Soc. Netw. Anal. Min. (2014) 4:157
DOI 10.1007/s13278-014-0157-9

ORIGINAL ARTICLE

Link prediction in directed social networks

Daniel Schall

Received: 1 March 2013 / Revised: 7 October 2013 / Accepted: 30 October 2013 / Published online: 4 February 2014
© Springer-Verlag Wien 2014

Abstract In today's online social networks, it becomes essential to help newcomers as well as existing community members to find new social contacts. In scientific literature, this recommendation task is known as link prediction. Link prediction has important practical applications in social network platforms. It allows social network platform providers to recommend friends to their users. Another application is to infer missing links in partially observed networks. The shortcoming of many of the existing link prediction methods is that they mostly focus on undirected graphs only. This work closes this gap and introduces link prediction methods and metrics for directed graphs. Here, we compare well-known similarity metrics and their suitability for link prediction in directed social networks. We advance existing techniques and propose mining of subgraph patterns that are used to predict links in networks such as GitHub, GooglePlus, and Twitter. Our results show that the proposed metrics and techniques yield more accurate predictions when compared with metrics not accounting for the directed nature of the underlying networks.

Keywords Social networks · Link prediction · Metrics · Directed graphs · Patterns

1 Introduction

Social networks have become ubiquitous in our everyday activities. People use social networks to communicate, collaborate, and share information. One of the most profound properties of social networks is their dynamic nature. People join and leave social networks. Also, the circle of friends may frequently change when people establish friendship through social links or when their interest in a social relationship ends and the link is removed. Due to the large number of users being part of today's online communities, it becomes increasingly cumbersome to find new contacts and friends. Many social network platform providers assist their users in establishing new social relations by making recommendations. The meaning of a recommendation varies depending on the concrete social network platform. In platforms such as Facebook (2103), a link between two people is established if both persons agree to have a friendship relation. The resulting network is thus *undirected* because both persons share mutual friendship. Another example of a social network is Twitter (2013). In Twitter, a link between two persons is established if a user is interested in news updates of another user. The link is thus directed because there is no mutual agreement needed to establish a link. The resulting network is *directed* and is also called follower network. User can follow an arbitrary number of other users to receive news or activity updates. The 'follow' feature is in widespread use in social networking services such as Twitter (2013), Facebook (2013), or GooglePlus (2013). In Facebook, the users can follow (or unfollow) news updates of their friends. The social (undirected) link between friends is maintained irrespective of the follow relationship. Recently, collaborative online platforms such as GitHub (2013) also offer social network features (e.g., following). GitHub is an online 'coding' community. Users contribute code and share repositories with the community. The 'follow' feature in GitHub allows users to keep track of updates regarding various software development activities such as coding or bug-fixing.

D. Schall (✉)
Siemens Corporate Technology, Siemensstrasse 90,
1211 Vienna, Austria
e-mail: daniel.schall@gmail.com; daniel.schall@siemens.com

All of the before mentioned platforms have a large number of users and benefit from 'follow' recommendations. Such recommendations can be formulated as a link prediction task. Link prediction in a directed follower network has the purpose of giving recommendations which a given user should follow. Despite some existing literature in the area of link prediction on Twitter (for example, see Brzozowski and Romero 2011; Romero and Kleinberg 2010), or prediction of positive/negative edges (Leskovec et al. 2010), there is relatively little work on link prediction in directed social networks. As reported in a recent survey (Lu and Zhou 2011), the existing studies on link prediction overwhelmingly focus on undirected networks. This work specifically addresses the link prediction problem in directed social networks.

The essential approach we follow in this work is to measure the similarity between a pair of nodes using *structural* information. We assume that a social network is modeled as a directed graph $G(V, E)$, where vertices V in the graph depict people and edges E between vertices relations between people. Structural information is, for example, the number of common neighbors between two nodes. Generally speaking, similarity of nodes can be measured as the number of common features. The goal of link prediction is to predict whether a link between two users will be established or if a link in a partially observed network is missing. The latter case is a common problem when the social network is obtained through a crawling procedure.

Here, we provide the following key contributions:

- In this work, we propose link prediction techniques using graph patterns (*triads*). Predictions are then given based on the probability that a given type of triad pattern will be closed.
- Here, we introduce a metric called *Triadic Closeness*. The application of the metric is discussed and evaluated.
- We have designed and implemented a link prediction framework. The main building blocks of the framework are discussed.
- We present an analysis of our link prediction techniques. We use three different social networks including GitHub, GooglePlus and Twitter to validate the proposed approach.

