Published in: World Wide Web journal.
This version of the article has been accepted for publication, after peer review and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections.
The Version of Record is available online at: https://doi.org/10.1007/s11280-015-0333-5

A triadic closure and homophily-based recommendation system for online social networks

Giuliana Carullo · Aniello Castiglione · Alfredo De Santis · Francesco Palmieri

Abstract Recommendation systems are popular both commercially and in the research community. For example, Online in Social Networks (OSNs) like Twitter, they are gaining an increasing 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. However, mining and analyzing data from large scale Social Networks can become critical in terms of computational resources. This is particularly true in the context of ubiquitous access, where resource-constrained mobile devices are used to access the social network services. To this end, designing architectures/solutions offering the possibility of operating in a Mobile Cloud scenario is of key importance. Accordingly, we present a new recommendation system scheme that tries to find the right trade-offs between the exploitation of the already existing links/relationships and the interest affinities between users. In particular, such scheme is based on an inherently parallel Hubs And Authorities algorithm together with similarity measures that, for scalability purposes, can be easily transposed in a cloud scenario. The first one let us leverage triadic closures while the second one takes into account homophily. The proposal is supported by an extensive performance analysis on publicly available Twitter data. In particular, we proved the effectiveness

G. Carullo · A. Castiglione · A. De Santis
Department of Computer Science, University of Salerno,
Via Giovanni Paolo II, 132, I-84084 Fisciano SA, Italy

G. Carullo
e-mail: g.carullo3@studenti.unisa.it

A. Castiglione
e-mail: castiglione@acm.org; castiglione@ieee.org

A. De Santis
e-mail: ads@unisa.it

F. Palmieri (✉)
Department of Industrial and Information Engineering, Second University of Naples,
Via Roma 29, I-81031 Aversa CE, Italy
e-mail: francesco.palmieri@unina.it

of the proposed recommendation system by using several performance metrics available in the literature which include precision, recall, F-measure and G-measure. The results show encouraging perspectives in terms of both effectiveness and scalability, that are driving our future research efforts.

Keywords Recommendation systems · Hubs and authorities (HITS) · Online social networks (OSNs) · Similarity · Mobile cloud computing · Twitter

1 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 condition the way people interacts. In particular, similarity has been detected to be a characterizing factor of paramount importance when dealing with people's behavior, spanning across several fields that include trust management [7] and enabling the formation of more specific relationships such as cooperation into organizations [24] or buyer-seller [11].

Furthermore, when dealing with friendship, it can be evaluated by observing the direction of information exchange [30]. 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 tend to become friends since people are likely to be linked and further discuss some topics with those sharing more attributes and common interests with them. Homophily strongly affects the friendship creation process. Several are the factors that are helpful in finding similar friends [23] including shared interests, followees 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 followee circle, the more is the likelihood of establishing friends. The last factor, reflects the similarity between users' image and attractiveness to the other ones. 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 establishes a link if he/she is interested in the same arguments that the person to which he/she is connecting is interested in.

When approaching problems like this one, dealing effectively with the large amount of involved data is of key importance. This is motivated by two non-negligible factors. First, traditional solutions may not scale well. Second, it is not possible to overlook the growing usage of mobile and ubiquitous devices to access any sort of social network-based online applications. Therefore, an architectural solution that easily fits new distributed computing models and processing paradigms such as Clouds and map-reduce, in order to benefit of their elasticity and horizontal scaling features, becomes of fundamental importance when designing recommendation systems in the context of OSNs.

The main contributions of this work are:

- We structured our recommendation system proposal by exploiting the triadic closure concept in social networks, that suggests new potential friends based on already existing friendship relations (i.e., FOF). Moreover, in order to avoid the *Rich-get-Richer* phenomenon, characterizing such kind of scale-free organizations, we also consider several similarity measures among users in the recommendation process, as preliminarily presented in [8]. In detail, the core concept is leveraging the Hubs And Authorities (H&A) algorithm in order to identify users/entities 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.
- The substantive object of this study is an in-depth evaluation of the proposed system on publicly available Twitter data. 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. [21], conducted a research in which they found that only 22 % of all connections on Twitter are reciprocal. Hence, having directed ties presents significant analytic 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. We extended our previous findings by also analyzing two important metrics: F-measure and G-measure. Indeed, only analyzing precision and recall separately may not be sufficient to evaluate the overall performance of a system. Moreover, we figured out how F-measure and G-measure vary depending on the number of friends each user has.
- We argued that approaching this problem in the context of Cloud Computing services available in a Mobile/Ubiquitous access scenario is the only viable and effective solution to cope with its obvious scalability challenges. This, will drive our future research.

The rest of the work is organized as follows. Section 2 presents the related efforts available in literature, while in Section 3 some basic background properties of OSN organizations, motivating the fundamental choices behind the proposal are reported. Section 4 presents the architectural overview of the recommendation system, followed by its performance evaluation and analysis whose layout and results are reported in Section 5. Final remarks and future works close the article in Section 6.

2 Related work

Several recommender systems have been proposed to help Twitter users to interact more easily and share information. Golder et Al. [16] 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 [15] 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. [3]. 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 friend recommendation approaches based on the structure of the network and FOF relations have been proposed in [25] and [29], presenting respectively a social genome-based scheme and a genetic algorithm for selecting users from the FOF group.

3 Basic online social network dynamics

OSNs represent a new generation of information sharing infrastructures, that while providing the traditional content/contact search and organization facilities, are substantially different from well-known network-organization such as the World-Wide-Web. In fact, unlike the Web, that is mainly organized around content sharing, OSNs are structured around the users and their mutual contacts and social relationships, where the users' interests, and endorsements/recommendations become fundamental drivers for effectively locating contributed contents and knowledge. Unfortunately, while a lot of information is available on the Web network properties, and on the dynamics that govern the hyperlinks between Web contents, a limited knowledge is available about the more intrinsic characteristics of online social network structure and their evolution on a wide scale, together with the properties that can be leveraged in OSN-based information systems.

Accordingly, an in-depth understanding of the structure characterizing relationships in online social networks, together with the processes shaping them, is necessary to design effective architectures, algorithms and mechanisms for inferring the degree of shared interests between two users, detecting the most trusted, reputable or authoritative users, and/or discovering information or influential sources based on their recommendations/suggestions.

