Transitive Node Similarity for Link Prediction in Social Networks with Positive and Negative Links

Panagiotis Symeonidis
Department of Informatics
Aristotle University
Thessaloniki, 54124, Greece
symeon@csd.auth.gr

Eleftherios Tiakas
Department of Informatics
Aristotle University
Thessaloniki, 54124, Greece
tiakas@csd.auth.gr

Yannis Manolopoulos
Department of Informatics
Aristotle University
Thessaloniki, 54124, Greece
manolopo@csd.auth.gr

ABSTRACT

Online social networks (OSNs) like Facebook, and Myspace recommend new friends to registered users based on local features of the graph (i.e. based on the number of common friends that two users share). However, OSNs do not exploit the whole structure of the network. Instead, they consider only pathways of maximum length 2 between a user and his candidate friends. On the other hand, there are global approaches, which detect the overall path structure in a network, being computationally prohibitive for huge-size social networks. In this paper, we define a basic node similarity measure that captures effectively local graph features. We also exploit global graph features introducing transitive node similarity. Moreover, we derive variants of our method that apply in signed networks. We perform extensive experimental comparison of the proposed method against existing recommendation algorithms using synthetic and real data sets (Facebook, Hi5 and Epinions). Our experimental results show that our FriendTNS algorithm outperforms other approaches in terms of accuracy and it is also time efficient. We show that a significant accuracy improvement can be gained by using information about both positive and negative edges.

Categories and Subject Descriptors

G.2.2 [**Graph Theory**]: Graph algorithms

General Terms

Algorithms, Experimentation

1. INTRODUCTION

Online social networks (OSNs) such as Facebook.com[1], Myspace[2], Hi5.com[3], etc. contain gigabytes of data that can be mined to make predictions about who is a friend of whom. OSNs gather information on users' social contacts, construct a large interconnected social network, and recommend other people to users based on their common friends.

In this paper, we focus on recommendations based on links that connect the nodes of an OSN, known as the *Link Prediction* problem, where there are two main approaches that handle it. The first approach is based on local features of a network, focusing mainly on the nodes structure; the second approach is based on global features, detecting the overall path structure in a network.

Facebook.com and Hi5.com, as shown in Figure 1, have adopted a local method for recommending new friends to a target user v_8: "People you may know : (i) users v_2, v_3, v_5, v_6 because you have one common friend (user v_1) (ii) user v_4 because you have one common friend (user v_9) ...". The list of recommended friends is ranked based on the number of common friends each candidate friend has with the target user. However, in the aforementioned example, the list of recommended friends cannot be ranked, because the number of common friends is the same for all recommended friends. Thus, user v_8 gets as friend recommendation user v_2 or v_3 or v_4 or v_5 or v_6 with equal probability.

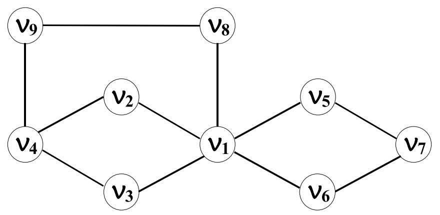


Figure 1: Example of Social Network.

Although Facebook.com and Hi5.com provide friend recommendations, they do not exploit effectively the similarities between the social graph nodes. Instead, they consider only pathways of maximum length 2 between a target user and his candidate friends. For example, according to existing OSNs, in Figure 1, user v_8 would get as friend recommendation with equal probability user v_2 or v_3 or v_4 or v_5 or v_6.

However, if we take into account the "strong" connection between v_8 and v_9 (due to the fact that v_9 does not share many edges with others) then v_4 should have a higher probability to be recommended as a friend to v_8. In contrast, other candidate friends (e.g. v_2, v_3, v_5, v_6) should have a

[1]http://www.facebook.com
[2]http://www.myspace.com
[3]http://www.hi5.com

lower probability to be recommended as friends to v_8 because of the "loose" connection between v_8 and v_1 (due to the fact that v_1 shares many edges with other nodes).

Compared to existing approaches, our method takes into account the local and the global features of a graph. In particular, we define a basic local similarity measure that captures effectively the proximity between neighbor graph nodes. We also exploit global graph features by introducing transitive node similarity. Thus, two persons that are connected with a path have a high probability to know each other, proportionally to: (i) the length of the path they are connected with, and (ii) the degree of similarity between the neighbor nodes that form that pathway.

Compared to the bulk of research on social networks that has focused almost exclusively on positive interpretations of links between people, we also study the interplay between positive and negative relationships. We connect our analysis to theories of signed networks, such as the Structural Balanced theory [8], and the Status theory [10, 11]. More details about these theories can be found later in Section 4.3.

The rest of this paper is organized as follows. Section 2 summarizes the related work, whereas Section 3 briefly reviews preliminaries in graphs employed in our approach. Section 4 defines a node similarity measure in OSNs. A motivating example, the proposed algorithm and an algorithm variation for signed networks are described in Section 4.2. Experimental results are given in Section 5. Finally, Section 6 concludes this paper.

2. RELATED WORK

The research area of link prediction in social networks, tries to infer which new interactions among members of a social network are likely to occur in the near future. There are two main approaches [12] that handle the *link prediction* problem. The first approach is based on local features of a network, focusing mainly on the nodes structure; the second approach is based on global features, detecting the overall path structure in a network.

There is a variety of local similarity measures [12] (i.e. Adamic/Adar index, Jaccard Coefficient, Common Neighbors index, etc.) for analyzing the "proximity" of nodes in a network. Among these indices, Adamic/Adar [1] index is reported [12] to attain the best performance in predicting new links in a social network. Adamic/Adar index, which is similar to Jaccard Coefficient (a commonly used similarity metric in information retrieval), measures how strongly "related" two web pages are. Common Neighbors index, also known as Friend of a Friend algorithm (FOAF) [2], is adopted by many popular OSNs, such as facebook.com and hi5.com for the friend recommendation task. FOAF is based on the common sense that two nodes v_x and v_y are more likely to form a link in the future, if they have many common neighbors. Finally, other local similarity measures are based on preferential attachment [12]. These similarity measures are based on the sum or product of nodes degree. The basic premise of preferential attachment is that the probability that a new edge involves a node is proportional to the current number of its neighbors.

