factor when dealing with people’s behavior and spans across several fields including trust management [1] and more specific relationships formation such as cooperation into organizations [2] or buyer-seller relationships [3].

Furthermore, when dealing with friendship, its value is seen from the direction of information exchange [4]. From the weak ties theory, the value of establishing friendships can be considered from two aspects: heterophily and homophily. Heterophily means that people from different backgrounds have differences in communication topics and information sources. Therefore, the more diversified their friends, the broader topics they may get exposed to. Homophily means that users with similar interests and background tends to become friends since people tend to be linked and further discuss some topics with those sharing more attributes with them. Homophily strongly affects the friendship creation process. Several are the factors that help to find similar friends [5] including shared interests, followers and followers. Indeed, people prefer to share knowledge with persons who have common interests with them. The second factor, highlights the FOF relationships creation. In terms of Twitter-like OSNs, this means that the more is the overlap of the follower circle, the more is the value in establishing friends. The last factor, reflects the similarity between two users’ image and attractiveness to the others. It is important to observe that the importance of homophily especially holds for Twitter-like OSNs that, unlike Facebook-like ones, have a strong information propagation power. Indeed, in this kind of networks, a person establish a link if he/she is interested in the same arguments of the person to which he/she is connecting is interested in.

In this paper, we present an algorithm based on the triadic closure concept that suggests new potential friends based on already existing friendships (i.e., FOF). Moreover, in order to avoid the Rich-get-Richer phenomenon, we also apply similarity measures. In particular, we exploit the Hubs And Authorities (H&A) algorithm in order to identify users that are more likely to be relevant to the interest of the target user, and the Tversky index to take into account the interest similarities between the target user and his/her set of friends. While the theoretical part of this work is a structural basis for tie recommendations, the substantive object of study is the evaluation of our model on Twitter. In contrast with networks such as Facebook, privacy, reciprocity and knowledge of the person to which the target user is connecting to are not
fundamental. Indeed, suggesting friends in environments like Twitter goes beyond the direct knowledge of a person since many connections with strangers do not need reciprocation. To confirm this hypothesis, KwaK et Al. [6], conducted a research in which they found that only 22% of all connections on Twitter are reciprocal. Hence, having directed ties presents significant analytical benefits since they inherently contain information on the power of the relationships between users. For this reason, we focus our study on recommendations in this type of environment.

The rest of the paper is organized as follows. Section II presents the related work, while in Section III the overview of the proposed algorithm is sketched. In Section IV and V the algorithm is described in depth. Section VI and Section VII present the metrics we used to evaluate our proposal and the experimental setup, respectively. Section VIII discusses on some experimental results, while final remarks and future work close the paper in Section IX.

II. RELATED WORK

Several recommender systems have been proposed to help Twitter users to interact more easily and share information. Golder et Al. [7] proposed a recommendation system based on the homophily concept. In particular, they leverage “shared interests and audience”, “reciprocity” and “filtered people”. Reciprocity means that is probable that a user will follow back his/her followers just to reciprocate the favor. However, the authors have not validated their model. The paper proposed by Garcia [8] identifies several factors that may be useful for recommending followees: popularity, activity (i.e., the number of tweets since entering the network), location, friends in common and content of tweets. However, they consider only popularity and activity in their analysis. This work has very good performances, but has a drawback: it does not rely on the triadic closure assumption, therefore it is feasible only if the algorithm has access to a big part of Twitter’s network data, that is not feasible for the general public and makes the proposed algorithm harder to scale. The network structure has been considered by Armentano et Al. [9]. Their proposal explores the target user neighborhood in search of candidate recommendations and then ranks them according to different features depending on several factors including the number of followers and followees and shared friends. However, the evaluation has been performed on a small set of target users. Thus, it would need further evaluation. Other approaches based on the structure of the network have been proposed by authors in [10] and [11] which present in both cases a genetic algorithm based on FOF.

III. FRIEND RECOMMENDATION OVERVIEW

In this section, we present the rough steps that we use to construct a list of friendships suggestions for a certain user. In the description of the algorithm we follow the terminology used by Twitter to refer to the type of link (follower/followees).