The connectivity structure of these organizations can be described by a graph whose nodes are associated to users/entities and the edges represent the relations (followee, follower, friend, etc.) between them. For generality sake these relations may be considered as directed ($A \rightarrow B$ does not imply $B \rightarrow A$), based on the non-reciprocal or non-symmetric nature of human relationships. This is the case, for example, of friendship relations on Twitter and Google+, whereas in Facebook and LinkedIn friendship is considered to be symmetric.

The behavior of OSNs is is hence determined by the mesh structure through which the network components are related each other.

3.1 Triadic closure

Friendship is one of the most common relations between entities/users in an OSN. People tend to have friends who are also friends with each other, resulting in sets of user nodes among which many edges exist in the OSN graph. This consideration is at the basis of the triadic closure concept.

Triadic closure is a very simple, intuitive and natural property, than can be easily observed in the behavior of most of the online social network organizations, and exploited by all the activities leveraging the relations between networked entities. In detail, if two entities within a social network have a friend in common, then there is an increased likelihood that they will become friends themselves in the future, and such likelihood grows with the number of common friends. This can be also considered from an alternate perspective: the closer are the relations between two entities and their mutual friends, the higher is their potential of becoming friends. In terms of network graph topology, such kind of transitivity implies the creation of a significant number of triangles within three vertexes sets each of which is connected to both the others. The name "triadic closure" derives from the fact that the above relation has the effect of "closing" the third side of the triangle between the involved entities. When observing the status of the social network graph at different times, we can appreciate in the last observation a significant number of new edges that are generated by such triangle-closing operation between the entities who had a common neighbor in the earlier observation. The fundamental role of triadic closure in OSNs has fostered the formulation of several simple social network metrics representing its degree of prevalence. The most famous of these is the clustering coefficient, defined as the probability that two randomly selected friends of a specified entity are friends with each other. More specifically, it can be seen as the fraction of couples of friends connected to each other by edges in the social network graph, or in other words, the number of triples having their third edge filled in order to complete a triangle. More formally, according to [26], the clustering coefficient can be seen as the probability two friends of a node know each other, defined as:

$$C = \frac{3 \cdot T}{V} \tag{1}$$

where T is the number of triangles in the network and V is the number of connected triples of nodes.

In general, the clustering coefficient in an OSN is a real number assuming values in the interval [0, 1] that becomes zero when there is no clustering at all, and one for maximal clustering, that is, when the network consists only of disjoint cliques.

3.2 Power laws and preferential attachment

Triadic closure is not the most immediate and surprising properties of OSNs. Several research experiences, available in literature, have shown that many real-world network organizations, including social networks exhibit power law attributes and scale-free behavior in their degree distribution [1, 2, 4, 22]. Specifically, the degree distribution $P(k)$, which is the probability of an arbitrary node to be connected to exactly k other ones, can be described by:

$$P(k) = ck^{-\lambda}, \quad k \geq m \tag{2}$$

with an exponent $2 < \lambda < 3$; where c is a normalization factor and m is the minimal number of relations on each node (usually taken to be $m = 1$). Here, we can also evidence that

$$\sum_{k=2}^{k_{max}} P(k) = 1 \tag{3}$$

where k_{max} is the maximum number of neighbors for each node within the network.

In other words, within almost all the online social network organizations, users tend to create a large, strongly connected backbone constituted of high-degree nodes assuming the role of "focal points", surrounded by many small clusters made of low-degree nodes. More specifically, in these organizations we can observe the existence of a huge number of nodes connected with a few edges, whereas only a small number of nodes (known as hubs) with a great number of edges keep the network connected. This implies a significant reduction in the number of hops between nodes and hence in network diameter (*small–world* property) [5, 14] associated to a high clustering coefficient.

The fundamental dynamics describing such a continuously evolving scheme follow the principle of preferential attachment, also known as cumulative advantage or "rich get richer" phenomenon, saying that the likelihood of a node being attached to a new link is proportional to the node's degree. That is, each node strives to establish new connections with neighbors characterized by an higher degree. More precisely, the probability p_i of attaching to a specific node i, with degree k_i, is given by

$$p_i = \frac{k_i}{\sum_{j \in N} k_j}. \tag{4}$$

where N is the complete set of available neighbor nodes with their associated degree k_j. All the nodes accepting connections or requiring new ones compete for new links.

In other words, the node degree determines the attractiveness of a node within the network evolution context. New users establish new associations preferentially to strongly connected "hub" nodes, characterized by an high number of links, forcing the power laws to hold into the resulting organization. According to the considerations reported in [12], in presence of a constantly uniform node deletion rate that remains significantly lower than the insertion one, these preferential attachment-based schemes tend to form a scale-free network with exponent λ slightly lower than 3.

4 The recommendation system architecture

In this section, we describe the recommendation system scheme by presenting the rough steps that lead to the construction of a list of friendships suggestions for a certain user/entity. In the description of the proposed architecture we follow the terminology used by Twitter when referring to the type of link (*follower/followee*).

In general, the proposed scheme 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 that, due to the *Rich-get-Richer* phenomenon, characterizing such kind of organizations, that a small set of users (assuming the role of hub nodes in a scale-free network) will acquire a lot of followers, whereas most of the other ones will be characterized by a very small (one or two) number or followers. In other words, a clustering phenomenon, characterized by the emergence of a connected component, can be observed around all these focal nodes that tend to acquire new followers at a higher rate than the other ones and maintaining them over time. Clearly, such behavior adversely conditions the aforementioned naïve recommendation approach by implicitly biasing it to suggest always the same very popular users. Accordingly, the goal of the proposed scheme is avoiding such skewed suggestions by taking into account for each friend of u two factors: i) his/her reputation, based on its trust degree/number of followers; ii) his/her similarity to u, based on a properly chosen affinity score.

The recommendation system architecture derived from these observations consists of three fundamental components which are:

– Hubs And Authorities - running H&A/HITS algorithm on the involved network, in order to analyze each user and determine the more trustable ones in his neighborhood as candidates to be suggested.