There is a variety of global approaches [12] (i.e Shortest Path algorithm, RWR algorithm, SimRank algorithm etc.). Liben and Kleinberg [12] claimed that the identification of the shortest path between any pair of nodes in a graph can be used for link prediction (friend recommendation). The computation of the shortest path between two nodes, can be made using any well-known shortest path algorithm [4, 6]. RWR algorithm [15] (Random Walk with Restart algorithm) is based on a Markov-chain model of random walk through a graph. RWR considers a random walker that starts from node v_x who chooses randomly among the available edges every time, except that, before he makes a choice, with probability c he goes back to node v_x (restart). Thus, the relevance score of node v_x with respect to node v_y is defined as the steady-state probability r_{v_x,v_y} that the random walker will finally stay at node v_y. SimRank [9] also computes a global similarity measure based on the structural context of a network that says "two objects are similar if they are related to similar objects". Recently, Clauset et al. [3] proposed an algorithm based on the hierarchical network structure.

The novelty of our approach compared to existing approaches is as follows:

In contrast to global algorithms, such as the Random Walk with Restart (RWR) algorithm [15], the Shortest Path [4, 6] algorithm etc., our method also takes into account local graph features (i.e. the weighted similarity between nodes that may share many edges with others). We selected RWR (reported to present good accuracy results in [12]) and Shortest Path algorithms as representatives of the global algorithms and compared them with our method. As will be shown experimentally later, our method outperforms RWR and Shortest Path. The reason is, they traverse globally the social network, missing to capture adequately the local graph characteristics.

In contrast to local similarity measures, such as FOAF [2] algorithm (also known as the Common Neighbors index [13]), the Adamic/Adar [1] index etc., we take into account also global graph features (i.e. paths connecting any pair of nodes in an OSN). We have compared our method against FOAF algorithm and Adamic/Adar index, as representatives of the local-based measures. As will be shown experimentally later, our method outperforms FOAF and Adamic-Adar index. The reason is, we do not take into account only pathways of length 2 to compute similarity between a pair of nodes in an OSN. Instead, we use an extensive similarity measure that takes into account transitive node similarity.

Apart from the aforementioned link prediction algorithms, which are based solely on the graph structure, there are alternative methods [14, 7] that also exploit other data sources such as users messages, users ratings, co-authored papers, common tagging etc. However, we focus only on recommendations based on the link structure of an OSN, and thus, we will exclude them from our experimental comparison.

3. PRELIMINARIES IN GRAPHS

In this section, we present the most important notations and the corresponding definitions used throughout the rest of the paper.

Let $\mathcal{G}$ be a graph with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. Every edge is defined by a specific pair of graph nodes (v_i, v_j), where $v_i, v_j \in \mathcal{V}$. We assume that the graph $\mathcal{G}$ is undirected and un-weighted, thus the graph edges do not have any weights, plus the order of nodes in an edge is not important. Therefore, (v_i, v_j) and (v_j, v_i) denote the same edge on $\mathcal{G}$. We also assume that the graph $\mathcal{G}$ has no multiple edges, thus if two nodes v_i, v_j are connected with an edge of $\mathcal{E}$, then there is no other edge in $\mathcal{E}$ also connecting them.

Finally, we assume that there are no loop edges on $\mathcal{G}$ (i.e. a node can not be connected to itself). The graph expressing friendships among users of an OSN, which can be seen in Figure 1, will be used as our running example throughout the rest of the paper. For our calculations, we will use well-known representations, such as the adjacency matrix $A_{n\times n}$, and the incidence matrix $R_{m\times n}$.

4. BASIC NODE SIMILARITY MEASURE

In this section, we define a basic node similarity measure to determine the proximity between any pair of neighbor nodes in a graph $\mathcal{G}$. Therefore, if v_i and v_j are two neighbor connected nodes of $\mathcal{G}$, we define a specific function $sim(v_i, v_j)$ that expresses their corresponding similarity in the range [0,1] and has all the required properties (i.e. positivity, reflexivity, symmetry etc.) of a well-defined measure. The more similar the nodes are, the more the value of $sim(v_i, v_j)$ will be close to 1. On the contrary, the more dissimilar the nodes are, the more the value of $sim(v_i, v_j)$ will be close to 0.

To capture proximity between node vectors, we apply the Jaccard Coefficient, which is able to measure the degree of overlap between node vectors, in contrast to other measures (i.e. dot product, Euclidean distance etc.), which cannot measure it. In particular, we use an extension of the Jaccard Coefficient that contains the cosine similarity metric as we have binary vectors. This extension is also called the Tanimoto coefficient [16], and for two binary vectors $\mathbf{r_i}, \mathbf{r_j}$ is defined as:

$$sim(v_i, v_j) = \frac{\mathbf{r_i}\cdot\mathbf{r_j}}{||\mathbf{r_i}||^2 + ||\mathbf{r_j}||^2 - \mathbf{r_i}\cdot\mathbf{r_j}}$$

By substitution of vector operations between $\mathbf{r_i}, \mathbf{r_j}$ in the previous equation with the corresponding values of the incidence matrix R, we derive the following equivalent equation:

$$sim(v_i, v_j) = \\ = \frac{\sum_{h=1}^{m} R[e_h, v_i]\cdot R[e_h, v_j]}{\sum_{h=1}^{m} R[e_h, v_i]^2 + \sum_{h=1}^{m} R[e_h, v_j]^2 - \sum_{h=1}^{m} R[e_h, v_i]\cdot R[e_h, v_j]} \quad (1)$$

Note that the term $\sum_{h=1}^{m} R[e_h, v_i]\cdot R[e_h, v_j]$ in the final derived equation, expresses the number of edges that the nodes v_i, v_j share, whereas the terms $\sum_{h=1}^{m} R[e_h, v_i]^2$, $\sum_{h=1}^{m} R[e_h, v_j]^2$ are equal to the degrees of nodes v_i, v_j respectively.

