2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

Link Prediction in Social Networks using Computationally Efficient Topological Features

Michael Fire, Lena Tenenboim, Ofrit Lesser, Rami Puzis, Lior Rokach and Yuval Elovici
Deutsche Telekom Laboratories at Ben-Gurion University of the Negev,
Department of Information Systems Engineering, Ben Gurion University, Be'er Sheva, 84105, Israel
Email: mickyfi@bgu.ac.il, lenat@bgu.ac.il, lessero@post.bgu.ac.il, puzis@bgu.ac.il, liorrk@bgu.ac.il, and elovici@bgu.ac.il

***Abstract*—Online social networking sites have become increasingly popular over the last few years. As a result, new interdisciplinary research directions have emerged in which social network analysis methods are applied to networks containing hundreds millions of users. Unfortunately, links between individuals may be missing due to imperfect acquirement processes or because they are not yet reflected in the online network (i.e., friends in real-world did not form a virtual connection.) Existing link prediction techniques lack the scalability required for full application on a continuously growing social network which may be adding everyday users with thousands of connections. The primary bottleneck in link prediction techniques is extracting structural features required for classifying links. In this paper we propose a set of simple, easy-to-compute structural features that can be analyzed to identify missing links. We show that a machine learning classifier trained using the proposed simple structural features can successfully identify missing links even when applied to a hard problem of classifying links between individuals who have at least one common friend. A new *friends measure* that we developed is shown to be a good predictor for missing links and an evaluation experiment was performed on five large social networks datasets: Facebook, Flickr, YouTube, Academia and TheMarker. Our methods can provide social network site operators with the capability of helping users to find known, offline contacts and to discover new friends online. They may also be used for exposing hidden links in an online social network.**

***Index Terms*—Link Prediction, HiddenLinks, Social Networks, Supervised Learning**

I. Introduction and Background

In recent years, online social networks have been growing exponentially. Offering individuals with similar personal and business interests the possibility of meeting and networking, social networks are creating new opportunities to develop friendships, share ideas and conduct business. Online social networking services such as Facebook, Twitter and Flickr, just to name a few, have become part of the daily life of millions of people around the world. The enormous growth of these networks has resulted in several research directions that examine the structural and behavioral properties of large-scale social networks. Typically researchers in this area collect information by using WEB crawler software. Such crawlers, however, may sometimes collect only partial information. This is due to various causes such as the attempt to access broken Web links, efforts by the social network operator to block various subscribers for one reason or another, communication failures etc. Consequently, those depending upon the Web crawler might find themselves with only partial information about the set of links within a social network. In this such cases, heuristic techniques for uncovering hidden links missed by the WEB crawler are useful for completing the network structure.

Detection of hidden links is also very practical in friend-suggestion mechanisms used by online social networks. In such cases, the hidden links may consist of existing social ties that have not yet been established in the particular social network. Chen et al. [1] depict several algorithms used by IBM on their internal social network, which enable its employees to connect with each other. The problem of predicting the existence of hidden links or of creating new ones in social networks is commonly referred to as the *Link Prediction* problem. Link prediction has also many applications outside the domain of social networks. For example, in bioinformatics link prediction can be used to find interactions between proteins [2]; in e-commerce it can help build recommendation systems [3]; and in the security domain link prediction can assist in identifying hidden groups of terrorists or criminals [4]. Since the Link Prediction problem is relevant to different scenarios, several algorithms have been proposed in recent years to solve it. Most of the solutions are generally based on supervised machine learning and selecting relevant features, Bayesian probabilistic models, relational Bayesian networks, or linear algebraic methods. Further details on these approaches can be found in a thorough survey written by Mohammed J. Zaki and Mohammad Al Hasan [5].

One common approach for solving the Link Prediction problem is using supervised learning algorithms. This approach was introduced by Liben-Nowell and Kleinberg in 2003 [6], who studied the usefulness of graph topological features by testing them on bibliographic data sets. In 2006 Hasan et. al [4] extended their work, and since then several other researchers have implemented this approach. Most of the solutions that these researchers proposed were tested on bibliographic or on co-authorship data sets [4], [6], [7], [8]. In 2009, Song et al. used matrix factorization to estimate the similarity between nodes in real life social networks such as Facebook and MySpace [9]. Recently, in 2011, W J. Cukierski et. al. [10] extracted 94 distinct graph features. Using a Random Forests classifier, they achieved impressive results in predicting links on Flickr datasets.

In order to build an efficient classifier for link prediction, it is crucial to define and calculate a set of graph structural

978-0-7695-4578-3/11 $26.00 © 2011 IEEE
DOI

features. When dealing with large scale graphs that may include millions of nodes and edges, one of the challenges is the computationally intensive extraction of such features. For example, Facebook has nearly 700 million registered users and each month an average 20 million new users are added to the network[11]. In addition the topologies of social networks have several well-known characteristics such as power law degree distribution [12] and the small world phenomenon [13]. The power law degree distribution in social networks suggests that there are individuals with a large number of connections (hubs). Computing local topological features on a subgraph consisting only of friends of these individuals may be computationally intensive.