This work is structured as follows: Sect. 2 discusses related work on the context of social networks and link prediction. Standard similarity metrics are introduced which will be compared against our proposed approach. In Sect. 3, our prediction approach is introduced followed by a brief description of the link prediction framework in Sect. 4. The results and experiments are discussed in Sect. 5. The conclusion with outlook to further work is given in Sect. 6.

2 Related work

We discuss related work by (1) highlighting literature and related approaches with respect to link prediction and node similarity indices and (2) local structures and patterns in social networks. Similarity indices are structured in *local*, *global*, and *semi-local* indices.

Local similarity indices A wide range of similarity metrics exist that can be used to predict links based on 'local' information (Adamic and Adar 2001; Aiello et al. 2012; Esslimani et al. 2011; Leicht et al. 2006; Ravasz et al. 2002; Rettinger et al. 2012; Salton and McGill 1986; Zhou et al. 2009). Local information is typically obtained by comparing degree of overlap of two individual friendship networks. Liben-Nowell et al. (Liben-Nowell and Kleinberg 2003) systematically compared a number of local similarity indices in many real networks. These metrics focus on undirected graphs without considering directed relations. The advantage of local indices is that they can be computed for large-scale networks and do not require a huge amount of computational resources.

Global Similarity Indices Global metrics take the properties of the whole social network into account. The Katz index (Katz 1953) is based on the ensemble of all paths between two nodes in the network. The index is computed as the sum over the collection of all paths and is exponentially damped by length to give the shorter paths more weight. Another class of metrics are random walk techniques. Well-known algorithms such as PageRank (Page et al. 1998) can be used to compute global importance metrics. The prediction is then based on the node's PageRank importance score. PageRank can be personalized to perform ranking with respect to a certain 'contexts' or topics (Schall 2012). A direct application of personalized PageRank are supervised random walks (SRW). In (Backstrom and Leskovec 2011), SRW were proposed to recommend links in networks such as Facebook. SimRank (Jeh and Widom 2002) is based on the idea that two nodes are similar if they are related to similar nodes. Global similarity indices naturally require information regarding the whole topology of the social network. Indeed, this information may not be available due to, for example, partially observed networks or in cases where the platform is decentralized. Another important aspect is performance and resource consumption. The calculation of global indices may be very time consuming and for large-scale networks the computation may not be feasible.

Semi-local indices Instead of taking the whole topology into account, semi-local indices omit information that makes little contribution to improve the prediction algorithm's accuracy. The Local Path Index (Meng et al. 2011; Zhou et al. 2009) for example, provides a trade-off between computational complexity and accuracy. Local

Random Walks (Backstrom and Leskovec 2011; Liu and Lu 2010) follow a similar idea by omitting information from very distant neighbors in the network.

Further methods for link prediction include hierarchical models (Clauset et al. 2008), stochastic block models (Airoldi et al. 2008; Holland et al. 1983; White et al. 1976), probabilistic models (Lu and Zhou 2011), and methods considering positive/negative links (Symeonidis and Mantas 2013) (see (Lu and Zhou 2011) for details on models and methods).

Patterns, triads and motifs The approach as proposed in this work takes the directed nature of follower networks into account. *Triadic closure* in social networks is the hypothesis that the formation of an edge between u and v is strongly dependent upon the degree of overlap of u's and v's individual friendship networks (for example, see Snijders 2012; Wasserman et al. 1994). However, an important theory in this context is the 'strength of weak ties' (Granovetter 1973) stressing the cohesive power of seemingly less important ties. Holland and Leinhardt (Holland and Leinhardt 1970) developed many important theories about social relations and how to detect structure in directed networks. As a more general conceptual framework, network motifs (Alon 2007; Milo et al. 2002) represent elementary building blocks in complex networks. Complex networks include social, technological, or biological networks. We build upon these ideas and propose link prediction considering triad patterns in social networks.

Finally, previously we stressed the importance of social networks and formations of social groups (teams) in the context of collaborative environments and novel crowdsourcing environments (Sautter and Bhm 2013; Schall 2012; Schall and Skopik 2012). We foresee important applications of link prediction in these areas.