The basic node similarity measure satisfies the positivity property, returning values into the interval [0,1]. Note that the maximum value of similarity (equal to 1) can be reached, when the two nodes are connected with only one edge and have no connections with other nodes. Moreover, Equation 1 can be simplified by using Theorem 1.

THEOREM 1. *If the basic node similarity measure of Equation 1 is applied in a graph $\mathcal{G}$ satisfying all mentioned assumptions of Section 3, then it is equivalent with the following Equation:*

$$sim(v_i, v_j) = \begin{cases} 0, & \textit{if } (v_i, v_j) \notin \mathcal{E} \wedge (v_j, v_i) \notin \mathcal{E} \\ \frac{1}{deg(v_i)+deg(v_j)-1}, & \textit{otherwise} \end{cases}$$

where $deg(v_i)$ and $deg(v_j)$ are the degrees of nodes v_i and v_j, respectively.

PROOF. The fact that $(v_i, v_j) \notin \mathcal{E}$ and $(v_j, v_i) \notin \mathcal{E}$ means that nodes v_i, v_j do not share any edges. Thus the term $\sum_{h=1}^{m} R[e_h, v_i]\cdot R[e_h, v_j]$ in Equation 1 is equal to 0 and $sim(v_i, v_j) = 0$.

If nodes v_i, v_j share one edge, then they can not share any other edge as explained in Section 3. Thus, the term $\sum_{h=1}^{m} R[e_h, v_i]\cdot R[e_h, v_j]$ in Equation 1 is equal to 1. Moreover, the terms $\sum_{h=1}^{m} R[e_h, v_i]^2, \sum_{h=1}^{m} R[e_h, v_j]^2$ are equal to the degrees of nodes v_i, v_j, respectively. In that case we have:

$$sim(v_i, v_j) = \frac{1}{deg(v_i) + deg(v_j) - 1},$$

and the theorem has been proved. □

Henceforth, Theorem 1 will be used in defining our basic similarity measure, which is based on the inverse sum of node degrees. However, someone could suggest the usage of any other local-based similarity measure [12] as described in Section 2. For this reason, our basic measure will be later experimentally compared with other measures, which are also based on the nodes degree and the preferential attachment process [12]: the sum of nodes degree and the product of nodes degree.

Now, let us calculate some similarity values on the graph of Figure 1 using Equation 1. The similarity between nodes v_1 and v_2 is: $sim(v_1, v_2) = \frac{1}{5+2-1} = \frac{1}{6} = 0.16$. The similarity between nodes v_2 and v_4 is: $sim(v_2, v_4) = \frac{1}{2+3-1} = \frac{1}{4} = 0.25$. Thus, the similarity score between nodes v_1, v_2 is less than that of v_2, v_4 because the degree of node v_1 is greater than that of v_4, whereas v_1 shares only one of its total 5 edges.

Collecting all similarity values between the nodes of a graph $\mathcal{G}$, we construct the *basic node similarity matrix* S of $\mathcal{G}$, which is an $n \times n$ matrix having n rows and n columns labeled by the graph nodes. The basic node similarity matrix values are defined as follows:

$$S[v_i, v_j] \ = \ sim(v_i, v_j)$$

In our running example, the basic node similarity matrix is depicted in Figure 2, where all values are rounded to the third decimal digit. As shown, user v_9 is more similar with user v_8 than user v_4. This is reasonable, because user v_4 is connected with 2 other nodes (v_2 and v_3), while user v_8 is connected with only 1 other node (v_1).

$$S = \begin{array}{ccccccccc|c} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & \\ 1 & 0.167 & 0.167 & 0 & 0.167 & 0.167 & 0 & 0.167 & 0 & v_1 \\ 0.167 & 1 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & v_2 \\ 0.167 & 0 & 1 & 0.25 & 0 & 0 & 0 & 0 & 0 & v_3 \\ 0 & 0.25 & 0.25 & 1 & 0 & 0 & 0 & 0 & 0.25 & v_4 \\ 0.167 & 0 & 0 & 0 & 1 & 0 & 0.333 & 0 & 0 & v_5 \\ 0.167 & 0 & 0 & 0 & 0 & 1 & 0.333 & 0 & 0 & v_6 \\ 0 & 0 & 0 & 0 & 0.333 & 0.333 & 1 & 0 & 0 & v_7 \\ 0.167 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.333 & v_8 \\ 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0.333 & 1 & v_9 \end{array}$$

Figure 2: Basic Node Similarity Matrix.

4.1 Transitive Node Similarity Measure

Based on Theorem 1, the similarity values between all non-neighbor nodes in a graph $\mathcal{G}$ are zero. For instance, in our running example, the similarity value between nodes v_1 and v_4 is zero, because they do not share any edge. However, users v_1 and v_4 have both user v_2 as a common friend, and thus they could be related in some way.

By using a transitive similarity we can efficiently solve this problem. In our method, we define a transitive node similarity, between two nodes v_i and v_j, denoted as extended similarity. Extended similarity is calculated by the product of the basic similarities between the nodes of the shortest path from v_i to v_j.