In general, the algorithm has been designed to exploit triadic closures by suggesting to a user \( U \) new potential friends based on his/her already existing friends. A naïve approach is to suggest users most followed by \( U \)’s friends. However, we know from the Rich-get-Richer phenomenon that a small set of users will have a lot of followers. Thus, the naïve approach will be biased to suggest always the same very popular users. The goal of the proposed algorithm is to avoid skewed suggestions by taking into account for each friend of \( U \) two factors: i) his/her reputation; ii) his/her similarity to \( u \) himself/herself.

The algorithm deriving from these observations is shown in Figure 1 and consists of three components which are:

- Hubs And Authorities - module that runs H&A algorithm on the considered network.
- Similarity Measure - module that computes the similarity between users using the Tversky index.
- Final Suggestions - module that combines output from the other modules to give a final rank at each user in the network.

IV. HUBS AND AUTHORITIES

Hubs And Authorities algorithm (H&A) - also known as Hyperlink-Induced Topic Search (HITS) - is a link analysis algorithm that ranks web pages. It was initially proposed by Jon Kleinberg [12] [13]. The idea behind the H&A algorithm is that certain web pages are sources of information for a given informational query. We call such pages authorities. On the other hand, there are many other pages that are hand-compiled lists of links to authoritative web pages on a given topic. These pages are called hubs and are not themselves authoritative sources of information. Rather, they are compilations that someone interested in the given topic has spent time putting together.

Borrowing from HITS, this module ranks the users in the neighborhood of the target user \( U \) in order to establish which of them are more trustable. Friends of the most trustable users are more likely to be suggested to the user \( U \). This reputation can be considered either locally or globally. The first metric is based on the triadic closure concept: the users in the neighborhood of the node \( U \) tends to be friends of each others.

![Fig. 1. General model](image-url)
The one which has more connections into this subnetwork is the one that receives the higher score. The second metric, simply expands this concept to the whole network. In the following we analyze the details of both metrics.

A. Local Hubs And Authorities

The first part of the algorithm we implemented focuses on suggesting users that are in the neighborhood of the target user $U$. First, we want to estimate the reputation of all $\text{friends}(U)$. The rationale behind this hypothesis is that it is well known from the literature that relationships tend to follow FOF formation. In particular, we rank each user based on its local popularity. Informally, this means that the more my friends trust a peer $V$ (i.e., many of my friends follow $V$), the more probable is my willingness to get connected to it.

We get this information by performing a Local Hubs And Authorities (Local H&A) in order to provide an indication on who are the better source of information in the network of user $U$. This is equivalent to search for the users that are trusted most by the users that the target user trusts.

Local H&A can be summarized in the following steps:
1) Starting with the target user $U$, we first obtain the list of users he/she follows, let’s call this set $F$.

$$F = \bigcup_{fr \in \text{followees}(U)} fr$$

2) Then we compute the set of hubs, namely $H$, as follows:

$$H = \bigcup_{fr \in F} \{ f \in \text{follower}(fr) \mid f \in F \}$$

It is important to observe that while running the H&A algorithm, user $U$ will be a hub that will equally boost the score of all authorities (because, he/she follows all of them).

3) To perform the Local H&A algorithm we consider all friends of the target user as authorities. Let this set be $A$, it is defined as follows:

$$A \equiv F$$

4) Finally, run H&A on $H$ and $A$. As in H&A a normalization step is required. However, instead of dividing each hub core by square root of the sum of the squares of all hub scores, and dividing each Authority score by square root of the sum of the squares of all Authority scores, we have slightly modified it as follows. We first subtract the minimum + 1 from every score and then we divide them for the maximum value computed after subtracting. This is equivalent to scale the scores in the $[0,1]$ range, where 1 is the score of the highest ranked authorities.

At the end of this steps, the scores of the authorities rank users depending on their (local) trustworthiness. Let’s $A_U$ be the set of authorities scores obtained at the end of the Local H&A.

B. Global Hubs And Authorities

The reputation of $\text{followees}(U)$ can also be weighted by considering their reputation based on their followers that are not among $\text{followees}(U)$; in other words, we consider the reputation of each friend of $U$ taking into account the whole network. To do so, we run a Global H&A, that works as follows:

1) The set of authorities $A$ is defined as in the previous case.
2) The set of hubs $H$, is given by the nodes which follow the authorities independently from if they are also followed from user $U$ or not.

$$H = \bigcup_{fr \in F} \text{follower}(fr)$$

The major drawback of this approach is that it may be very expensive in cases where users have millions of followers (e.g., The Economist has 5.34 million followers).