Table 1 Similarity-based metrics

Metric	Definition	Description
Common Neighbors (CN)	$\lvert\Gamma(u) \cap \Gamma(v)\rvert$	Intersection set size of joint neighbors between nodes u and v
Salton Index (SA)	$\frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\sqrt{k_u \times k_v}}$	The degree of u and v is depicted by k_u and k_v, respectively. In literature, the Salton index (Salton and McGill 1986) is also called the cosine similarity
Jaccard Index (JA)	$\frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\lvert\Gamma(u)\cup\Gamma(v)\rvert}$	Jaccard similarity index with $\Gamma(u) \neq \emptyset$ and $\Gamma(v) \neq \emptyset$.
Sørensen Index (SO)	$\frac{2\lvert\Gamma(u)\cap\Gamma(v)\rvert}{k_u+k_v}$	The Sørenson index (Sørensen 1957) is mainly used for ecological data. The index is identical to the Dice's coefficient
Hub Promoted Index (HP)	$\frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\min(k_u,k_v)}$	The links adjacent to hubs are likely to be assigned higher scores, since the denominator min (k_u, k_v) is determined by the lower degree (Ravasz et al. 2002)
Hub Depressed Index (HD)	$\frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{\max(k_u,k_v)}$	Similar to HPI but with the opposite effect with regards to adjacent hub links
Leicht-Holme-Newman Index (LHN)	$\frac{\lvert\Gamma(u)\cap\Gamma(v)\rvert}{k_u \times k_v}$	The denominator $k_u \times k_v$ is proportional to the expected number of common neighbors (Leicht et al. 2006)
Adamic-Adar Index (AA)	$\sum_{z\in\Gamma(u)\cap\Gamma(v)} \frac{1}{\log(k_z)}$	The index assigns the less-connected neighbors more weight than CN (Adamic and Adar 2001)
Resource Allocation Index (RA)	$\sum_{z\in\Gamma(u)\cap\Gamma(v)} \frac{1}{k_z}$	Similar to AA, RA depresses the contribution of the high-degree common neighbors (Zhou et al. 2009)

3 Link prediction using graph patterns

3.1 Similarity-based metrics

The focus of this work are local similarity indices and their extension towards link predictionc using graph patterns. Table 1 lists a set of well-known similarity metrics. The shared feature of the metrics is that computation of similarity is based on the set of joint neighbors. These metrics provide the basis for the definition of our *triadic closeness* (TC) similarity metric. The definition of the TC metric will be provided in the following. Furthermore, the metrics in Table 1 will be used in a comparative study to test the effectiveness of the proposed technique. Table 1 provides a mathematical definition along with a brief description of the given metric. Given node u, the set of neighbors is depicted by $\Gamma(u)$. The degree of node u is depicted as $k_u = |\Gamma(u)|$.

These metrics have the drawback that they do not account for the directed nature of follower networks. In other words, the metrics in Table 1 do not allow for differentiation whether a link will be established from, say, u to v or from v to u. As a next step, we introduce patterns to account for directed links in social networks.

3.2 Triad patterns

In social network theory, a basic unit of analysis is a dyad. In undirected networks, a dyad is a pair of nodes who may share a social relation with one another. In directed networks, a dyad consists of a pair of nodes who may share a social relation through mutual links, an unreciprocated

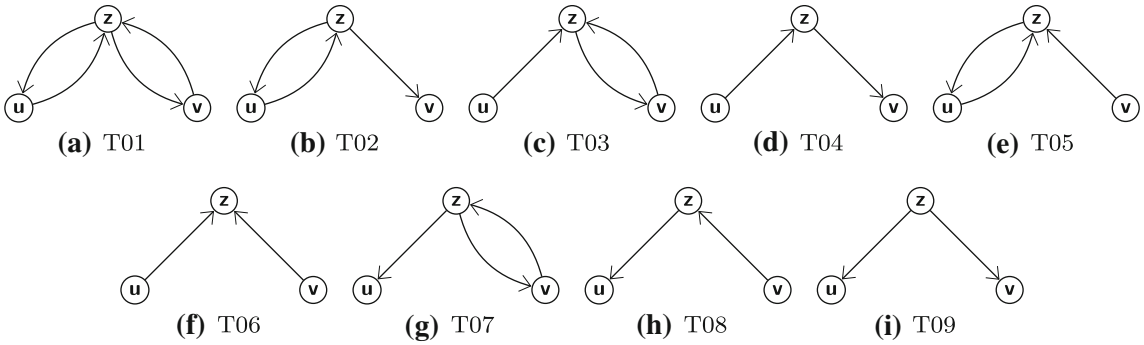

Fig. 1 Triad graph patterns

relation, or no relation. Unreciprocated means that one node is interested in the other node but not vice versa. A triad is a set of three parties, which consists of three dyads. A triad is 'closed' if all nodes are linked with each other in some manner. A closed triad is also called triangle.