This shortest path expresses the minimum number of edges required to connect the two nodes, as all edges of graph $\mathcal{G}$ are not weighted. Therefore, we define the following extended node similarity measure for any two nodes of $\mathcal{G}$:

$$esim(v_i, v_j) = \begin{cases} 0, \text{if there is no path between } v_i, v_j \\ sim(v_i, v_j), \quad \text{if } v_i, v_j \text{ are neighbors} \\ \prod_{h=1}^{k} sim(v_{p_h}, v_{p_{h+1}}), \qquad \text{otherwise} \end{cases} \tag{2}$$

where $v_{p_1}=v_i$, $v_{p_{k+1}}=v_j$ and the nodes v_{p_h} (for h=2,…,k) are all the intermediate nodes that the shortest path from v_i to v_j passes through. Note that, in case that v_i, v_j are neighbor nodes, the shortest path between them is the single edge connecting them, and this explains why $esim(v_i, v_j) = sim(v_i, v_j)$.

In our running example, according to the previous definition, the extended similarity between nodes v_1 and v_4 using Equation 2 equals:

$$sim(v_1, v_4) = sim(v_1, v_2) \cdot sim(v_2, v_4) = \frac{1}{6} \cdot \frac{1}{4} = \frac{1}{24} = 0.042$$

as the shortest path between v_1, v_4 is: $v_1 \rightarrow v_2 \rightarrow v_4$ (the alternative path $v_1 \rightarrow v_3 \rightarrow v_4$ has the same length and the same similarity score since nodes v_3 and v_2 have equal degrees). Note that the extended similarity score between nodes v_1, v_4 is less than the basic similarity score of v_1, v_2 (0.167) and v_2, v_4 (0.25).

Collecting all the extended similarity values between the nodes of a graph $\mathcal{G}$, we construct the *extended node similarity matrix ES* of $\mathcal{G}$. It is a matrix which has the same dimensionality and structure with the basic node similarity matrix S. Its values are defined as follows:

$$ES[v_i, v_j] \;=\; esim(v_i, v_j)$$

In our running example, the extended node similarity matrix is depicted in Figure 3, where all values are rounded to the third decimal digit.

It is important to note that using the extended node similarity in a connected graph, such as the graph $\mathcal{G}$ of our running example, all values of ES will be positive numbers (non-zero values). This is due to the fact that there is always a shortest path between any pair of node of a connected graph.

4.2 The FriendTNS Algorithm

In this section, we present the proposed algorithm, denoted as *FriendTNS* (Friend Transitive Node Similarity), we analyze its steps, provide implementation details and discuss its time and space complexity.

$$ES = \begin{array}{ccccccccc|c} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & \\ 1 & 0.167 & 0.167 & 0.042 & 0.167 & 0.167 & 0.056 & 0.167 & 0.056 & v_1 \\ 0.167 & 1 & 0.028 & 0.25 & 0.028 & 0.028 & 0.009 & 0.028 & 0.062 & v_2 \\ 0.167 & 0.028 & 1 & 0.25 & 0.028 & 0.028 & 0.009 & 0.028 & 0.062 & v_3 \\ 0.042 & 0.25 & 0.25 & 1 & 0.007 & 0.007 & 0.002 & 0.083 & 0.25 & v_4 \\ 0.167 & 0.028 & 0.028 & 0.007 & 1 & 0.028 & 0.333 & 0.028 & 0.009 & v_5 \\ 0.167 & 0.028 & 0.028 & 0.007 & 0.028 & 1 & 0.333 & 0.028 & 0.009 & v_6 \\ 0.056 & 0.009 & 0.009 & 0.002 & 0.333 & 0.333 & 1 & 0.009 & 0.003 & v_7 \\ 0.167 & 0.028 & 0.028 & 0.083 & 0.028 & 0.028 & 0.009 & 1 & 0.333 & v_8 \\ 0.056 & 0.062 & 0.062 & 0.25 & 0.009 & 0.009 & 0.003 & 0.333 & 1 & v_9 \end{array}$$

Figure 3: Extended Node Similarity Matrix.

The basic task of the *FriendTNS* algorithm is simple: to compute the similarities from a specific node (user) v_0 to all other nodes (users) in a graph, using our basic and extended node similarity measures. The algorithm input is the graph $\mathcal{G}$, the node v_0, which represents the target user that will take friend recommendations and the number r of friends that will be recommended to him. The output is the recommendations array $recom[r]$.

Firstly, FriendTNS initializes the arrays and computes all node degrees from the graph data. Then, it computes all similarities for the target user v_0 in an array $s[n]$. It uses the formula of Theorem 1, if the examined node v_i is a neighbor of v_0. Otherwise, it uses Equation 2. Friends can be recommended to v_0 according to their weights in $s[n]$. Therefore, we sort the similarity list $s[n]$, we keep an index $ind[n]$ for the corresponding node ID's, and we recommend the top-r nodes (users), which are not already friends of v_0.

In our running example, user v_8 would receive user v_4 as friend recommendation, because his similarity score (0.083) is greater than the similarity score of users v_2, v_3, v_5, v_6 (0.028). Note that the similarity values of the neighbor nodes of v_8 (and v_8 itself) are ignored as these are already friends of the target user v_8. The resulting recommendation is reasonable, due to the fact that user v_9 (which is responsible for recommending user v_4 to target user v_8) does not share many edges with others. In contrast, user v_1 (which is responsible for recommending users v_2, v_3, v_5, v_6 to target user v_8) shares many edges with others. Thus, our *FriendTNS* algorithm is able to capture the associations among the graph nodes.

FriendTNS keeps the graph nodes and edges in memory using an adjacency list representation, which requires an $O(n+m)$ space, where n is the total number of nodes and m is the total number of edges. All other arrays $(s, ind, recom)$ require $O(n)$ space. Therefore, our FriendTNS total space complexity is $O(n + m)$.