In this paper, we present a solution to the Link Prediction problem based on a machine learning classifier trained on a small set of easy-to-compute topological features. These classifiers are evaluated using several social network datasets such as Facebook[14], Flickr[15], Youtube[16], Academia[17] and TheMarker[18]. Further information on these social networks is presented in Section II. In addition, in Section III-A, we introduce a topological feature, *friends-measure*. This simple feature, a variation of the well-known Katz measure, estimates how well the friends of two users know each other.

In Section IV we show that this measure is more valuable for link prediction than the well-known *common-friends* feature and *Jaccard's-coefficient* [19].

The rest of the paper is organized as follows: In Section II we give a brief overview of the online social networks whose structures were used in this study. Section III describes the experimental framework and methods used to develop and evaluate a link predictor. The topological features used in this study are formally defined in Section III-A. Section IV presents our results including the area under the ROC curve (AUC) and classification accuracy, the contribution of each feature set, and the information gain value for the different features. Finally, in Section V we present our conclusions.

II. Online Social Networks Datasets

In this paper we evaluate link prediction classifiers in relation to five social network datasets: Facebook, Flickr, YouTube, Academia.edu, and TheMarker. Facebook is a social networking service and website launched in February 2004. As of June 2011, Facebook had nearly 700 million active users. Facebook users may create a personal profile, add other users as friends, and interact with other members. Since the friendship link between two members must be reciprocal, the existence of a link between member A and member B induces a mutual connection. We therefore refer to Facebook's underlying friendship graph as an undirected one. The Facebook data used in this research was obtained from [20].

Flickr is an image and video hosting website that enable its members to socially interact via comments and to follow each other by means of posted videos and images. Links between members do not require mutual approval since one may choose to follow any other visible member. Therefor the underlying graph that represents the Flickr social network is regarded as directed. We obtained a subgraph of Flickr users from [21].

YouTube is a popular video-sharing site that includes a social network. YouTube switched from directed links to a two-phase symmetric link-creation process in 2007. In this paper we use the dataset published by [22], which was collected while YouTube was still a directed graph.

The following two datasets where obtained for this research using a web-crawler. Academia.edu is a platform for academics to share and follow research underway in a particular field or discipline. Academics upload their papers to share them with other academics in over 100,000 research areas. An Academia social network member may choose to follow any of the other members in this network, hence the directed nature of the links within this network.

TheMarker Cafe is an Israeli online social network site that allows its member to connect and interact. Since most of its members are Israeli, most interaction and communication among members is done in Hebrew. Due to the geographic and demographic nature of this network, it is smaller in scale compared to the other networks. TheMarker friendship connection is reciprocal, hence its underlying social structure may be represented as an undirected graph.

In summary, Facebook and TheMarker are undirected networks while Flickr, YouTube and Academia are directed networks.

Details of the datasets are summarized in Table I.

TABLE I
SOCIAL NETWORKS DATASETS

Network	Is Directed	Node Number	Edge Number	Date
Academia	Yes	200,169	1,398,063	2011
Facebook	No	63,731	1,545,686	2009
Flickr	Yes	1,133,54	7,237,983	2010
TheMarker	No	65,953	1,572,684	2011
YouTube	Yes	1,138,499	4,945,382	2007

III. Methods and Experiments

Since our goal was to identify and predict a set of hidden links within a social network structure, we chose to use machine learning methods which allowed us to develop a link classifier which can predict the likelihood of a link in the social graph. We collected several social network datasets for this purpose. Some of the datasets (Academia and TheMarker) were collected using a dedicated Web crawling code that had been previously developed. The other datasets (from Flickr, YouTube and Facebook) were gathered from several online resources. Since we focused on predicting links based only on graph topology, we extracted a set of features from the corresponding graphs of the social networks. These attributes were then fed into WEKA [23], a popular suite of machine learning software written in Java and developed at the University of Waikato, New Zealand. In addition to these features sets, the WEKA software received the social network graphs as input and for each such network a set of links that are included in the graph (also referred to as positive links) as well as a set

of links that are not part of the social graph's original links set (referred to as negative links). With the goal of developing a preferred classifier, we then performed supervised learning, using various machine learning algorithms. The rest of this section describes the set of features that has been extracted from the social network graphs; the methods used to compose the training and test sets; and the machine learning algorithms that were examined.

A. Feature Extraction

This section describes all features that were extracted and used during our experiment. Let $G = \langle V, E \rangle$ be the graph that represents the topological structure of a general social network. Edges in the graph are denoted by $e = (u, v) \in E$ where $u, v \in V$. Our aim was to build a simple classifier, using machine learning techniques, so that for each two nodes (also referred to as vertices) $v, u \in V$ can predict whether or not the connection between u and v has a high probability of existing. Such classifier could then be used to decide whether $(u, v) \in E$ or $(u, v) \notin E$.

The features for this classifier were extracted from the topological structure of the graph. For each edge candidate for classification, we extracted a set of topological features. These features assist in estimating the chance that a given edge indeed exists in the graph. The edge features depend on the type of the graph. If the graph is directed, then we can extract more features based on the direction of the edges. Below we describe the topological features that were used to build the classifier.