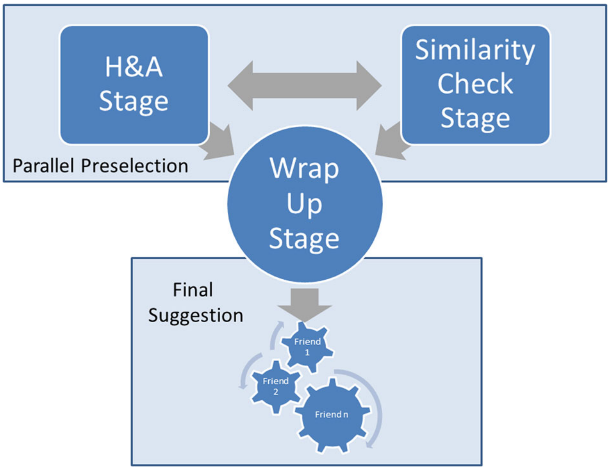


Figure 1 General three-stages system architecture

- Similarity Check - computing the similarity/affinity score between users by using the Tversky index, needed for refining the previous selection.
- Wrap Up - combining output from the previous components by finally ranking each user in the network in order to perform recommendation.

Each component can be considered as a specific step of a three-stages architecture where the former two stages can be partially run in parallel and the latter one depends on the results of the previous stages. Furthermore, a *mapreduce*-based processing structure within the context of a distributed Hadoop cluster can significantly improve the performance of the HITS algorithm on large graphs. The overall architecture is graphically sketched in Figure 1.

4.1 Initial ranking: the hubs and authorities stage

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 [19, 20]. 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 component 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. The one which has more connections into this sub-network 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.

4.1.1 Considering local hubs and authorities

The first component of the proposed scheme focuses on suggesting users that are in the neighborhood of the target user u. First, we want to estimate the reputation of all u's friends defined as the set of elements that follow u and are followed by 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 u 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 $u \in U$, where U is the set of all the OSN users, we first obtain the set $F(u)$ of users he/she follows:

$$F(u) = \bigcup_{f \in U} \{f \mid u \mapsto f\} \tag{5}$$

where $\mapsto$ represents the *follow* relationship between users.

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

$$H(u) = \bigcup_{g \in F(u)} \{f \in \Phi(g) \mid f \in F(u)\} \tag{6}$$

where $\Phi(g)$ is the set of followers associated to the user g, defined as:

$$\Phi(u) = \bigcup_{f \in U} \{f \mid f \mapsto u\} \tag{7}$$

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, as authorities, the set of all friends of the target user u. Let this set be defined as:

$$A(u) = \bigcup_{g \in \Phi(u)} \{f \in F(g) \mid f \in \Phi(u)\} \tag{8}$$

that is, Hubs and Authorities are defined in terms of one another according to a mutual recursion relation.

4. Finally, run H&A on $H(u)$ and $A(u)$ for score calculation. As in H&A a normalization step is required. However, instead of dividing each hub score by the square root of the sum of the squares of all hub scores, and dividing each Authority score by the 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 subtraction. This is equivalent to scale the scores in the $[\epsilon, 1]$ 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 A_l be the set of authorities scores obtained at the end of the Local H&A so that $A_l(i)$ indicates the local authority score of the i-th follower.

4.1.2 Considering global hubs and authorities

The reputation of $F(u)$ can also be weighted by considering their reputation based on their followers that are not among $F(u)$; 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 initial authorities $A(u)$ is defined as in the previous case (see (8)).
2. The set of hubs $H(u)$, is given by the nodes which follow the authorities independently from if they are also followed from user u or not.

$$H(u) = \bigcup_{g \in F(u)} \Phi(g) \tag{9}$$

   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 $H(u)$ and $A(u)$, considering only authorities scores as in the previous case.

Let A_g be the set of authority scores obtained at the end of the Global H&A algorithm, so that $A_g(i)$ indicates the global authority score of the i-th follower. It can be observed that the ranking process does not consider $\Phi(u)$ 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 $F(u)$.

4.2 Ranking refinement: performing similarity check

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 letting very important users with real life personal friends (e.g., journals' Twitter pages), to fairly contribute to the final suggestion. 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 scheme models similarity. Then, we describe how similarity and results from H&A are combined together to provide final scores.

This affinity score is based on the idea that if a 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. More formally, let $X = F(u)$ and $Y = F(v)$, the similarity between X and Y 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:

$$Tversky(X, Y) = \frac{|X \cap Y|}{|X \cap Y| + \alpha|X - Y| + \beta|Y - X|} \tag{10}$$

where α and β are parameters such that $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$. The asymmetry of this index also reflects the natural asymmetry between hubs and authorities. Let us 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 α and β. Given the index $Tversky(X, Y)$, we can model this evidence by properly choosing values for α and β such that $\alpha > \beta$. More details on the actual choice of these parameters are presented in Section 5.2.

4.3 The final wrap up stage

The final score $r(i)$ for each friend $i \in A(u)$ (referred to the target user u as defined in (8)) is given by a linear combination of local and global authority scores together with the affinity score $\Psi(\cdot)$ of i respect to u, so that:

$$r(i) = a \cdot A_l(i) + b \cdot \Psi(i) + c \cdot A_g(i) \tag{11}$$

where the affinity $\Psi(i)$ of i with the target user u is such that:

$$\Psi(i) = Tversky(F(u), F(i)) \tag{12}$$

It should be observed that, the choice of the coefficients a, b and c may impact performances. For example, giving more importance to the affinity score $\Psi(i)$ 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 (in a descending way) 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, limiting the number of suggestions.

5 Performance evaluation and results analysis

In order to prove the effectiveness of the proposed recommendation system and analyze its performances, we conducted several experiments and collected and analyzed the obtained results.

5.1 Performance metrics

First of all, it is necessary to measure the closeness of predictions made by the associated three-stages scheme to users' real preferences, by using a numerical representation of the observed behavior. To this end, several performance metrics have been proposed in the literature [17, 28]. However, accuracy, precision and recall have been recognized to be the most used metrics.

5.1.1 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:

$$accuracy = \frac{number \quad of \quad good \quad cases}{number \quad of \quad possible \quad cases} \tag{13}$$

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

$$accuracy = \frac{number \quad of \quad successful \quad recommendations}{number \quad of \quad possible \quad recommendations} \tag{14}$$

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.

5.1.2 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 focuses 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.

To compute these metrics, we consider the confusion matrix in Table 1. Given this matrix it is possible to compute Precision and Recall as follows:

$$precision = \frac{a}{a+b} \tag{15}$$

$$recall = \frac{a}{a+c} \tag{16}$$

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.