Figure 1 shows triad patterns of the actors u, z, and v. Edges are directed because our aim is to model patterns in directed social networks (e.g., follower networks). All patterns are open triads with z being the common neighbor of u and v. The questions with regards to link prediction can be stated as follows:

- What is the likelihood that u will establish a link towards v?
- In a partially observed network, is there a missing link pointing from u to v?

In Fig. 1, all possible connectivity configurations between u, z, and v are shown with the condition that u and v are not directly connected. In this work, open triads are labeled as T0X where X is the running index with $X = [1, 9]$. The pattern T01 shows the case where u and z as well as v and z are mutually connected. According to the theory of triadic closure, the chances are high that u will also connect to v (i.e., z's friends will likely become u's friends). In a follower network, the pair u and z and v and z would mutually follow each other. In T02, only u and z are mutually connected to each other. The node v is followed by z but the relationship is not reciprocated. T03, T05, and T07 depict complementary cases where a mutual relation among one dyad exists. The other cases depicted by T04, T06, T08, and T09 show patterns without mutual relations among the dyads. The goal of link prediction is to determine which of the triads are or will be closed (i.e., becoming a triangle). A triad can be closed as follows if u establishes a link to v, v establishes a link to u, or if u and v establish a link mutually.

The following figures show closed triads based on T01 and T09 (the first and the last pattern of Fig. 1 are shown for brevity). Figure 2 shows the patterns where the triads

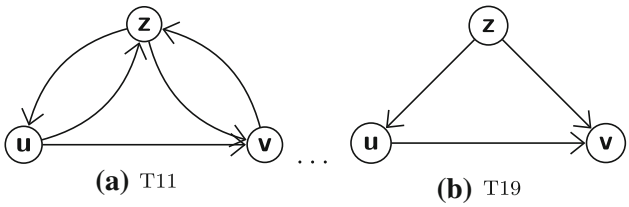

Fig. 2 Closed triads T1X

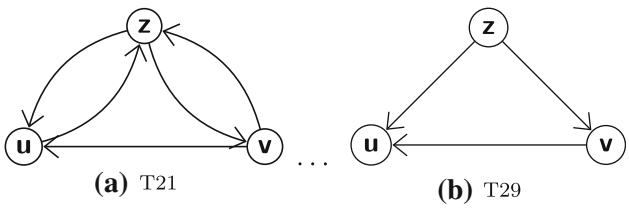

Fig. 3 Closed triads T2X

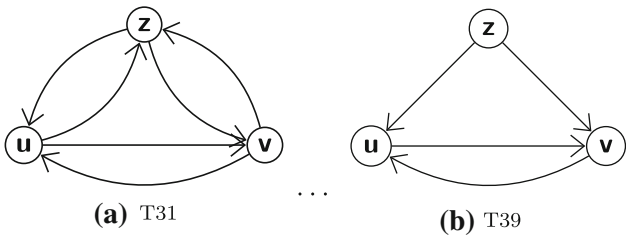

Fig. 4 Closed triads T3X

are closed from u to v. The patterns are labeled similarly as in Fig. 1 but with a base offset of 10. Thus, the label of triads that are closed via u to v is T1X with $X = [1, 9]$.

In the same manner, the label of triads that are closed via v to u is T2X with $X = [1, 9]$ (see Fig. 3).

Finally, if the triads are closed by mutually connected u and v (see Fig. 4), the labels T3X with $X = [1, 9]$ are applied. To summarize our discussions regarding triad patterns, triads that are relevant for link prediction may have 36 different configurations with regards to how nodes

are connected to each other through directed links. Open triads have 2 connected dyads and have 2 to 4 links. Closed triads have 3 connected dyads and have 3 to 6 links. The next step is to introduce a novel metric to calculate a score for link prediction based on the presented triad patterns.

3.3 Triadic closeness

When considering a given pattern T0*X* (open triads as depicted by Fig. 1), the basic question is which of those patterns are likely closed triads (in the case of missing links) or which of those patterns will likely be closed in the future. Here, we introduce *triadic closeness* (TC) to measure how close a pair of disconnected nodes are in terms of how the pair is connected through triads. TC is based on the following basic idea:

$$\text{Triadic closeness} \propto \frac{\text{Number of closed triads}}{\text{Number of potentially closed triads}}$$

Triadic Closeness is thereby based on the ratio of the number of closed triads versus the number of potentially closed triads. Indeed, the chance that a given T0*X* triad will be closed depends on the actual social network and is most likely not the same for all follower networks. We define the TC score of the pair u and v in a directed graph G as follows:

$$\mathrm{TC}_{uv} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} w^{\mathrm{P}}(u, v, z) \times w(z) \tag{1}$$