*1) **Vertex Features**:* Let be $v \in V$, a neighborhood ($\Gamma(v)$) of v is defined as the set of v's friends, namely, vertices that are adjacent to v. In directed graphs (e.g., Academia, Flickr) the set of users that v follows (i.e., there is a directed link from v to these users) is different from the set of users that follow v (i.e., there is a directed link from them to v). We can therefore define outgoing ($\Gamma_{out}(v)$) and incoming ($\Gamma_{in}(v)$) neighborhoods respectively. A neighborhood of v can also include v or exclude it from the set of vertices. Inclusion and exclusion of v in the neighborhood generate sub-graphs that are very different with respect to their topological properties as shown below. Following are the formal definitions of neighborhoods that were used to extract topological features:

$$\begin{aligned}
&\Gamma(v) := \{u | (u, v) \in E \text{ or } (v, u) \in E\} \\
&\Gamma_{in}(v) := \{u | (u, v) \in E\} \\
&\Gamma_{out}(v) := \{u | (v, u) \in E\} \\
&\Gamma^{+}(v) := \Gamma(v) \cup \{v\}
\end{aligned} \tag{1}$$

Based on the definition of neighborhoods, we can also define subgraphs induced by these neighborhoods. We defined the *neighborhood-subgraphs* of v as:

$$\begin{aligned}
&nh\text{-}subgraph(v) = \{(x, y) \in E | x, y \in \Gamma(v)\} \\
&nh\text{-}subgraph^{+}(v) = \{(x, y) \in E | x, y \in \Gamma^{+}(v)\}
\end{aligned} \tag{2}$$

Using the above neighborhood definitions, we can create the following features for vertex v:

Vertex degree features: using the neighborhoods definition we defined the *degree* of v as:

$$d(v) = |\Gamma(v)| \tag{3}$$

For a directed graph G, we defined the *in-degree*, *out-degree*, and *bi-degree* features as follows: $d_{in}(v) = |\Gamma_{in}(v)|$, $d_{out}(v) = |\Gamma_{out}(v)|$ and $d_{bi}(v) = |\Gamma_{in}(v) \cap \Gamma_{out}(v)|$. Using the degree features, we defined *degree-density* features as:

$$\begin{aligned}
&in\text{-}degree\text{-}density(v) = \frac{d_{in}(v)}{d(v)} \\
&out\text{-}degree\text{-}density(v) = \frac{d_{out}(v)}{d(v)} \\
&bi\text{-}degree\text{-}density(v) = \frac{d_{bi}(v)}{d(v)}
\end{aligned} \tag{4}$$

In social networks, the degree feature represents the number of friends or followers that user v has.

Vertex subgraphs features: Using the *neighbor-subgraphs*, we defined the following features that denote the number of edges within the *neighbor-subgraphs* for each vertex v:

$$\begin{aligned}
&Subgraph\text{-}Edge\text{-}Number(v) = |nh\text{-}subgraph(v)| \\
&Subgraph\text{-}Edge\text{-}Number(v)^{+} = |nh\text{-}subgraph^{+}(v)|
\end{aligned} \tag{5}$$

We can also define the density of each subgraph namely:

$$\begin{aligned}
&Density\text{-}nh\text{-}subgraph(v) = \frac{d(v)}{|nh\text{-}subgraph(v)|} \\
&Density\text{-}nh\text{-}subgraph^{+}(v) = \frac{d(v)}{|nh\text{-}subgraph^{+}(v)|}
\end{aligned} \tag{6}$$

If the G is directed, we can also calculate both the number of strongly connected components (*SCC*) and the number of weakly connected components (*WCC*) [24] within the vertex subgraphs. The average number of vertices in these components can also be useful for classifying links as shown in Section IV:

$$\begin{aligned}
&avg\text{-}scc(v) = \frac{d(v)}{scc(nh\text{-}subgraph(v))} \\
&avg\text{-}wcc(v) = \frac{d(v)}{wcc(nh\text{-}subgraph(v))} \\
&avg\text{-}scc^{+}(v) = \frac{d(v)}{scc(nh\text{-}subgraph^{+}(v))}
\end{aligned} \tag{7}$$

The number of weakly / strongly connected components in v's subgraph may provide an indication of the number of different social groups v belongs to.

*2) **Edge Features**:* Let be $u, v \in V$ where $e = (u, v) \notin E$. Using the neighborhoods of u and v , one can extract several feature sets. These features include the number of common-friends u and v have (*common-friends(u,v)*); the number of distinct friends u and v have (*total-friends(u,v)*); the number of connections between u and v neighborhoods (*friends-measure(u,v)*); and many other features that we define below. These features help us determine the likelihood that a connection between u and v exists.