5.1.3 F-measures and G-measure

F-measures [6] try to gather into a single value both precision and recall metrics. It is important to evaluate precision and recall in conjunction, because it is easy to optimize one of the metrics by declining the other. As an example, a recommendation system may recommend a large number of users, thus obtaining an high coverage. However, this negatively impact on the precision.

These measures are formulated as follows:

$$F_\beta = \frac{precision \cdot recall}{(1-\beta) \cdot precision + \beta \cdot recall} \tag{17}$$

The parameter β enables us to weight in different manners precision and recall. Hence, by varying the value of β, we obtain different measures. In our evaluation, at first we considered precision and recall to be equally important, by observing F_1, resulting in the harmonic mean of precision and recall:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{18}$$

Table 1 Confusion matrix

	Successful	Non-Successful
Retrieved	a	b
Non-Retrieved	c	d

Furthermore, we also evaluated the achieved performance for $\beta = 0.5$, in order to observe the $F_{0.5}$ measure, putting more emphasis on precision than recall.

Other than F-measure we also consider the G-measure which is given by:

$$G = \sqrt{precision \cdot recall} \tag{19}$$

Note that on the one hand, the F-measure references the True Positives to the Arithmetic Mean of Predicted Positives and Real Positives, being a constructed rate normalized to an idealized value. On the other hand, the Geometric Mean of Recall and Precision (G-measure) effectively normalizes True Positives to the Geometric Mean of Predicted Positives and Real Positives.

5.2 Experimental setup

Our basic goal in setting up our proof-of-concept experimental setup is testing our scheme on ground-truth publicly available data and make our experiments reproducible. Therefore, we evaluated it on data from Twitter that can be found on `snap.stanford.edu` [18]. This data set 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 data set 10 times took 15 minutes. Thus, a single execution of the basic algorithm takes 1 and a half minutes.

5.2.1 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 scheme 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 scheme, holding out each time a random subset of friends.

5.2.2 Impacting factors

The testing scenario has many variables that can influence the performance results of our proposed scheme, 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 5, 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 recommendation system, and then suggesting only the top ranked x % users, with x in $\{1\ \%, 10\ \%, 25\ \%, 50\ \%, 75\ \%, 100\ \%\}$.

The second factor, is also interesting for examination/testing. Indeed, the performances of our system may vary by changing α and β coefficients. Recall 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 Section 4.3 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, $a, b = 1$).

5.3 Results analysis

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 data set. In particular, we first analyze the distribution of the number of friends.

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

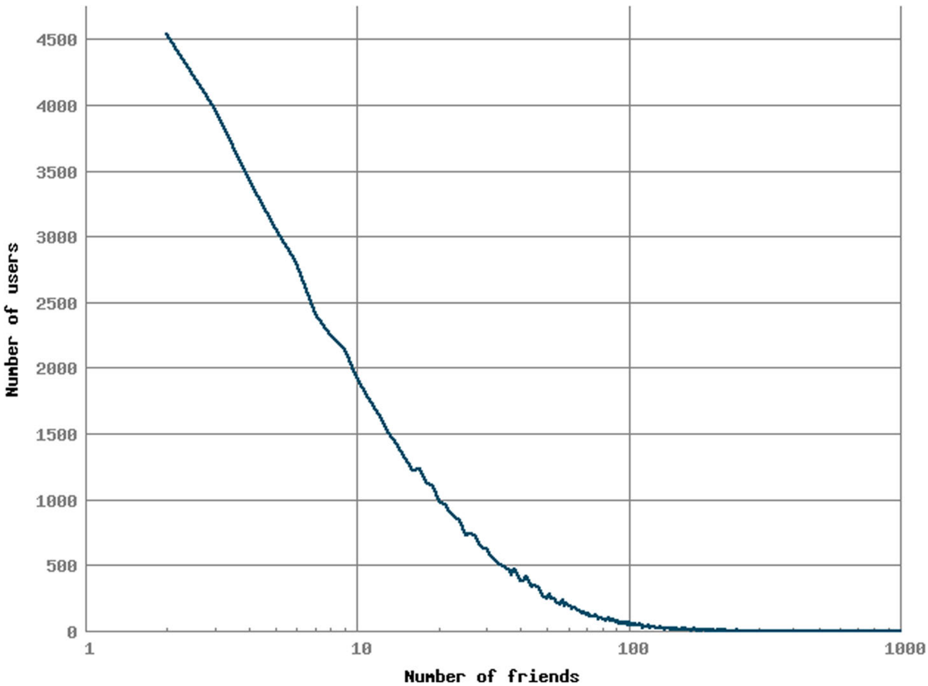


Figure 2 Distribution of the number of friends

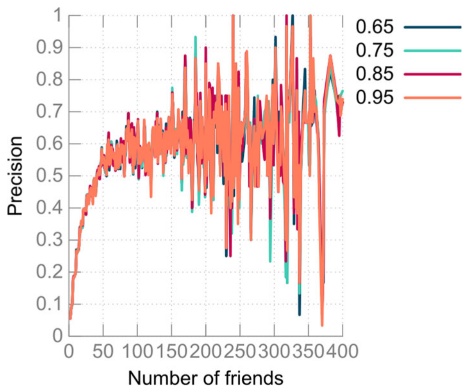


Figure 3 1 % precision – normal (*not boosted*) similarity, before Bezier interpolation

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

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.