The score is calculated over all common neighbors. For a given neighbor z, the product is calculated by the triad weight $w^{\mathrm{P}}(u, v, z)$ times the neighbor specific weight $w(z)$. The triad weight $w^{\mathrm{P}}(u, v, z)$ is defined as follows:

$$w^{\mathrm{P}}(u, v, z) = \frac{F(T(u, v, z) + 10) + F(T(u, v, z) + 30)}{F(T(u, v, z))} \tag{2}$$

The function $T(u, v, z)$ retrieves the triad pattern ID that matches the triad u, v, and z. The term $(T(u, v, z) + 10)$ simply means that the ID of the closed triad counterpart of $T(u, v, z)$ is obtained (closed via u to v). Similarly, $(T(u, v, z) + 30)$ gets the closed counterpart triad ID wherein $T(u, v, z)$ is closed through mutual links between u and v. The function $F(\cdot)$ retrieves the frequency of the given triad pattern. Prior to performing the calculation of $w^{\mathrm{P}}(u, v, z)$, the frequencies of triads in a particular social network are computed by an algorithm and saved in a database. Afterwards, $F(\cdot)$ simply retrieves the triad frequency from a database (zero if the given triad was not detected in the graph).

The neighbor specific weight $w(z)$ can be tuned to account for the characteristics of specific social networks. For the basic case with $w(z) = 1, \mathrm{TC}_{uv}$ is only based on $w^{\mathrm{P}}(u, v, z)$. We define the weight as $w(z) = \frac{1}{k_z}$ to give less-connected neighbors more weight and thus TC_{uv} becomes:

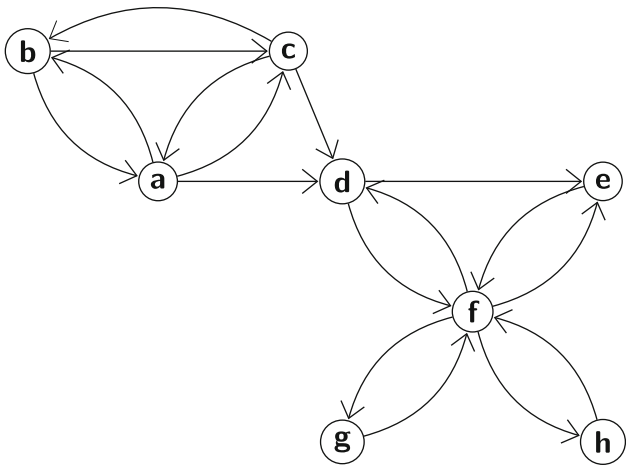

Fig. 5 Example network

$$\mathrm{TC}_{uv} = \sum_{z \in \Gamma(u) \cap \Gamma(v)} w^{\mathrm{P}}(u, v, z) \times \frac{1}{k_z} \tag{3}$$

Neighbors that are unique to only a few users are weighted more with $w(z) = \frac{1}{k_z}$ than popular neighbors. Popular neighbors are those with a high degree k_z and especially in networks such as Twitter these neighbors may be celebrities that may not have great significance for the triadic closure process. From a technical point of view, TC_{uv}'s behavior is comparable to Adamic-Adar Index (AA) (Adamic and Adar 2001) or the Resource Allocation Index (RA) (Zhou et al. 2009) (see also Table 1). Indeed, the indices lack the notion of patterns and have been designed with undirected friendship networks in mind.

3.4 Triadic closeness example

To give a concrete example, consider the artificial network as depicted by Fig. 5 and suppose triadic closeness TC_{gh} shall be calculated between the nodes g and h.

The triad frequencies are listed in Table 2. Algorithm 1 shows the steps for counting the frequency of patterns in graph G.

Algorithm 1 Pattern counting algorithm.

```
input: directed graph G
for each Vertex u ∈ G do
    for each Vertex z ∈ getNeighbors(G, u) do
        for each Vertex v ∈ getNeighbors(G, z) do
            if equals(v, u) then
                continue
            end if
            id ← getPatternId(G, u, z, v)
            count(id) // increment count by 1
        end for
    end for
end for
```