Common-Friends: The common-friends of $u, v \in V$ refers to the size of the common set of friends that both u and v possess. The formal common-friends definition for an undirected graph G is:

$$common\text{-}friends(u, v) = |\Gamma(v) \cap \Gamma(u)| \tag{8}$$

For a directed graph G, we can also define *common-friends* based on the edge direction: $common\text{-}friends_{in}(u, v) = |\Gamma_{in}(v) \cap \Gamma_{in}(u)|, common\text{-}friends_{out}(u, v) = |\Gamma_{out}(v) \cap \Gamma_{out}(u)|$, and $common\text{-}friends_{bi}(u, v) = |\Gamma_{bi}(v) \cap \Gamma_{bi}(u)|$. The relevance of the *common-friends* feature is very intuitive. It is expected that the larger the size of the common neighborhood, the higher the chances that both vertices will be connected. The *common-friends* feature was widely used in the past link prediction on several datasets, and found to be very helpful [3], [6], [8], [9], [10].

Total-Friends: For two vertices u, v, we can define the number of distinct friends that u and v have together, namely: Let be $u, v \in V$ we define the *total-friends* of u, v to be the number of distinct neighbors u, v has:

$$total\text{-}friends(u, v) = |\Gamma(u) \cup \Gamma(v)| \tag{9}$$

Jaccard's coefficient: *Jaccard's-coefficient* is a well-known feature for link prediction [3], [6], [8], [9], [10]. The Jaccard coefficient, which measures the similarity between sample sets, is defined as the size of the intersection divided by the size of the union of the sample sets. In our approach it indicates whether two social network members (vertices in the corresponding graph) have a significant amount of common-friends regardless of their *total-friends* set size. A higher value of *Jaccard's-coefficient* denotes stronger tie between two friends. The coefficient defines the ratio between the number of *common-friends* and the number of *total-friends*, namely:

$$jaccard's\text{-}coefficient(u, v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \tag{10}$$

Transitive Friends: If G is a directed graph, we can calculate the number of *transitive-friends* of u, v and v, u:

$$transitive\text{-}friends(u, v) = |\Gamma_{out}(u) \cap \Gamma_{in}(v)| \tag{11}$$

Preferential-attachment-score: One well-known concept in social networks is that users with many friends tend to create more connections in the future. This is due to the fact that in some social networks, like in finance, the *rich get richer*. We estimate how "rich" our two vertices are by calculating the multiplication between the number of friends or followers each vertex has, namely:

$$preferential\text{-}attachment\text{-}score(u, v) = |\Gamma(u)| \cdot |\Gamma(v)| \tag{12}$$

In several previous works on link prediction this attribute was found to be a very significant feature [4].

Katz-measure In 1953, L. Katz proposed a path-ensemble based proximity measure [25]. The *Katz-measure* is a variant of the *shortest-path measure* (see III-A4). The idea behind the *Katz-measure* is that the more paths there are between two vertices and the shorter these paths are, the stronger the connection. The *Katz-measure* is defined as:

$$katz(u, v) = \sum_{l=1}^{l_{max}=\infty} \beta^l |path^l_{u,v}| \tag{13}$$

where $|path^l_{u,v}|$ is the number of paths between u and v with length of l. The problem with the *Katz-measure* is that it has cubic complexity. This complexity makes it unfeasible for use in large social networks. Consequently, we did not use the *Katz-measure* feature when building our classifier. Instead we used the *Friends-measure* that can be regarded as an approximation of the *Katz-measure*.

Friends-measure When looking at two vertices in a social network, we can assume that the more connections their neighborhoods have with each other, the higher the chances the two vertices are connected. We take the logic of this statement and define the *Friends-measure* as the number of connections between u and v neighborhoods. The formal definitions of *Friends-measure* is: Let be $G = \langle V, E \rangle$ and $u, v \in V$.

$$friends - measure(u, v) = \sum_{x \in \Gamma(u)} \sum_{y \in \Gamma(v)} \delta(x, y) \tag{14}$$

where we define the function $\delta(x, y)$ as:

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \text{ or } (x, y) \in E \text{ or } (y, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

One can notice that in undirected networks, the *Friends-measure* is a private case of the *Katz-measure* where $\beta = 1$ and $l_{max} = 2$.

Opposite direction friends: For a directed graph G, we can create a specific measure that indicates if reciprocal connections exists between the nodes

$$opposite\text{-}direction\text{-}friends(u, v) = \begin{cases} 1 & \text{if } (v, u) \in E \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

*3) **Edge Subgraph Features**:* Let be $u, v \in V$ using the neighborhoods definitions from III-A1, we can define the following subgraphs:

$$\begin{aligned} nh\text{-}subgraph(u, v) &= \{(x, y) \in E | x, y \in \Gamma(u) \cup \Gamma(v)\} \\ nh\text{-}subgraph^{+}(u, v) &= \{(x, y) \in | x, y \in \Gamma^{+}(u) \cup \Gamma^{+}(v)\} \end{aligned} \tag{16}$$

These subgraphs contain data about the number of connections between the links of u, v including the inner connections

between each vertex neighborhood. These types of graphs were also used to extract features for link prediction by W J. Cukierski et. al.[10] Another subgraph from which we can create features is the *inner-connection subgraph*:

$$\begin{aligned} inner\text{-}subgraph(u,v) \quad = \quad & \{(x,y) \in E| \\ & (x \in \Gamma(u) \text{ and } y \in \Gamma(v)) \text{ or} \\ & (x \in \Gamma(v) \text{ and } y \in \Gamma(u))\} \end{aligned} \tag{17}$$

In social networks, the *inner-subgraph* represents the number of connections between the friends of each user.