5.3.1 Accuracy, precision and recall

Our first result concerns the first two impacting factors presented in Section 5.2.2: 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 5.3 % of the possible successful users. It is important to observe that the recall is more noisy than the precision. Recall from

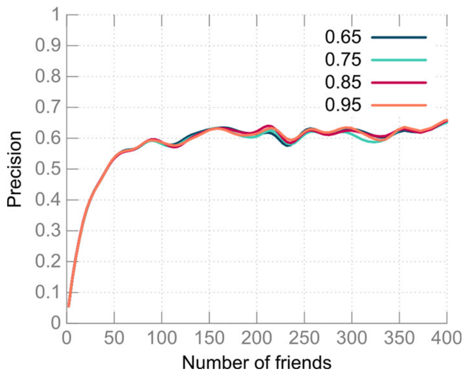
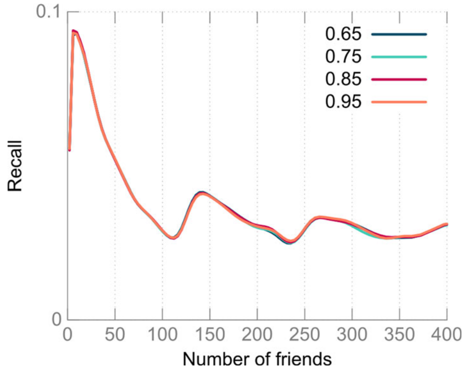


Figure 4 1 % precision – normal similarity

Figure 5 1 % recall – normal similarity



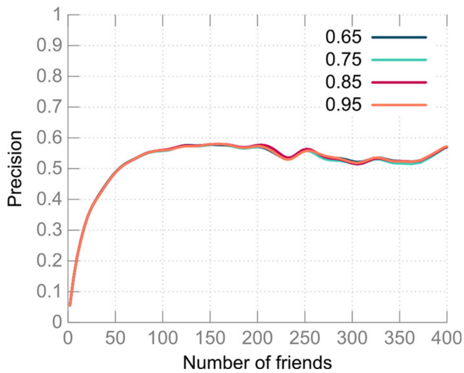
Section 5 that accuracy and precision values correspond. Bearing this into our minds, we can also see that relevant recommendations tend to be clustered towards the top of recommendation lists since the high accuracy value.

Let us stress that in our scenario, it is more important providing absolutely correct suggestions to some users (i.e., having a high precision) than providing all good suggestions (i.e., having a high recall). In other words, our recommendation system 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.

Figures 6 and 7 show precision and recall when to the user is presented a list of possible friendships which is long at least as the 10 % of all possible suggestions. In this case, the precision is still high and just slightly less than the previous analyzed case. Moreover, the recall value doubled. This is a good trade off since the ratio of excluded good suggestions decreased, while the top 10 % mostly comprises successful suggestions.

In the 100 % case, our system is suggesting to the target user all the friends of his friends, thus it represents an upper bound on the recall performance of any algorithm based on the idea of triadic closure. In other words, any algorithm that suggests to a target user a subset of the friends of his/her friends can not have a higher recall score of an algorithm that suggests all the friends of his friends. For this reason, the recall score of the 100 % reaches

Figure 6 10 % precision – normal similarity



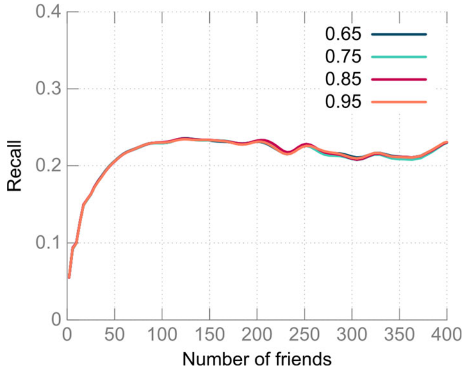


Figure 7 10 % recall – normal similarity

the highest possible value. As shown in Figure 8, it mainly ranges between 0.6 and 0.7. Figure 9 shows the precision, which is near 0.18 for most of the users. It is worth to consider that the fact that the precision declines when the recommendation-list size increases confirm our hypothesis that successful recommendations are showed in the highest part of the ranking.

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

5.3.2 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, like the previously analyzed case, for the boosted similarity score there are no substantial differences between the Tversky coefficients. The trends of precision and recall are shown in Figures 10 and 11. They are close to the previously presented ones. Therefore, we focus in the following on the comparison between the performances in case of boosted and normal similarity.

Since there are no significant differences between coefficients, we avoid to clutter charts by considering as representative trend only the curve given by $\alpha = 0.85$. Figure 12 shows

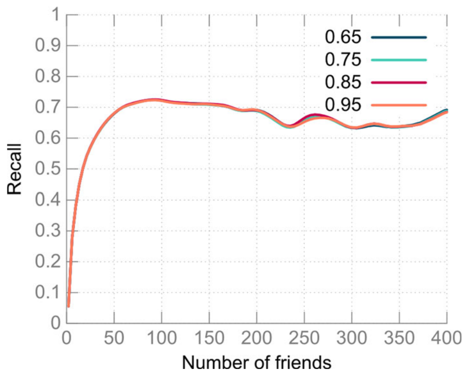
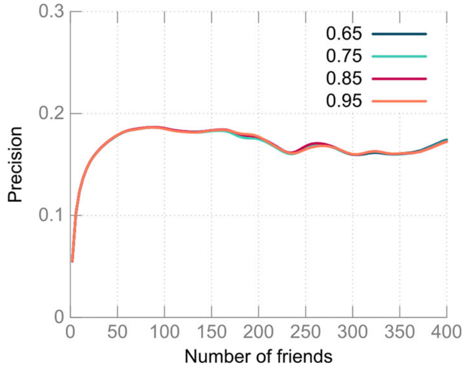


Figure 8 100 % recall – normal similarity

Figure 9 100 % precision – normal similarity



that not boosting similarity score gives higher precision when returning the top 1 % of suggestions. On the other hand, recall value does not change as shown in Figure 13.

However, this behavior does not hold for all the other cases in which more than 10 % suggestions are taken into account. This evidence can be seen, for example, in Figures 14 and 15. This highlights an interesting fact: when only few suggestions are considered, avoiding to boost similarity performs better. This is because a high intrinsic value is assigned, in the case of recommending few suggestions, to the popularity score computed via Local H&A. Recall that the Local Hubs And Authority let emerge the most popular (trustable) users inside the network of friends of the person who receives the suggestions. Consider, for example, a network in which the user u who receives suggestions has two friends u_1 and u_2. Suppose that one of them, say u_1 has a link to the other and not vice versa. Even if the u_2 is more similar to u than u_1, the fact that u_2 has a link to u_1 suggests that it is probable that also u_1 is similar to u. Thus, u_1's friends may be more interesting than the u_2's links from u's point of view.

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.