For FriendTNS's similarity calculations, we use the one-to-all nodes shortest path algorithm of Fredman-Tarjan [6], which has a complexity of $O(m + n \log n)$, and when occurs an update into the shortest path tree of the graph, we immediately update the transitive similarity values on-the-fly. Therefore, the computational complexity of FriendTNS is $O(m + n \log n)$.

4.3 Extending FriendTNS for Signed Networks

In this Section, we derive variants of FriendTNS that apply to directed networks and networks with weighted edges, including the case of edges with negative weights (signed networks).

In signed networks edges have positive (+1) as well as negative (-1) weights. Such signed graphs arise for instance in social networks (i.e. Epinions.com, Shashdot Zoo, etc.) where negative edges denote enmity instead of friendship. In such signed graphs, FriendTNS's basic similarity measure of Theorem 1, which is the inverse of the sum of nodes' degree, can be adjusted accordingly based on the Status theory [10, 11].

Based on Status theory [10, 11], the positive nodes' in-degree $deg_{in}^{+}(x)$ and the negative nodes' in-degree $deg_{out}^{-}(x)$ of a node x increase its status. In contrast, the positive nodes' out-degree $deg_{out}^{+}(x)$, and the negative nodes out-degree $deg_{in}^{-}(x)$ decrease its status. In the following, our basic similarity measure is transformed, so that it can take into account the aforementioned properties of Status Theory: $sim(v_i, v_j) = \frac{1}{\sigma(v_i)+\sigma(v_j)-1}$, where $\sigma(x) = deg_{in}^{+}(x) + deg_{out}^{-}(x) - deg_{out}^{+}(x) - deg_{in}^{-}(x)$.

As already stated, in networks with negative edge weights the concept of transitivity has to take into account negative values. Thus, for our extended similarity measure of Theorem 2, if some edges have negative weight, the total weight of a shortest path can be calculated as the product of the edges's weights, based on the assumption of multiplicative transitivity of the structural balance theory [8, 11], as formulated in the graph-theoretic language by Hage and Harary (1983).

Structural balance theory considers the possible ways in which triangles on three individuals can be signed. Triangles with three positive signs exemplify the principle that "the friend of my friend is my friend", whereas those with one positive and two negative edges capture the notions "the enemy of my friend is my enemy", "the friend of my enemy is my enemy", and the "enemy of my enemy is my friend".

5. EXPERIMENTAL EVALUATION

In this section, we compare experimentally our approach with existing friend recommendation algorithms. Henceforth, our proposed approach is denoted as FriendTNS. We use in the comparison the Random Walk with Restart [15] algorithm, the Shortest Path[4] algorithm, the Adamic and Adar [1] algorithm and the Friend of a Friend [5] algorithm, denoted as RWR, Shortest Path, Adamic/Adar and FOAF, respectively. Our experiments were performed on a 3 GHz Pentium IV, with 2 GB of memory. All algorithms were implemented in C. To evaluate the examined algorithms, we have generated synthetic data sets and chosen three real data sets from the Facebook, Hi5, and Epinions web sites.

5.1 Real Data Sets

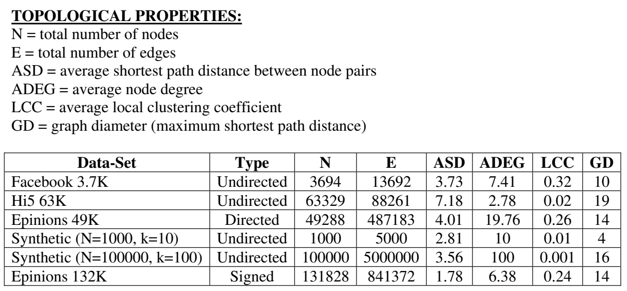
We used the Epinions[4] data set, which is a who-trusts-whom social network. In particular, users of Epinions.com express their Web of Trust, i.e. reviewers whose reviews and ratings they have found to be valuable. It contains 49K users and 487K edges among pair of users. Moreover, we crawled the Facebook website on the 30th of October, 2009. Our data crawling method was the following: For each user u, we traverse all his friends and then traverse the friends of each of u's friends etc. We created a data set with 3694 users, denoted as Facebook 3.7K. Moreover, from the Hi5 web site, we crawled 63329 users and all of their friends, denoted as Hi5 63K [5], available from the Hi5 site on the 15th of April, 2009. Finally, we also use in our comparison the extended Epinions 132K data set, which consist of positive and negative edges. A positive edge implies friendship/trust whereas a negative edge implies enmity/distrust.

[4]http://www.trustlet.org/wiki/Downloaded_Epinions_dataset

5.2 Generation of Synthetic Data Sets

In contrast to purely random (i.e., Erdos-Renyi) graphs, where the connections among nodes are completely independent random events, our synthetic model ensures dependency among the connections of nodes, by characterizing each node with a ten-dimensional vector with each element a randomly selected real number in the interval $[-1, 1]$. This vector represents the node's intrinsic features such as the profile of a person. Two nodes are considered to be similar and thus of high probability to connect to each other if they share many close attributes. Given a network size N and the degree k of each node, we start with an empty network with N nodes. At each time step, a node with the smallest degree is randomly selected (there is more than one node having the smallest degree). Among all other nodes whose degrees are smaller than k, this selected node will connect to the most similar node with probability $1-p$, while a randomly chosen one with probability p. This process will be terminated when all nodes are of degree k. The parameter $p \in [0, 1]$ represents the strength of randomness in generating links, which can be understood as noise or irrationality that exists in almost every real system. Based on the above procedure, we have created 2 synthetic data sets based on different network sizes N (1000, 100000), with k nodes degree equal to 10 for the first synthetic data set and with k equal to 100 for the second synthetic data set, whereas p is equal to 0.2 for both data sets.

We calculated several topological properties of the synthetic and real data sets which are presented in Figure 4.