Edge subgraphs edges number: Using the above definitions we can create features by counting the number of edges in each subgraph:

$$\begin{aligned} & |nh\text{-}subgraph(u,v)| \\ & |nh\text{-}subgraph^{+}(u,v)| \\ & |inner\text{-}subgraph(u,v)| \end{aligned} \tag{18}$$

Edge subgraphs components number: We can also count the number of strong and weak components for each subgraph:

$$\begin{aligned} & scc(nh\text{-}subgraph(u,v)), wcc(nh\text{-}subgraph(u,v)) \\ & scc(nh\text{-}subgraph^{+}(u,v)) \\ & scc(inner\text{-}subgraph(u,v)), wcc(inner\text{-}subgraph(u,v)) \end{aligned} \tag{19}$$

*4) **Path Features**:* Let be $u, v \in V$. We defined the following feature based on the length of the path between u and v:

Shortest Path: We defined the *Shortest-path(u,v)* feature to be the shortest path length between u and v in G. If G is a directed graph, we can also define the feature *shortest-path(v,u)*. The shortest path feature has been explored in several papers and found to be one of the most significant features in link prediction [4].

B. Experimental Setup

As part of our experiment we built a classifier for each social network graph G by randomly choosing 25,000 positive edges that exist in the graph and 25,000 negative edges that do not exist in the graph. We also randomly chose 25,000 negative edges, where the shortest path between each edge vertices was two. By using these randomly selected edges, we created two training sets: The first training set (i.e., the "easy" train set) was built by taking 25,000 positive edges and 25,000 negative edges. The second training set (i.e., the "hard" train set) was built by taking 25,000 positive edges and 25,000 negative edges in distance of two hops. These sets where formed in order to create datasets for training our classifiers.

Afterwards, in order to extract features, we developed a Python code using the Networkx graph package [26]. This code extracted the vector of features for each edge (u, v) in the training sets. For each vertices u, v, we extracted all the vertex features (see III-A1), and all the edge features for the edge (u, v) (see III-A2). In total, we extracted 53 features for directed graphs and 33 features for undirected graphs.

For each social network in our datasets, we created several feature subsets according to the different characteristics of the features. Specifically, we created the following feature subsets:

- **All-features subset:** contains all the extracted features: 53 features for directed networks and 33 for undirected networks.
- **Friends-features subset:** contains the following features: vertices *degree* features; *Common-friends*; *Total-friends*; *Preferential-attachment-score*; and *Friends-measure*. A total of 8 features for undirected networks, and 15 features for directed networks were created.
- **Common-friends subset:** contains only the *Common-friends* feature.
- **Friends-measure subset:** contains only the *Friends measure* feature.
- **Jaccard's coefficient:** contains only the *Jaccard's coefficient* feature.

In our experiment we evaluated a number of popular machine learning methods such as C4.5 decision trees, k-nearest-neighbors (kNN), naive-Bayes, support vector machines (SVM), and artificial neural networks (ANN).

Instead of only using a single technique, we also applied "committee machines", a well-known technique in machine learning (sometime associated with a more specific term such as ensemble learning or a mixture of experts) in which the outputs of several classifiers (experts) are combined. Each of the classifiers solves the same original task. Combining these classifiers usually results in a better composite global model, with more accurate and reliable estimates or decisions than can be obtained from using a single model. This idea imitates a common human characteristic – the desire to obtain several opinions before making any crucial decision.

It is known that combining different types of classifiers can improve predictive performance, mainly due to the phenomenon that various types of classifiers have different "inductive biases". In particular, it has been shown that combining diverse classifiers can be used to reduce the variance-error (i.e., error due to sampling variation) without increasing the bias-error (i.e., error due to an inadequate model). Additionally, many participants in prediction contests combine various models in order to achieve the best results (see, for example, Koren 2009[27]).

We used WEKA's [23] C4.5 (J48), IBk, NaiveBayes, SMO, MultilayerPerceptron, Bagging, AdaBoostM1, RotationForest and RandomForest implementations of the corresponding algorithms. For each of these algorithms most of the configurable parameters were set to their default values except for the following. For C4.5, the minimum number of instances per leaf parameter was set to 10; for IBk, its k parameter was set to 10; for SMO the NormalizedPolyKernel with its default parameters was used. The ensemble methods were configured as follows. The number of iterations for all ensemble methods was set to 100. The Bagging, AdaBoostM1 and RotationForest algorithms were evaluated using J48 as the base classifier

with the number of instances per leaf set to 10 and 250. The Bagging and RotationForest algorithms performed best with a minimal number of instances set to 10. The minimal number of instances for AdaBoostM1 was set to 250.

IV. Results

For each social network and subset of features we evaluated the list of machine-learning algorithms (see III-B) using a 10-fold cross-validation approach.

For example, AUC results on the Facebook network using a set of all features and a subset of friends-features are presented in Figures 1 and 2.