3) Finally run H&A algorithm on $H$ and $A$, considering only authorities scores as in the previous case.

Let $A_0$ be the set of authorities scores obtained at the end of the Global H&A algorithm. It can be observed that the ranking process does not consider $\text{follower}(U)$ as potential users for suggestions for two reasons: i) it makes the algorithm vulnerable to spamming; ii) if they are links to actually interesting users, it is likely that those interesting users are already linked by some $\text{followees}(U)$.

V. Boosting Similarity

Other than solely ranking users depending on the structure of the network, we also want to boost friends’ scores based on the similarity among users. The reason is to let fairly contribute to the final suggestion very important users (e.g., journals Twitter pages) with real life personal friends of the given user. Hence, the idea is to weight the authority score with an affinity score that gauges the similarity of $U$ with his/her friends based on their common friends. Hence, in the following, we first define how our algorithm models similarity. Then, we describe how similarity and results from H&A are combined together to provide final scores.

A. Similarity Measure

This affinity score is based on the idea that if the user $U$ follows a user $V$ and also follows a lot of user $V$’s friends, it is likely that he/she would like to follow also users which are followed by $V$ but which he/she is not directly connected to. This similarity can be measured using the so called Tversky index. This index is asymmetric and can be expressed as follows. Given two sets $X$ and $Y$, the considered index is a number between 0 and 1 such that:

$$Tversky(X,Y) = \frac{|X \cap Y|}{|X \cap Y| + \alpha|X - Y| + \beta|Y - X|}$$
where \( \alpha \) and \( \beta \) are parameters such that \( \alpha, \beta \geq 0 \) and \( \alpha + \beta = 1 \). The asymmetry of this index let express the following evidence: the fact that many friends of \( U \) follow \( V \) is more important than the fact that many friends of \( V \) do not follow \( U \). This reverberates on the choice of the parameter \( \alpha \) and \( \beta \). Given the index \( Tversky(U, V) \), we can model this evidence by properly choosing values for \( \alpha \) and \( \beta \) such that \( \alpha > \beta \). More details on the actual choice of these parameters are presented in Section VII.

### B. Final Suggestions

The final score for each friend \( i \in F \) is given by a linear function depending on both authority scores and affinity score such that:

\[
  r(i) = a \cdot A_i(i) + b \cdot affinity(i) + c \cdot A_y(i)  \quad (1)
\]

The choice of the coefficients \( a \), \( b \) and \( c \) may impact performances. For example, giving more importance to affinity score by using a high value to the coefficient \( b \), may increase the overall effectiveness. The last step is to choose the persons to suggest. This can be done by simply ordering (descending) users depending on their \( r(i) \) score and showing the top \( k \) friends as friendship suggestions. Finally, it can be observed that \( k \) is an implementation-specific parameter.

### VI. PERFORMANCE MEASURES

In this paper we focus on the performances of the algorithm that exploits only Local H&A and similarity. In order to measure the closeness of predictions made by the algorithm to users’ real preferences, a numerical representation is needed. To this end, several metrics have been proposed in the literature [14] [15]. However, accuracy, precision and recall have been recognized to be the most used metrics.

### A. Accuracy

Accuracy is a well-known metric into the field of Artificial Intelligence and it measures the quality of nearness to the truth or the true value achieved by a system. In general terms, it can be formulated as:

\[
  accuracy = \frac{\text{number of good cases}}{\text{number of possible cases}} \quad (2)
\]

When applied to recommendation systems it can be re-written as:

\[
  accuracy = \frac{\text{number of successful recommendations}}{\text{number of possible recommendations}} \quad (3)
\]

We consider a recommendation as a successful recommendation if the recommended relationship is close to the user’s real willingness to establish a connection with that user.

### B. Precision and Recall

In recommendation systems, for the user is important to receive result as an ordered list of recommendations, from best to worst. However, in certain cases the user does not care much about the exact ordering of the list. In fact, a set of few good recommendations is fine. Bearing this fact into the evaluation of the proposed system, we can apply classic Information Retrieval (IR) metrics: Precision and Recall. This is because IR focus on the retrieval of relevant documents from a pool, which is not far from the related task of the recommendation of interesting friendships from a pool of users.