Figure 10 1 % precision – boosted similarity

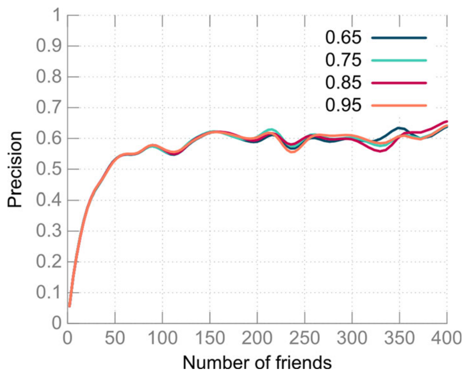
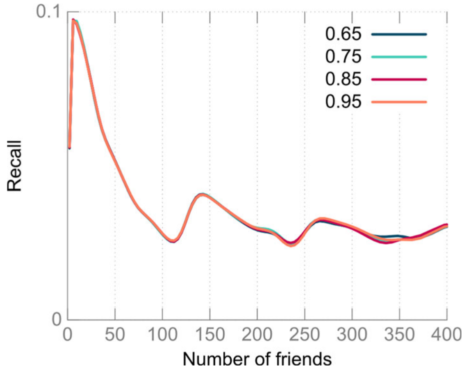


Figure 11 1 % recall – boosted similarity



5.3.3 *F-measures and G-measure*

We used the F_1 measure to characterize the system's performance as a function of the number of users grouped by category. In particular, we considered four categories:

- **few friends:** includes all the users which have a number of following users ranging from 2 to 30;
- **medium friends:** includes all the users which have a number of following users ranging from 30 to 100;
- **many friends:** includes all the users which have a number of following users ranging from 100 to 200;
- **full many friends:** includes all the users which have a number of following users ranging from 200 to 400;

F_1 scores presented in Table 2 confirm our results presented in the previous subsection: when presenting only the top 1 % of suggestions, the normal version performs slightly better than the boosted one. Indeed, except for the case of few friends, the F_1 score of the normal version has an increased value of about 0.01. In all the other cases, the boosted version outperforms the normal one of about 0.01. However, we can see from Figure 16, that there is no significant improvement between normal and boosted similarity when considering the top 75 % of suggestions.

Figure 12 1 % precision – boosted similarity

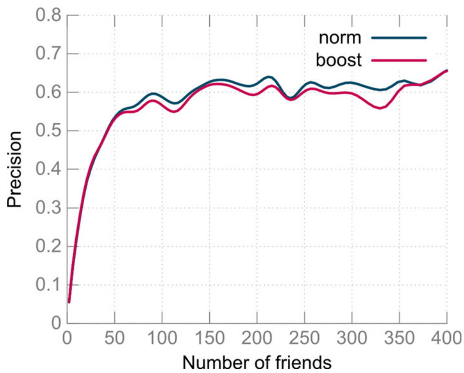


Figure 13 1 % recall – boosted similarity

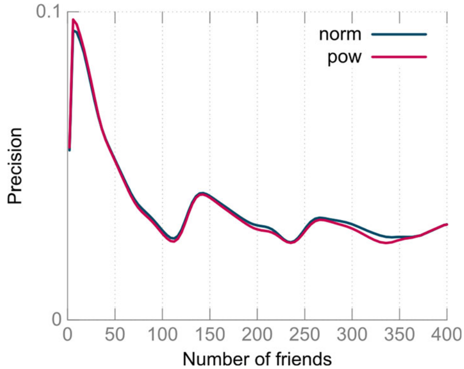


Figure 14 10 % precision – boosted similarity

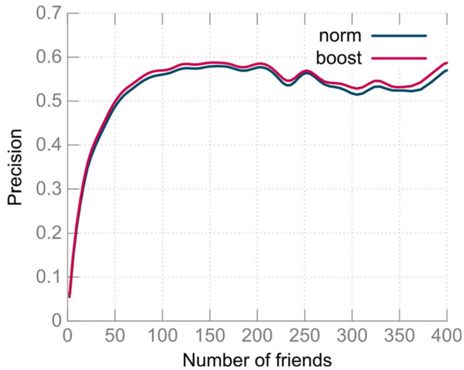


Figure 15 10 % recall – boosted similarity

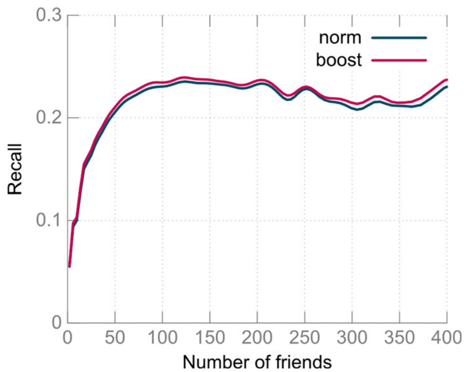


Table 2 Normal versus boosted similarity

2–30 friends

	Normal					Boosted				
Top	Precision	Recall	F_1	F_{05}	G	Precision	Recall	F_1	F_{05}	G
1 %	0.2923	0.0844	0.1309	0.1958	0.1570	0.2978	0.0862	0.1337	0.1998	0.1602
10 %	0.2765	0.1336	0.1801	0.2278	0.1922	0.2852	0.1379	0.1859	0.2350	0.1983
25 %	0.2386	0.2253	0.2317	0.2358	0.2318	0.2490	0.2351	0.2418	0.2461	0.2419
50 %	0.1894	0.3320	0.2412	0.2072	0.2508	0.1985	0.3480	0.2528	0.2172	0.2628
75 %	0.1552	0.4061	0.2246	0.1771	0.2511	0.1629	0.4263	0.2357	0.1859	0.2635
100 %	0.1347	0.4502	0.2074	0.1567	0.2463	0.1415	0.4728	0.2178	0.1646	0.2587

30–100 friends

	Normal					Boosted				
	Precision	Recall	F_1	F_{05}	G	Precision	Recall	F_1	F_{05}	G
1 %	0.5523	0.0441	0.0816	0.1670	0.1560	0.5411	0.0433	0.0802	0.1641	0.1531
10 %	0.5129	0.2142	0.3022	0.4011	0.3315	0.5242	0.2189	0.3089	0.4099	0.3388
25 %	0.3972	0.3903	0.3938	0.3958	0.3938	0.4086	0.4015	0.4050	0.4071	0.4050
50 %	0.2836	0.5481	0.3738	0.3139	0.3942	0.2915	0.5634	0.3842	0.3227	0.4053
75 %	0.2206	0.6392	0.3280	0.2538	0.3755	0.2263	0.6560	0.3366	0.2605	0.3853
100 %	0.1820	0.6972	0.2886	0.2135	0.3562	0.1864	0.7144	0.2957	0.2188	0.3649