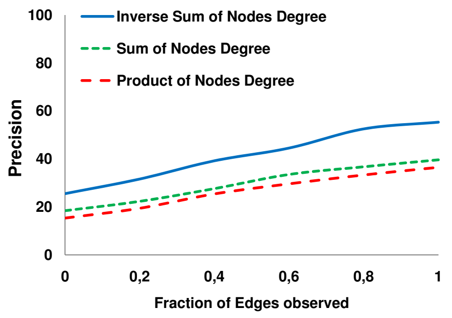
TOPOLOGICAL PROPERTIES:
N = total number of nodes
E = total number of edges
ASD = average shortest path distance between node pairs
ADEG = average node degree
LCC = average local clustering coefficient
GD = graph diameter (maximum shortest path distance)

Data-Set	Type	N	E	ASD	ADEG	LCC	GD
Facebook 3.7K	Undirected	3694	13692	3.73	7.41	0.32	10
Hi5 63K	Undirected	63329	88261	7.18	2.78	0.02	19
Epinions 49K	Directed	49288	487183	4.01	19.76	0.26	14
Synthetic (N=1000, k=10)	Undirected	1000	5000	2.81	10	0.01	4
Synthetic (N=100000, k=100)	Undirected	100000	5000000	3.56	100	0.001	16
Epinions 132K	Signed	131828	841372	1.78	6.38	0.24	14

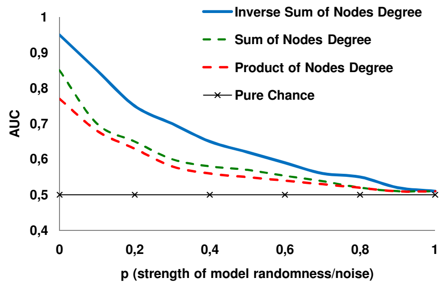
Figure 4: Topological properties of the real and synthetic data sets.

As shown in Figure 4, Epinions 49K and Facebook 3.7K present (i) a large clustering coefficient (LCC), and (ii) a small average shortest path length (ASD). These topological features can be mainly discovered in small-worlds networks. Small-world networks have sub-networks that are characterized by the presence of connections between almost any two nodes within them (i.e. high LLC). Moreover, most pairs of nodes are connected by at least one short path (i.e. small ASD).

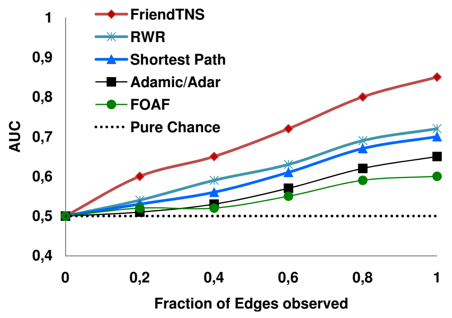
[5]Our Facebook and Hi5 data sets are available in our web site: `http://delab.csd.auth.gr/~symeon`



(a)



(b)



(c)

Figure 5: (a) Precision vs. different basic similarity measures for the synthetic 1K data set. (b) AUC vs. p strength of graph model randomness for the synthetic 1K data set. (c) Comparison of FriendTNS, RWR, Shortest Path, Adamic/Adar, FOAF and the pure chance algorithm for the Epinions 49K data set.

In contrast, as also shown in Figure 4, Hi5 63K has a very small LLC (0.02) and a quite big ASD (7.18). In other words, Hi5 data set can not be considered as a small-world network, since (i) most of its nodes can not be reached from every other by a small number of hops or steps and (ii) does not have sub-networks that are a few edges shy of being cliques.

5.3 Experimental Protocol and Evaluation

Our evaluation considers the division of friends of each target user into two sets: (i) the training set $\mathcal{E}^T$ is treated as known information and, (ii) the probe set $\mathcal{E}^P$ is used for testing and no information in the probe set is allowed to be used for prediction. It is obvious that, $\mathcal{E} = \mathcal{E}^T \cup \mathcal{E}^P$ and $\mathcal{E}^T \cap \mathcal{E}^P = \oslash$. Therefore, for a target user we generate the recommendations based only on the friends in $\mathcal{E}^T$.

Each experiment has been repeated 30 times (each time a different training set is selected at random) and the presented measurements, based on two-tailed t-test, are statistically significant at the 0.05 level. All algorithms predict the friends of the target users' in the probe set.

We use the classic precision/recall metric as performance measure for friend recommendations. For a test user receiving a list of k recommended friends (top-k list), precision and recall are defined as follows:

Precision is the ratio of the number of relevant users in the top-k list (i.e., those in the top-k list that belong in the probe set $\mathcal{E}^P$ of friends of the target user) to k.

Recall is the ratio of the number of relevant users in the top-k list to the total number of relevant users (all friends in the probe set $\mathcal{E}^P$ of the target user).

Moreover, we use the AUC statistic to quantify the accuracy of prediction algorithms and test how much better they are than pure chance, similarly to the experimental protocol followed by Clauset et al. [3]. AUC is equivalent to the area under the receiver-operating characteristic (ROC) curve. It is the probability that a randomly chosen missing link (a link in $\mathcal{E}^P$) is given a higher similarity value than a randomly chosen non-existent link (a link in $U - \mathcal{E}^T$, where U denotes the universal set). In the implementation, among n times of independent comparisons, if there are n' times the missing link having higher similarity value and n'' times the missing link and nonexistent link having the same similarity value, we define AUC, as follows: $AUC = (n' + 0.5 \times n'')/n$. If all similarity values are generated from an independent and identical distribution, the accuracy should be about 0.5. Therefore, the degree to which the accuracy exceeds 0.5 indicates how much better the algorithm performs than pure chance.