### TABLE I

**CONFUSION MATRIX**

<table>
<thead>
<tr>
<th></th>
<th>Successful</th>
<th>Non-Successful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieved</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Non-Retrieved</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

To compute these metrics, we consider the confusion matrix in Table I. Given this matrix it is possible to compute Precision and Recall as follows:

\[
  \text{precision} = \frac{a}{a + b} \quad (4)
\]

\[
  \text{recall} = \frac{a}{a + c} \quad (5)
\]

The meaning of these two metrics is intuitive. Recall means that a recommendation system should not return irrelevant results in the top results, although it should be able to return as many relevant results as possible. The Precision is the proportion of top results that are relevant, which is the recommender’s capacity of showing only “Successful” recommendation, while minimizing the mixture of them with “Non-Successful” ones. The Recall is the proportion of all relevant results included in the top results. In other words, the Recall measures the capacity of obtaining all the successful recommendation present in the pool.

### VII. EXPERIMENTAL SETUP

In order to prove the effectiveness of the proposed system, we conducted some experimental results. Our goal is to evaluate our model on ground-truth data. Therefore, we evaluated it on data from Twitter that can be found on snap.stanford.edu [16]. This dataset consists of 81306 users and 1768149 friendships. Recommendations have been computed for all users in the network. We run our tests on a quad-core with hyper-threading Intel Xeon E31240 processor with a base frequency of 3.30 GHz and 8 GB RAM. Recommending users for the whole dataset 10 times took 15 minutes. Thus, a single execution of our algorithm takes 1 and a half minutes.

### A. Holdout Validation

Evaluating a recommendation system without either the interaction of the involved users or having no knowledge about the users’ interest is a difficult task. To overcome these
difficulties whilst validating our system, we use the *Holdout Validation* method. In general terms, this method, also known as *True Validation*, considers a pseudo-randomly chosen subset of the initial sample and use it as testing set. The remaining observations are retained as the training data.

In order to have an overall measure of effectiveness of our system, we run this validation method for all users in belonging to the network, except those users having only one friends. In our settings, we implemented hold out evaluation in the following way: given a user, we randomly hold out the 20% of his/her friends; then, we run our recommendation algorithm considering the remaining friends. We consider our suggestions correct if they recommend users that are in the hold out set.

To iron out outlying results that could be caused by holding out a set of friends that is crucial for good recommendations, we average our performance metrics on 10 runs of our recommendation algorithm, holding out each time a random subset of friends.

### B. Impacting Factors

The testing scenario has many variables that can influence the performance results of our algorithm, namely the number of recommendations presented to the user, Tversky index coefficients and score weighting. The first factor strongly affects the values of precision and recall because, as better explained in Section VII, recommending many users easily boosts recall while decreasing precision, whereas suggesting few users has the opposite effect. As better explained in the following sections, we tested the sensitivity of recall and precision of our proposal by ranking suggestion following the score assigned by our algorithm, and then suggesting only the top ranked x% users. In particular, we focus on the top 1% since usually only few suggestions are presented to the target user.

The second factor, is also interesting to test. Indeed, the performances of our system may vary by changing α and β coefficients. Recall from Subsection V-A that we want to model the asymmetry between the user receiving the suggestion and his/her friend, from which suggestions are taken. We also tested how much the choice of these coefficients impacts on the proposed solution. In particular, we tested for the following values of α: 0.65, 0.75, 0.85, 0.95.

The third factor is a critical feature. Recall from Subsection V-B that the final score assigned to each friend of the node that receives the suggestion is a linear function depending on authority score and affinity score. A first simple test consists of testing if is better to give more importance to similarity with respect to the H&A result. Therefore we tested if a choice of $a = 1$ and $b = 5$ fits better in respect to equally choosing $a$ and $b$ (i.e. $a, b = 1$).

### VIII. EXPERIMENTAL RESULTS

In this section we describe the results of the offline evaluation presented above. Before delving into details with in-depth analysis, we need to do some observations on the considered dataset. In particular, we first analyze the distribution of the number of friends.

![Figure 2. Distribution of the number of friends](image)

Figure 2 shows that the considered distribution follows a power-law. We can see that there are very few users that follow more than 400 users, hence it is hard to collect statistically significant results about the performance of our algorithm for these users. For this reason, in the following we present data regarding only users that follow at most 400 people.

Second, from Figure 3, we derive that the data we consider is particularly noisy. Hence, for the remaining part of the paper, we smooth original data using Bezier interpolation.