Fig. 1. AUC Results - Facebook using all features

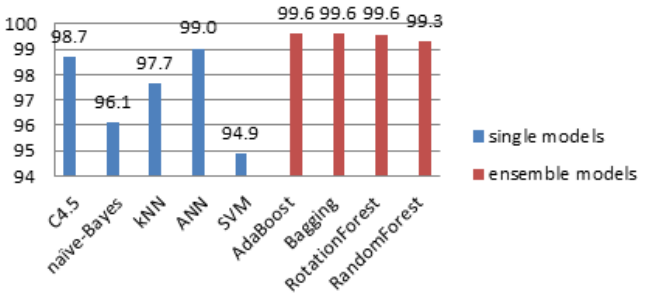
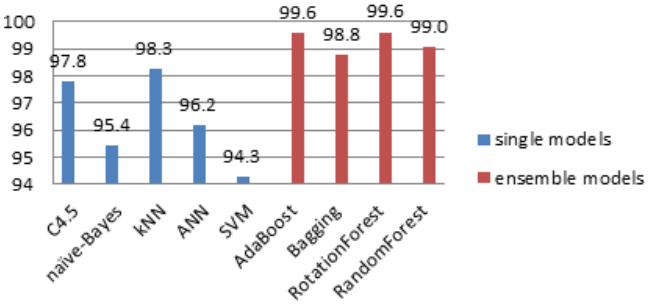


Fig. 2. AUC Results - Facebook using friends-features



It can be seen that the ensemble schemas achieved the highest predictive performance. However, their running times were very long. Among the single models, the C4.5 decision tree and artificial neural networks (ANNs) also achieved a relatively high performance. However, ANN models are difficult to interpret and have longer construction times than C4.5. Consequently, we found that C4.5 provides the best trade-off between computational costs and accuracy since its AUC results are, in most cases, slightly lower than those of the ensemble methods. On the other hand, its computational times are much faster. Another advantage of the C4.5 is that its models are easy to understand and analyze. Obviously, when the predictive performance is of the highest importance, more time- consuming and sophisticated methods can be applied. Tables II and III present the AUC and accuracy (i.e., percent. of correct predictions at threshold 0.5) results of the C4.5 method for each social network and feature subset.

TABLE II
Easy trainset - C4.5 AUC Results

Features	Academia	Facebook	Flickr	TheMarker	YouTube
All-features	0.992	0.987	0.999	0.971	0.999
Friends-features	0.985	0.977	0.988	0.967	0.997
Friends-Measure	0.931	0.922	0.952	0.915	0.915
Common-Friends	0.859	0.939	0.828	0.893	0.759
Jaccard's-coefficient	0.859	0.925	0.829	0.896	0.759

TABLE III
Easy trainset - C4.5 Accuracy Results

Features	Academia	Facebook	Flickr	TheMarker	YouTube
All-features	0.965	0.966	0.994	0.93	0.993
Friends-features	0.952	0.953	0.948	0.925	0.988
Friends-Measure	0.933	0.922	0.937	0.917	0.917
Common-Friends	0.861	0.94	0.831	0.896	0.76
Jaccard's-coefficient	0.861	0.928	0.832	0.886	0.76

We also created datasets in which the negative edge vertices were chosen randomly, but at a distance of two hops from each other. The AUC and accuracy results for these datasets using the C4.5 method are presented in Tables IV and V.

TABLE IV
Hard trainset - C4.5 AUC Results

Features	Academia	Facebook	Flickr	TheMarker	YouTube
All-features	0.976	0.898	0.967	0.928	0.998
Friends-features	0.946	0.893	0.955	0.92	0.991
Friends-Measure	0.788	0.762	0.902	0.838	0.884
Common-Friends	0.8	0.817	0.889	0.797	0.884
Jaccard's-coefficient	0.8	0.79	0.924	0.738	0.896

We used the Friedman test proposed by J. Demsar[28] for validating the statistical significance of differences between the evaluated sets of features. For this test we focused on the results of the C4.5 method. With a confidence level above 0.99, the results reveal that there are significant differences between the evaluated sets of features. Then, we proceeded with the Nemenyi post-hoc test [28] to compare the feature sets to each other. Interestingly, the following pairs of sets were found to be not significantly different: *all-features* (avg. rank 1) vs. *friends-features* (avg. rank 2), *common-friends* (avg. rank 4.25) vs.