Table 2 Triads in example network

ID	Pattern	Frequency ↓
1	$u \leftrightarrow z \leftrightarrow v$	10
31	$u \leftrightarrow z \leftrightarrow v \leftrightarrow u$	6
2	$u \leftrightarrow z \rightarrow v$	2
3	$u \rightarrow z \leftrightarrow v$	2
4	$u \rightarrow z \rightarrow v$	2
5	$u \leftrightarrow z \leftarrow v$	2
7	$u \leftarrow z \leftrightarrow v$	2
8	$u \leftarrow z \leftarrow v$	2
12	$u \leftrightarrow z \rightarrow v \leftarrow u$	2
27	$u \leftarrow z \leftrightarrow v \rightarrow u$	2
36	$u \rightarrow z \leftarrow v \leftrightarrow u$	2
11	$u \leftrightarrow z \leftrightarrow v \leftarrow u$	1
21	$u \leftrightarrow z \leftrightarrow v \rightarrow u$	1
32	$u \leftrightarrow z \rightarrow v \leftrightarrow u$	1
33	$u \rightarrow z \leftrightarrow v \leftrightarrow u$	1
35	$u \leftrightarrow z \leftarrow v \leftrightarrow u$	1
37	$u \leftarrow z \leftrightarrow v \leftrightarrow u$	1

As shown in Fig. 5, the node f connects g and h via triad T01. Related to T01 for calculation are T11 and T31. The weight $w^{\mathrm{P}}(u, v, z)$ for the network in Fig. 5 is given as $w^{\mathrm{P}}(u, v, z) = \frac{6+1}{10} = 0.7$. Thus, TC_{gh} is given as $\mathrm{TC}_{gh} = 0.7 \times \frac{1}{4} = 0.175$. Using the triadic closeness concept, with a probability of 0.17, T01 will be closed from g to h.

4 Link prediction framework architecture

One of the goals of the presented work was to design a modular and reusable framework for link prediction. The framework must be able to handle different social networks that can range from a few thousands to millions of nodes in the social graph. This section gives an overview of the link prediction framework architecture and a description of the evaluation methodology.

The main building blocks and layers are depicted by Fig. 6. The overall framework is segmented into a multi-layer architecture consisting of three layers: (1) data layer, (2) prediction layer, and (3) presentation layer. Each layer has distinct responsibilities and each block within a given layer provides a well-defined set of interfaces. The central goal of the system is to provide an extensible framework that can be enhanced with new metrics and prediction methods. In addition, it should be easy to add new social networks (datasets) and to compare the results of different experiments and algorithms.

Data layer The bottom-most layer is concerned with low-level data handling and persistence management. With regards to the basic link prediction task, the data layer passes an instance of a social network graph to the upper layer. Our framework has the ability to access data from (a) the *Flat File Store* and (b) the *Database*. The Flat File Store is used for simple social network data files that are small to medium in their size (e.g., 10^3–10^5 nodes) and is read only. The *Parser* reads and interprets files stored in the Flat File Store. The Parser performs the pre-stage processing for the *Graph Mapper*, which creates the social network graph including node and edge attributes (if available). The Database is able to manage large social networks (large networks may consist of up to 5×10^7 nodes[1]) that may also include details regarding user profiles and user activity. The *Query Handler* interfaces with the database to retrieve the social networks graph structure, user profile details, other social network-related information and also information related to patterns and experiments. The *Update Handler* is responsible for persistence management and writes data to the database. The *Import Library* allows external social networks and networks stored in the Flat File Store to be migrated to the Database. Migration from the Flat File Store is needed if performance is insufficient (read operations) or if the management of social network (meta)data through the Flat File Store becomes impractical. Another source of information are external APIs of social network or community platforms. GitHub, for example, provides an API (2013) to retrieve the follower network. The Import Library provides a rate-aware API invocation scheduler to retrieve large social networks respecting the social network providers' API policies (e.g., number of invocations per hour).

Prediction Layer The middle layer groups the logic for metric calculation, pattern mining (triad detection), prediction, and prediction result evaluation. The *Similarity Library* is responsible for calculating various similarity indices including local similarity indices (see Table 1), global similarity indices, and semi-local indices. The system block *Similarity Metrics* computes local similarity indices (including TC) and semi-local indices including Local Path Index (see (Lu and Zhou 2011)) and the Shortest Path Index (i.e., the average Dijkstra Shortest Path Index between, say, u and v's neighbors $\Gamma(v)$ to measure similarity between u and v). *Iterative Algorithms* have been designed to calculate metrics such as SimRank (Jeh and Widom 2002) and PageRank variants such as personalized PageRank (Jeh and Widom 2003; Page et al. 1998; Schall 2012). Note, the evaluation of global and semi-local indices is not within the scope of this work. Here, we focus on local indices in conjunction with triad patterns. The

[1] The upper bound for which the Data Layer has been tested was a network consisting of approximately 5×10^7 nodes and 1.5×10^9 edges.