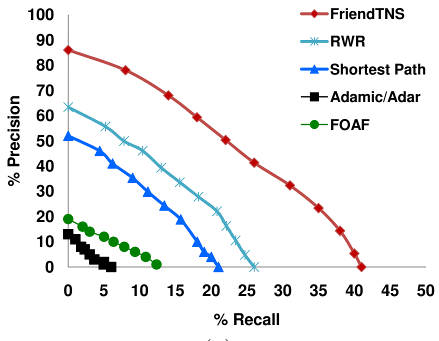
5.4 FriendTNS Sensitivity Analysis

In this Section, we study the sensitivity of FriendTNS in terms of accuracy performance. In particular, we test how the performance of FriendTNS is affected, when (i) it is combined with different basic similarity metrics, (ii) it runs on synthetic data sets with different controllable sparsity, and (iii) it runs with different graph model randomness.

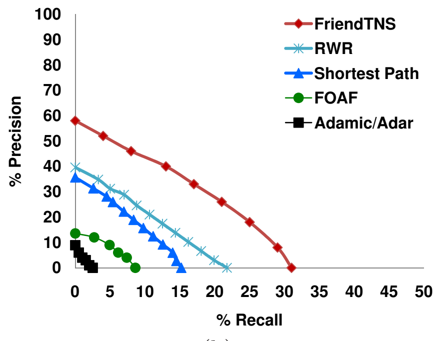
As the basic similarities of our proposed algorithms are calculated using the inverse sum of node degrees (Theorem 1), it is very interesting to compare the precision of our basic similarity measure with the corresponding precision of two other similarity measures, which are based on preferential attachment process [12]: the sum of node degrees and the product of node degrees. The basic premise of preferential attachment is that the probability that a new edge involves node v_i is proportional to current number of neighbors of v_i. Thus, we used the three aforementioned measures to calculate the extended similarities of FriendTNS algorithm for the synthetic 1K data set and then, we computed the precision attained by each measure vs. the fraction of observed links used in the training set (p parameter is fixed to 0.2). Figure 5(a) depicts the results.

We observe that the inverse sum of nodes degree measure outperforms both other measures. The same result holds for the 100K synthetic data set, but we do not present it due to lack of space. In the following, we adopt the inverse sum of nodes degree measure as the default basic similarity measure of FriendTNS. Notice also that, as we increase the fraction of observed edges, the precision of all algorithms is increased too. This is reasonable, since every prediction algorithm is expected to give higher accuracy for a denser network.

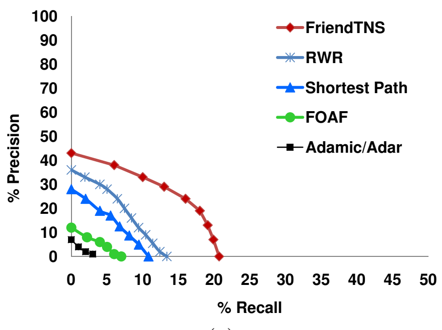
In our synthetic model, the parameter $p \in [0, 1]$ represents the strength of randomness in generating links. Next, we test FriendTNS's sensitivity with different graph model randomness. As shown in Figure 5(b), when the strength of randomness is weak, the inverse sum of nodes degree performs better than the other metrics. However, as the strength of randomeness becomes high all metrics cannot perform better than pure chance.



(a)



(b)



(c)

Figure 6: Comparison of FriendTNS, RWR, Shortest Path, Adamic/Adar and the FOAF algorithm for (a) Facebook 3.7K, (b) Epinions 49K and (c) Hi5 63K data sets.

5.5 Precision Comparison with other Methods

In this section we compare FriendTNS with other methods in terms of accuracy. The performance of all tested algorithms should be at least better than the case, where the friend recommendations would be performed by pure chance. Thus, to more meaningfully represent the friend recommendation algorithms' accuracy performance, we also use as baseline algorithm a pure chance predictor which simply randomly selects pairs of nodes in graph $\mathcal{G}$ to be friends. We use the AUC statistic, which looks at an algorithm's overall ability to rank all the missing connections over nonexisting ones. For the Epinions 49K data set, as shown in Figure 5(c), we plot a curve for AUC vs. the fraction of observed links used in the training set. As shown, FriendTNS does far better than pure chance, indicating that FriendTNS is a strong predictor of missing structure. The main reason is that FriendTNS captures effectively the local and global graph features. Notice that we have also verified the same results for the other real networks (Hi5 and Facebook), but we do not present them due to lack of space.

Next, we proceed with the comparison of FriendTNS with RWR, Shortest Path, Adamic-Adar, and FOAF algorithms, in terms of precision and recall. This reveals the robustness of each algorithm in attaining high recall with minimal losses in terms of precision. We examine the top-k ranked list, which is recommended to a target user, starting from the top friend. In this situation, the recall and precision vary as we proceed with the examination of the top-k list.

For the Facebook 3.7K data set, in Figure 6(a) we plot a precision vs. recall curve for all algorithms. As expected, all algorithms' precision falls as k increases. In contrast, as k increases, recall for all algorithms increases as well. FriendTNS attains the best results with impressive high precision. The reason is that FriendTNS exploits global and local features of the social graph by combining the basic with the extended similarity measure. In contrast, RWR traverses only globally the social network, missing to capture adequately the local characteristics of the graph. Moreover, Shortest Path does not take into account the reduced similarity between connected nodes that do not share many edges with others. Furthermore, Adamic/Adar and FOAF algorithms exploit only local features of the social network.

The precision of FriendTNS is impressive in this specific data set. The main reason is the topological characteristics of Facebook 3.7K data set. It presents (i) a large clustering coefficient (LCC) equal to 0.32, and (ii) a small average shortest path length (ASD) equal to 3.73. Thus, Facebook 3.7K can be considered as a small-world network .

For the Epinions 49K data set, as shown in Figure 6(b), we also plot precision vs. recall curve for all algorithms. FriendTNS again attains the best results. The precision of FriendTNS is again quite high. Based on its topological features, Epinions 49K can be also considered as a small-world network, since it presents high LLC and small ASD.