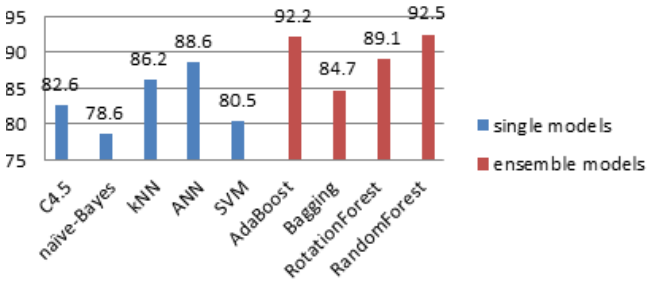
TABLE V
HARD TRAINSET - C4.5 ACCURACY RESULTS

Features	Academia	Facebook	Flickr	TheMarker	YouTube
All-features	0.925	0.828	0.921	0.86	0.989
Friends-features	0.881	0.825	0.885	0.86	0.963
Friends-Measure	0.785	0.757	0.781	0.806	0.87
Common-Friends	0.783	0.807	0.855	0.792	0.842
Jaccard's-coefficient	0.738	0.765	0.766	0.688	0.805

friends-measure (avg. rank 3.8), *common-friends* (avg. rank 4.25) vs. *Jaccard's-coefficient* (avg. rank 3.95), and *Jaccard's-coefficient* (avg. rank 3.95) vs. *friends-measure* (avg. rank 3.8).

We also tested our classifiers against the IJCNN Social Network challenge test set. The IJCNN test set was created using different random edge chooser algorithm. The IJCNN random algorithm was written in order to make link prediction more difficult and can be somewhat biased. Using the IJCNN random algorithm, we generated a training set of 25,000 positive edges and 25,000 negative edges and built a classifier using the features presented in section III-A. The AUC results of our classifier using the C4.5 algorithm when running on all features are presented on Figure 3. The AUC results achieved by C4.5 using only friends-features are presented in Figure 4.

Fig. 3. AUC Results - Flickr-kaggle competition using all features data

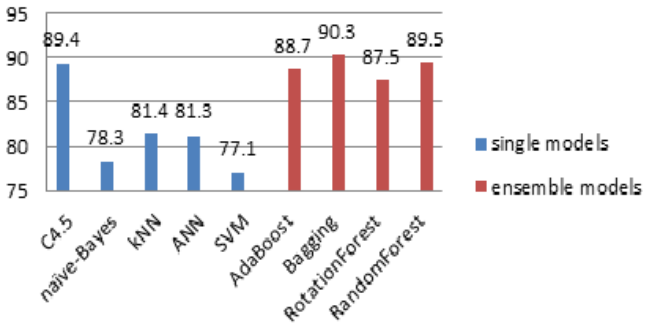


It can be seen that on this test data, the AdaBoost and Random Forest algorithms achieved the highest AUC results, 0.922 and 0.925 respectively. A variant of our classifier participated in the IJCNN Social Network challenge and achieved an AUC result of 0.9244 on the challenge test set. The average time for extracting features for each edge in the IJCNN test set was 0.64 seconds[1] instead 10 seconds per edge that was obtained by Cukierski et. al. [10].

To obtain an indication of the usefulness of various features, we analyzed their importance using Weka's information gain attribute selection algorithm. The top five attributes with the highest rank for each one of training sets for all social networks are presented in Tables VI and VII.

[1]We ran our algorithm using Python 2.7, on a regular Dell Latitude E6420 laptop with i7 core, and 8GB RAM

Fig. 4. AUC Results - Flickr-kaggle competition using friends-features data



It can be seen that different attributes are the most influencing for various social networks, however the Friends-measure is among the most influencing features on almost all of the networks.

TABLE VI
EASY TRAIN SET - INFOGAIN VALUE OF DIFFERENT FEATURES

Transitive-Friends(u,v)	Total-Friends(u,v)	Shortest-path(u,v)	scc(nh-subgraph+(u,v))	scc(nh-subgraph(u,v))	scc(Inner-subgraph(u,v))	Preferential-attachment-score(u,v)	Opposite-direction-friends(u,v)	Jaccard's-coefficient(u,v)	Friends-measure(u,v)	dout(u)	Density(nh-subgraph(u,v))	d(u)	Common-Friends(u,v)	|nh-subgraph+(u,v)|	|nh-subgraph(u,v)|	|Inner-subgraph(u,v)|	
0.4	0.3	0.4	0.2	0.2	0.2	0.4	0.3	0.5	0.7	0.3	0.2	0.3	0.5	0.3	0.2	0.6	Academia
	0.3	0.7	0.5	0.0	0.0	0.4		0.7	0.7		0.2	0.3	0.7	0.2	0.2	0.7	Facebook
0.0	0.7	0.0	0.7	0.7	0.7	0.7	0.3	0.1	0.4	0.2	0.2	0.3	0.1	0.3	0.2	0.3	Flickr
0.2	0.6	0.0	0.7	0.7	0.6	0.8	0.3	0.8	0.6	0.3	0.6	0.5	0.5	0.5	0.5	0.5	TheMarker
0.3	0.6	0.2	0.4	0.5	0.5	0.7	0.6	0.3	0.6	0.5	0.4	0.5	0.3	0.5	0.4	0.5	YouTube