![Figure 3. 1% precision normal](image)

Finally, it is important to note that in our case, the definitions of accuracy and precision are the same. Hence, the showed charts consider only precision and recall, since the accuracy scores follow the same trend of precision.

### A. Accuracy, Precision and Recall

Our first result concerns the first two impacting factors presented in Subsection V-B the number of recommendations
presented to the user and Tversky index coefficients. For the sake of brevity and clarity we only present the most significant results.

To begin with, Figures 4 and 5 show how precision and recall vary when the top 1% of suggestions is presented to the users. Overall the different recommendation strategies appear to perform well across the different coefficients, generating precision scores which are almost all near 0.6. For example, for those users who have 50 friends, we have that the precision of about 0.54. This means that the 54% of the suggested users is a friend in which they are actually interested in. However, the recall value is small, hence highlighting that we are suggesting only a small portion of the total successful users. For example, if we consider again those users who have 50 friends, we have that the recall value is about 0.053. This means that we are suggesting only the 5.4% on the possible successful users. It is important to observe that the recall is more noisy than the precision. Recall from Section VI that accuracy and precision values correspond. Bearing this into our minds, we can also see that relevant recommendations tend to be clustered towards the top of recommendation lists since the high accuracy value.

Let us stress that in our scenario, it is more important to provide to users correct suggestions (i.e., having a high precision) than to provide all correct suggestions (i.e., having a high recall). In other words, our algorithm usually suggests a small subset of all the actual friends of a given user, however these suggestions are mostly correct. In a practical realization we would be able to successfully provide a handful of suggestion that are relevant for the target user.

Our second result is about the value of the coefficients used in the Tversky index formula. As can be seen in the presented charts, the general outcome of the experimental comparison is similar for all the values taken by the coefficients.

**B. Score Weighting**

An interesting result concerns the score weighting factor. We checked whether boosting the affinity score helps in improving performances in terms of precision and recall. We first found that, likely the previous analyzed case, for the boosted similarity score there are no substantial differences between the Tversky coefficients. The trend of precision and recall is shown in Figures 6 and 7. It is close to the previous presented one. Therefore, we focus in the following on the comparison between the performances in case of boosted and normal similarity.

![Fig. 4. 1% precision normal](image1)
![Fig. 5. 1% recall normal](image2)

Given the above results, we stress that in our scenario, it is more important to provide to users correct suggestions (i.e., having a high precision) than to provide all correct suggestions (i.e., having a high recall). In other words, our algorithm usually suggests a subset of the actual friends of a given user, however these suggestions are mostly correct. In a practical realization we would be able to successfully provide a handful of suggestion that are relevant for the target user.

However, we suppose that this behavior depends on the number of considered suggestions. Indeed, our hypothesis is that avoiding to boost similarity performs better in this case. This hypothesis will be investigated in a more comprehensive study in our future research.

**IX. CONCLUSION AND FUTURE WORK**

We presented and experimentally evaluated an algorithm based on H&A that exploits similarities to compute friendship suggestions. A dataset with more than 80,000 users was analyzed to test our algorithm. The deriving results showed that boosting the weight of similarities between users lead to recommenders that on the average provide more accurate recommendations. We believe that the presented algorithm may be improved by embedding other similarity measures, including analysis of hashtags, conversational likelihood, retweets, tweet volume and location.

As future work, we first plan to extend this work by considering apart 1% top rank users also additional percentages (i.e., 10%, 25%, 50%, 75%, 100%). In addition, the evaluation of our algorithm on different similarity boosting factors is worth to be investigated. Indeed, this approach may be able to highlight the best way to combine both worlds (triadic
closures and similarity) and let our algorithm perform even better. Another interesting comparison may be to check how the score vary setting the $\alpha$ coefficient to assume a value smaller than 0.50. Other metrics may be applied to validate our proposal, including coverage, E-measure and F-measure. We also plan to implement and evaluate the performances of the algorithm exploiting Global H&A and to validate our proposal on different networks. Finally, we will evaluate strategies to work around structural holes in the follower-followee relationships. In particular, we will explore the hypothesis that suggesting random users, or users that are only remotely linked to the target user, is a viable technique to provide new weak ties, that we know from OSN studies to be very important for the gathering of relevant information among socially heterogeneous communities.

REFERENCES