100–200 friends

	Normal					Boosted				
	Precision	Recall	F_1	F_{05}	G	Precision	Recall	F_1	F_{05}	G
1 %	0.6085	0.0343	0.0650	0.1400	0.1445	0.5945	0.0336	0.0636	0.1369	0.1413
10 %	0.5733	0.2321	0.3305	0.4431	0.3648	0.5817	0.2356	0.3353	0.4496	0.3702
25 %	0.4287	0.4212	0.4250	0.4272	0.4250	0.4380	0.4304	0.4341	0.4364	0.4341
50 %	0.2950	0.5740	0.3897	0.3268	0.4115	0.3013	0.5864	0.3981	0.3338	0.4203
75 %	0.2244	0.6550	0.3343	0.2584	0.3834	0.2291	0.6691	0.3414	0.2638	0.3915
100 %	0.1822	0.7069	0.2897	0.2139	0.3588	0.1858	0.7214	0.2955	0.2182	0.3661

200–400 friends

	Normal					Boosted				
	Precision	Recall	F_1	F_{05}	G	Precision	Recall	F_1	F_{05}	G
1 %	0.6167	0.0291	0.0556	0.1224	0.1340	0.5972	0.0282	0.0538	0.1185	0.1297
10 %	0.5404	0.2187	0.3114	0.4175	0.3437	0.5507	0.2228	0.3173	0.4255	0.3503
25 %	0.3762	0.3747	0.3755	0.3759	0.3755	0.3855	0.3839	0.3847	0.3852	0.3847
50 %	0.2576	0.5111	0.3425	0.2859	0.3628	0.2648	0.5255	0.3522	0.2940	0.3730
75 %	0.1999	0.5950	0.2993	0.2306	0.3449	0.2053	0.6109	0.3073	0.2367	0.3541
100 %	0.1647	0.6526	0.2631	0.1937	0.3279	0.1691	0.6700	0.2701	0.1989	0.3366

An interesting result emerges from the observation of which is the category that benefits most of our proposal.

As shown in Figure 16, the category that reaches highest performances is the one with many users (i.e., [100; 200]). This is due to the nature of the proposed recommendation system scheme and the structure of the network. While there are enough samples in the

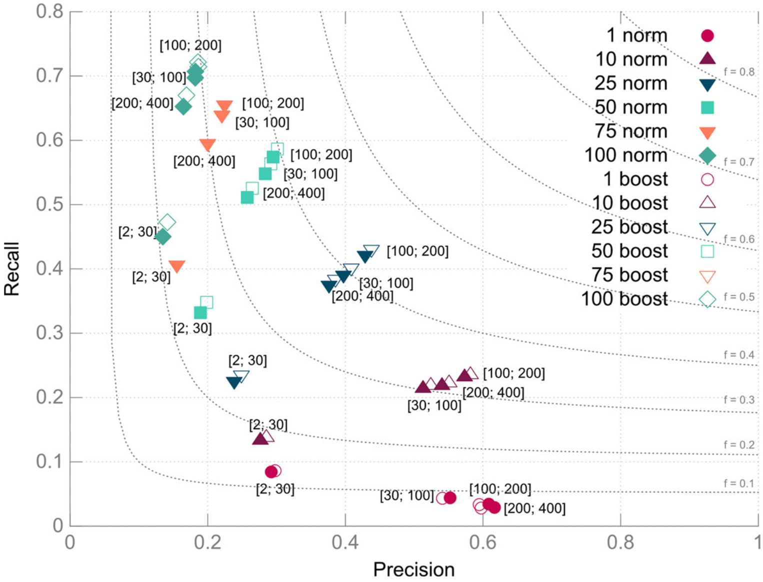


Figure 16 F_1 score normal versus boosted similarity

category, thus giving a good estimation of the trend, on the other hand, these users have a number of friends that is sufficient to ensure effective operations. Indeed, the system is able to successfully estimate the popularity score and to perform an accurate assessment of the similarity score. Analogous performances are shown by the category of medium users for the same reason. In the other situations, the same performances are not shown because they suffer from a lack in those two factors. Indeed, for the *few friends* category our scheme is not able to properly estimate the ratings giving the scarcity of friends on which runs. Furthermore, for the *full many* friends category, the performances are decreased of a small factor because of the sparsity of users' sampling. Indeed, in Twitter, becoming a follower of a user is a deliberative act and most users limit who they follow to avoid being swamped with too many messages.

So far, our evaluation considered the F-measure that equally weights precision and recall scores. However, as previously explained, for our recommendation system it is more important to have a high precision (i.e., to suggest user sets that are composed mostly by interesting users) than to have a high recall (i.e., to suggest all possible interesting users), therefore it could be argued that it would be valuable for the research community to agree on a different F-measure that weights more precision than recall.

Hence, we also tested how the algorithm performs with F_{05}.

Table 2 also shows results for the F_{05}-measure. Two main consideration can be done. The first one is that, as in the previous case, the boosted similarity performs better in all cases than the normal one. The second observation regards the comparison between F_1-measure and F_{05}-measure. In particular, when considering top 1 %, 10 % and 25 %, the F_{05} value is always greater than the F_1 one. However, no significant improvements are shown.

Finally, results for the G-measure are also presented in Table 2. We can notice that the obtained values, when applying this metric, are always greater than those obtained when considering F_1-measure. But, in this case, as well as for F_{05}-measure, results are almost stable.

6 Conclusion and future work

We presented and experimentally evaluated a recommendation algorithm based on Hubs And Authorities that exploits similarities to compute friendship suggestions. A dataset with more than 80,000 users was analyzed to test our proposal. The obtained 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 scheme may be improved by embedding other similarity measures, including analysis of hashtags, conversational likelihood, retweets, tweet volume and location.

As future work, we plan to evaluate our scheme on different similarity boosting factors. This approach may be able to highlight the best way to combine both worlds (triadic closures and similarity) and let the proposed algorithm perform even better. Another interesting comparison may be to check how the score varies by setting the α coefficient to assume a value smaller than 0.50. We also plan to implement and evaluate the performances of the algorithm exploiting Global Hubs And Authorities and to validate our proposal on different networks. Finally, we will evaluate strategies to work around structural holes in 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 Social Network studies to be very important for the gathering of relevant information among socially heterogeneous communities.