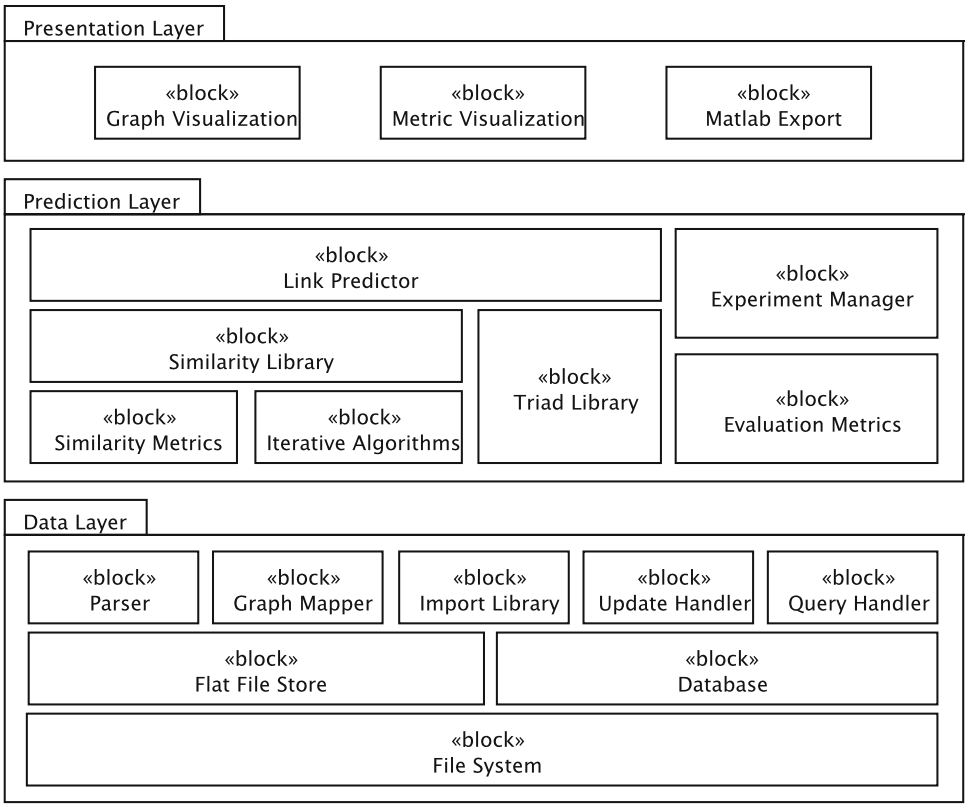

Fig. 6 Link prediction framework overview

relationship between (local) triad patterns and global or semi-local indices is a whole new subject of investigation itself. Next to the Similarity Library, the *Triad Library* provides the capabilities for triad pattern detection and caching. Clearly, scanning the entire social network graph for triad patterns is a time-consuming task with complexity $\mathcal{O}(m)$ where m is the number of edges in the graph (see Batagelj and Mrvar 2001) for related triad detection algorithms).

Thus, triad pattern mining is usually performed once for a given social network and subsequent metric calculations use the precomputed triad frequencies.

The *Link Predictor* is the main component that performs the prediction task. This can be done to predict future links, which do not yet exist, or predict links, which are 'missing' (unobserved). Here, we focus on the latter case where we assume that certain links are missing between pairs of nodes. To test the metrics' accuracy, we divide the set of edges E randomly into the prediction (or training) set E^{P} and the validation set E^{V}. The prediction algorithm bases its calculation upon the prediction graph $G^{\mathrm{P}}(V, E^{\mathrm{P}})$, whereas the accuracy of the prediction results is determined by inspecting the missing (randomly removed) edges in E^{V}. Note, no information from the set E^{V} is allowed to be used for prediction so that $E^{\mathrm{P}} \cup E^{\mathrm{V}} = E$ and $E^{\mathrm{P}} \cap E^{\mathrm{V}} = \emptyset$. Furthermore, to speed up computation of prediction results, the Link Predictor can perform node sampling to calculate predictions for a subset of node pairs instead of calculating predictions for all node pairs in the entire graph. For that purpose, the predictor samples a set of random nodes U^{P} with $k > 0$ and divides the set into two subsets U^{R} and U^{T} with $U^{\mathrm{R}} \cup U^{\mathrm{T}} = U^{\mathrm{P}}$, $U^{\mathrm{R}} \cap U^{\mathrm{T}} = \emptyset$, and $U^{\mathrm{P}} \subset U$. The set U^{P} contains all nodes that are used for link prediction, the set U^{R} contains the root nodes (source vertices of predicted links) and U^{T} contains the target nodes (target vertices of predicted links). The set E^{V} contains only directed links whose source vertex is in U^{R} and whose target vertex is in U^{T}. For one given experiment, the same node set U^{P} and edge set E^{V} is used to be able to compare the results of different metrics among each other.