TABLE VII
HARD TRAIN SET - INFOGAIN VALUE OF DIFFERENT FEATURES

Transitive-Friends(u,v)	Total-Friends(u,v)	Shortest-path(u,v)	scc(nh-subgraph+(u,v))	scc(nh-subgraph(u,v))	scc(Inner-subgraph(u,v))	Preferential-attachment-score(u,v)	Opposite-direction-friends(u,v)	Jaccard's-coefficient(u,v)	Friends-measure(u,v)	dout(u)	Density(nh-subgraph(u,v))	d(u)	Common-Friends(u,v)	|nh-subgraph+(u,v)|	|nh-subgraph(u,v)|	|Inner-subgraph(u,v)|	
0.4	0.1	0.2	0.1		0.1	0.1	0.3	0.3	0.3	0.3	0.1	0.2	0.3	0.2	0.2	0.1	Academia
	0.1	0.0	0.0	0.0	0.0	0.2		0.3	0.3		0.2	0.3	0.4	0.2	0.2	0.2	Facebook
0.2	0.6	0.0	0.7	0.7	0.6	0.8	0.3	0.8	0.6	0.3	0.6	0.5	0.5	0.5	0.5	0.5	Flickr
	0.2	0.0	0.0	0.0	0.0	0.3		0.2	0.4		0.4	0.4	0.3	0.4	0.4	0.4	TheMarker
0.5	0.2	0.2	0.2		0.1	0.3	0.6	0.5	0.6	0.3	0.2	0.3	0.5	0.3	0.3	0.2	YouTube

V. Conclusion

This paper presents methods for constructing efficient and effective classifiers based on a set of features that are easy and fast to calculate. We achieved this by defining a set of computationally efficient features and extracting them from five real social network datasets. We created several feature subsets according to their characteristics and evaluated classifier performance for each one of these subsets with several machine learning algorithms. The evaluation demonstrated that our models performed well in terms of accuracy and AUC measures for all the tested datasets, see IV. The best results were obtained using all the features, but we also demonstrated that even if a smaller subset of the features, such as *friends-subset* , is being used, it would be sufficient for obtaining good results.

Another contribution of this paper is that we introduced a topological feature *friends-measure* which is very simple to calculate . The experimental results demonstrated that using the *friends-measure* for link prediction gives better results when compared to the use of the well-known *common-friends* and *Jacquard's-coefficient* features. It was also found that using attribute information gain analysis, the friends-measure is among the most influential features in all of the evaluated networks.

We demonstrated as well that our models provide good results even when tested on links with end vertices that are two hops away from each other. Such results demonstrate the ability to predict link creation within tightly coupled social communities and show that the obtained classifiers can distinguish between friends and non-friends even if the two vertices have at least one common friend (i.e., they are two hops from each other).

Our research currently considers link prediction using only graph topology features. A possible future research direction is to analyze other types of social network features and to examine their impact on link prediction. Examples of other type of features are: content-based features such as posted messages, demographic features (e.g., gender, age etc.) and affiliation-related features. Another future direction would be to examine our algorithms in relation to different domains such as bioinformatics networks.

References

[1] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy, "Make new friends, but keep the old: recommending people on social networking sites," in *Proceedings of the 27th international conference on Human factors in computing systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 201–210. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518735

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic block models for relational data with application to protein-protein interactions," *Proceedings of Ineterational Biometric Society-ENAR Annual Meetings*, 2006.

[3] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, 2005.

[4] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," *SDM Workshop of Link Analysis, Counterterrorism and Security*, 2006.

[5] M. A. Hasan and M. J. Zaki, *Social Network Data Analytics*, C. C. Aggarwal, Ed. Springer, 2011.

[6] D. Liben-Nowell and J. Kleinber, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, 2007.

[7] J. R. Doppa, J. Yu, P. Tadepalli, and L. Getoor, "Chance-constrained programs for link prediction," *In Proceedings of Workshop on Analyzing Networks and Learning with Graphs at NIPS Conference*, 2009.

[8] H. R. Sa and R. B. C. Prudencio, "Supervised learning for link prediction in weighted networks," *III International Workshop on Web and Text Intelligence*, 2010.

[9] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '09. New York, NY, USA: ACM, 2009, pp. 322–335. [Online]. Available: http://doi.acm.org/10.1145/1644893.1644932

[10] W. J. Cukierski, B. Hamner, and B. Yang, "Graph-based features for supervised link prediction," *International Joint Conference on Neural Networks*, 2011.

[11] "Insidefacebook," http://www.insidefacebook.com/2011/06/12/facebook-sees-big-traffic-drops-in-us-and-canada-as-it-nears-700-million-users-worldwide/.

[12] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[13] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[14] "Facebook," http://www.facebook.com/.

[15] "Flickr," http://www.flickr.com/.

[16] "Youtube," http://www.youtube.com/.

[17] "Academia.edu," http://academia.edu/.

[18] "Themarker-cafe," http://cafe.themarker.com/.

[19] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2005.

[20] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)*, August 2009.

[21] "Ijcnn social network challenge," http://www.kaggle.com/c/socialNetwork/Data.

[22] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, November 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278

[24] A. Gibbons, *Algorithmic graph theory*. Cambridge ; New York: Cambridge University Press, 1985.

[25] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, March 1953. [Online]. Available: http://ideas.repec.org/a/spr/psycho/v18y1953i1p39-43.html

[26] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," *In Proceedings of the 7th Python in Science Conference (SciPy2008)*, 2008.

[27] Y. Koren, "The bellkor solution to the netflix grand prize," 2009.

[28] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.