We also plan to assess, motivated by [27], the performance of the presented architecture in a real Cloud scenario. This is because, the computation power needed, especially if enriched with the factors presented above, may not be easily affordable, due to enormous amount of data available and the number of entities composing the network. Let us stress, that using Cloud assumes a key importance especially with the explosion of mobile applications and the support of Cloud Computing for a lot of services offered to mobile users. As a reference, Mobile Cloud Computing (MCC) has been introduced as a solution in mobile environments [10] in which the processing of tons and tons of data is required. Moreover, drove by [13, 31, 32] and [9] we plan to enrich our proposal also with the impacting factors they presented. However the effort needed for exploiting MCC is non-negligible.

References

1. Adamic, L., Buyukkokten, O., Adar, E.: A social network caught in the web. First Monday **8**(6) (2003)
2. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. CoRR cond-mat/0106096 (2001)
3. Armentano, M., Godoy, D., Amandi, A.: A topology-based approach for followees recommendation in Twitter. In: The 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, vol. 756, pp. 22–29 (2011)
4. Bak, P.: How Nature Works: The Science of Self-organized Criticality, 1st edn. Copernicus. Springer (1996). http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0387947914
5. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509 (1999). http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9910332
6. Billsus, D., Pazzani, M.: User modeling for adaptive news access. User Modelling and User-Adapted Interaction **10**(2-3), 147–180 (2000). Cited By (since 1996)204
7. Carullo, G., Castiglione, A., Cattaneo, G., De Santis, A., Fiore, U., Palmieri, F.: FeelTrust: Providing trustworthy communications in Ubiquitous Mobile environment. In: Proceedings of the International Conference on Advanced Information Networking and Applications, AINA, pp. 1113–1120 (2013)
8. Carullo, G., De Santis, A., Castiglione, A.: Friendship Recommendations in Online Social Networks. In: 2014 6th International Conference on Intelligent Networking and Collaborative Systems (INCoS), p. (to appear). doi:10.1109/INCoS.2014.32 (2014)

9. Chard, K., Caton, S., Rana, O., Bubendorfer, K.: Social Cloud: Cloud Computing in Social Networks. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 99–106. doi:10.1109/CLOUD.2010.28 (2010)
10. Dinh, H.T., Lee, C., Niyato, D., Wang, P.: A survey of mobile cloud computing: architecture, applications, and approaches. Wirel. Commun. Mob. Comput. **13**(18), 1587–1611 (2013). doi:10.1002/wcm.1203
11. Doney, P.M., Cannon, J.P.: An examination of the nature of trust in buyer-seller relationships. J. Mark., 35–51 (1997)
12. Dorogovtsev, S.N., Mendes, J.F.F.: Scaling properties of scale-free evolving networks: continuous approach. Phys. Rev. E **63**, 056,125 (2001). doi:10.1103/PhysRevE.63.056125
13. Eagle, N., Pentland, A.S., Lazer, D.: Inferring friendship network structure by using mobile phone data. Proc. Natl. Acad. Sci. **106**(36), 15,274–15,278 (2009). doi:10.1073/pnas.0900282106
14. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. SIGCOMM Comput. Commun. Rev. **29**, 251–262 (1999). doi:10.1145/316194.316229
15. Garcia, R., Amatriain, X.: Weighted Content Based Methods for Recommending Connections in Online Social Networks. In: Proceedings of the 2nd ACM RecSys'10 (2010)
16. Golder, S.A., Yardi, S., Marwick, A., Boyd, D.: A structural approach to contact recommendations in online social networks. In: Workshop on Search in Social Media, SSM (2009)
17. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. J. Mach. Learn. Res. **10**, 2935–2962 (2009)
18. Leskovec, J.: Stanford snapshots: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data/ (2013)
19. Kleinberg, J.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
20. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. In: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 668–677 (1998)
21. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web, WWW '10, pp. 591–600 (2010)
22. Laherrère, J., Sornette, D.: Stretched exponential distributions in Nature and Economy: "Fat tails" with characteristic scales. The European Physical Journal B - Condensed Matter and Complex Systems **2**(4), 525–539 (1998). doi:10.1007/s100510050276
23. Liang, Y., Li, Q.: Incorporating interest preference and social proximity into collaborative filtering for folk recommendation. In: SWSM 2011 (SIGIR workshop) (2011)
24. McAllister, D.J.: Affect-and cognition-based trust as foundations for interpersonal cooperation in organizations. Acad. Manag. J. **38**(1), 24–59 (1995)
25. Naruchitparames, J., Gunes, M., Louis, S.: Friend recommendations in social networks using genetic algorithms and network topology. In: 2011 IEEE Congress of Evolutionary Computation, CEC 2011, pp. 2207–2214 (2011)
26. Newman, M.E.: The structure and function of complex networks. SIAM rev. **45**(2), 167–256 (2003)
27. Noordhuis, P., Heijkoop, M., Lazovik, A.: Mining Twitter in the Cloud: A Case Study. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 107–114. doi:10.1109/CLOUD.2010.59 (2010)
28. Hernández del Olmo, F., Gaudioso, E.: Evaluation of recommender systems: A new approach. Expert Syst. Appl. **35**(3), 790–804 (2008)
29. Silva, N., Tsang, I.R., Cavalcanti, G., Tsang, I.J.: A graph-based friend recommendation system using genetic algorithm. In: 2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010 (2010)
30. Xie, J., Li, X.: Make best use of social networks via more valuable friend recommendations. In: 2012 2nd International Conference on Consumer Electronics, Communications and Networks, CECNet 2012 - Proceedings, pp. 1112–1115 (2012)
31. Yerva, S., Jeung, H., Aberer, K.: Cloud based social and sensor data fusion. In: 2012 15th International Conference on Information Fusion (FUSION), pp. 2494–2501 (2012)
32. Yu, Z., Zhou, X., Zhang, D., Schiele, G., Becker, C.: Understanding social relationship evolution by using real-world sensing data. World Wide Web **16**(5-6), 749–762 (2013). doi:10.1007/s11280-012-0189-x