For the Hi5 63K data set, as shown in Figure 6(c), we plot precision vs. recall curve for all algorithms. However, the overall performance of FriendTNS, RWR and Shortest Path algorithms is significantly decreased compared with the results in both previous data sets. The main reason is the high sparsity (i.e. very small ADEG equal to 2.78) of the Hi5 data set. Moreover, it has very small LLC and quite big ASD (7.18). In other words, Hi5 data set can not be considered as a small-world network.

5.6 Time Comparison with other methods

In this section, we compare FriendTNS against the RWR, Shortest Path, Adamic/Adar, and FOAF algorithms in terms of time complexity. We have created 2 synthetic data sets based on different network sizes N (1000, 100000), where N is the total number of nodes in the network. For the first synthetic data set the k nodes degree is equal to 10, whereas k is equal to 100 for the second one. All recorded times are refer to the total required time for calculating similarities for a target node with all other nodes in a graph. Each algorithm's performance is obtained by averaging over 30 independent realizations. Figure 7 depicts the results. As shown, RWR present the worst results because it calculates the inverse of an $n \times n$ matrix. As expected, FOAF and Adamic/Adar algorithms, outperform the other algorithms due to their simpler complexity since they are local-based similarity measures. However, as already shown in Section 5.5, both methods perform the worst results in terms of accuracy.

5.7 FriendTNS Accuracy in Signed Networks

In this section, we present the accuracy performance of FriendTNS when we take into account positive and negative links of a signed network, i.e. extended Epinions 132K data set. We have two different variants of FriendTNS: The first variation considers only positive links and is denoted as $FriendTNS^{+}$. The second variation considers both pos-

Data-Set	FriendTNS	Shortest Path	Adamic/Adar	RWR	FOAF
Synthetic-(N=1000, k=10)	0.012sec	0.011sec	0.003sec	0.017sec	0.002sec
Synthetic-(N=100000, k=100)	1.102sec	1.05sec	0.394sec	1.621sec	0.280sec

Figure 7: Time complexity of the synthetic data sets. A smaller value is better.

itive and negative links and is denoted as $FriendTNS^{+}_{-}$. Figure 8 presents the precision and recall diagram for both versions of FriendTNS. As shown, $FriendTNS^{+}_{-}$ outperforms $FriendTNS^{+}$. The reason is that $FriendTNS^{+}_{-}$ exploits positive and negative links. This means that if we use information about negative edges for predicting the presence of positive edges we get an accuracy improvement of FriendTNS predictions. These results clearly demonstrate that there is, in some settings, a significant improvement to be gained by using information about negative edges, even to predict the presence or absence of positive edges.

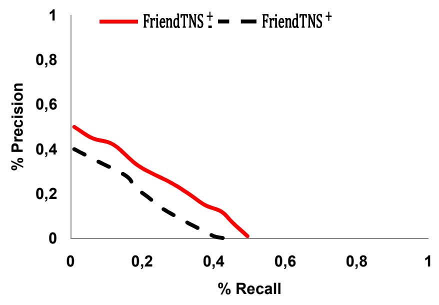


Figure 8: Accuracy performance of FriendTNS in terms of precision/recall.

6. CONCLUSIONS

In this paper, we proposed the FriendTNS algorithm to provide more accurate friend recommendations. We defined a transitive node similarity measure in OSNs by taking into account local and global features of a social graph. We also derived variants of our method that apply to signed networks. We performed an extensive experimental comparison using the three real social networks (Facebook, Epinions and Hi5 data sets). We have shown that our FriendTNS algorithm provides more accurate friend recommendations compared to existing approaches. In future, we want to improve friend recommendations based on other features that OSNs offer, such as photos and video tagging, and common applications. The combination of such features can provide valuable information and therefore yield to more accurate friend recommendations.

7. REFERENCES

[1] L. Adamic and E. Adar, "How to search a social network.", *Social Networks*, Vol.27, No.3, pp.187-203, 2005.

[2] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites.", *Proceedings of the 27th International Conference on Human Factors in Computing Systems (CHI'09)*, pp.201-210, 2009.

[3] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks.", *Nature*, pp.453, 2008.

[4] T. Cormen, C. Leiserson, R. Rivest, and S. Stein, "Introduction to Algorithms", *MIT Press*, 3rd edition, 2001.

[5] Facebook, Official blog, *http://blog.facebook.com/blog.php?post=15610312130.*

[6] M. Fredman and R. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms.", *Journal of the ACM*, Vol.34, pp.596-615, 1987.

[7] J. Golbeck, "Personalizing applications through integration of inferred trust values in semantic web-based social networks.", *Proceedings of Semantic Network Analysis Workshop at the 4th International Semantic Web Conference*, 2005.

[8] P. Hage and F. Harary, "Structural models in anthropology.", *Cambridge University Press*, Vol.56, 1984.

[9] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity.", *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'02)*, pp.538-543, 2002.

[10] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks.", *Proceedings of the 19th international conference on World Wide Web (WWW'10)*, pp.641-650, 2010.

[11] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media.", *Proceedings of the 28th international conference on Human factors in computing systems (CHI'10)*, pp.1361-1370, 2010.

[12] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks.", *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM'03)*, 2003.

[13] L. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins, "Geographic routing in social networks.", *Proceedings of National Academy of Sciences (PNAS'05)*, Vol.102, No.33, pp.11623-11628, 2005.

[14] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems.", *Proceedings of Federated International Conference On The Move to Meaningful Internet*, pp.492-508, 2004.

[15] J. Pan, H. Yang, C. Faloutsos, and P. Duygulu, "Automatic multimedia cross-modal correlation discovery.", *Proceedings 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pp.653-658, 2004.

[16] T. Tanimoto, *IBM Internal Technical Report*, 1957.