The basic steps of the Link Predictor are straightforward. Algorithm 2 depicts the steps:

Algorithm 2 Link prediction algorithm.

```
1: input: G(U, E^P), U^R, U^T, E^V
2: for each User u ∈ U^R do
3:   for each User v ∈ U^T do
4:     // True Answer
5:     answer ← HasEdge(u, v, E^V)
6:     // Calculate Similarity Scores
7:     for each Metric m ∈ M do
8:       s_uv ← CalculateScore(u, v, m, G)
9:       // Save Result to Experiment Database
10:      SaveResult(u, v, m, s_uv, answer)
11:    end for
12:  end for
13: end for
```

The predictor loops through U^R and U^T and calculates similarity scores s_{uv} using each metric $m \in M$ provided by the Similarity Library. M can be configured dynamically by enabling/disabling the desired metrics to be used in each experiment. The prediction result s_{uv} and the actual true answer {0, 1} (*HasEdge* is a binary classifier that determines whether the set E^V contains a directed edge between u and v) are saved in the experiment database.

To compare the results of different similarity algorithms, the *Evaluation Metrics* component provides standardized comparison methods. In particular, we compare results using the *receiver operating characteristic* (ROC) curve. ROC curves are commonly used in the machine learning community for the link prediction task (Aiello et al. 2012; Clauset et al. 2008). ROC curves are created by plotting the true positive rate over the false positive rate. The *area under the ROC curve* (AUC) (Bradley 1997) can be interpreted as the probability that a randomly chosen missing link (i.e., a link in E^V) is given a higher score than a randomly chosen non-existing link (Lu and Zhou 2011).

Higher AUC values, which are in the range [0, 1], indicate better prediction performance. Another common metric to measure a prediction algorithm's accuracy is *HitRatio* (or recall). In general, HitRatio is defined as the ratio of selected relevant items to the number of relevant items. For example, the HitRatio is typically measured at a threshold HitRatio@n where n is the number of selected items. In this work, we focus on both ROC curves and AUC as well as HitRatio for experiment evaluation and metric comparison. The *Experiment Manager* saves and retrieves experiment results from the Database and computes aggregates of results.

Presentation Layer The frontend of the prediction framework is a presentation layer that has visualization and export capabilities. The *Graph Visualization* allows to view typical graph properties by mapping node/edge features into a visual representation. To do so, the correspondence between discrete or continuous values and visual properties (color, node size, etc.) needs to be established. The *Metric Visualization* is the most important tool for evaluating the results of the similarity algorithms and link predictor. ROC curves help to identify which methods and parameter settings are best suited for a given type of social network. The various network idiosyncrasies such as average degree $\langle k \rangle$ may demand for metric tuning. A detailed discussion regarding metric accuracy will follow in Sect. 5. The Metric Visualization helps to understand the accuracy and suitability of different metrics. The *Matlab Export* allows to export experiment results to a Matlab compatible format to utilize various Matlab toolboxes.

5 Experiments and discussions

We obtained three directed social follower networks to compare different metrics and to validate the suitability of TC. The networks (the whole follower network or subsets thereof) include GitHub (2013), GooglePlus (2013), and Twitter (2013).

5.1 Dataset characteristics

GitHub The first network is based on GitHub's follower network (2013). The graph was imported in our prediction framework through the GitHub API (2013) in December 2012. The basic network characteristic in terms of follower (indegree) distribution is depicted by Fig. 7. The plot shows a power-law distribution with the basic property $N(k) \sim k^{-1.49}$ where $N(k)$ is the number of nodes with indegree k. The follower graph counts 1,105,150 users and 1,898,034 following relations (edges). Nearly 70 % of users (767, 975) have no followers (zero indegree) and again about 70 % of users (769, 283) do not follow any other user (zero outdegree).

GooglePlus. The second network represents a subset of nodes and edges from GooglePlus (2013). We have obtained the network (plain text files) from the Stanford Large Network Dataset Collection (2013). A description of the network is also given in (McAuley and Leskovec 2012). The degree distribution is depicted by Fig. 8. The network consists of 107,614 nodes and 13,673,453 edges. At a technical level, the network is managed within the prediction framework's Flat File Store. The average degree

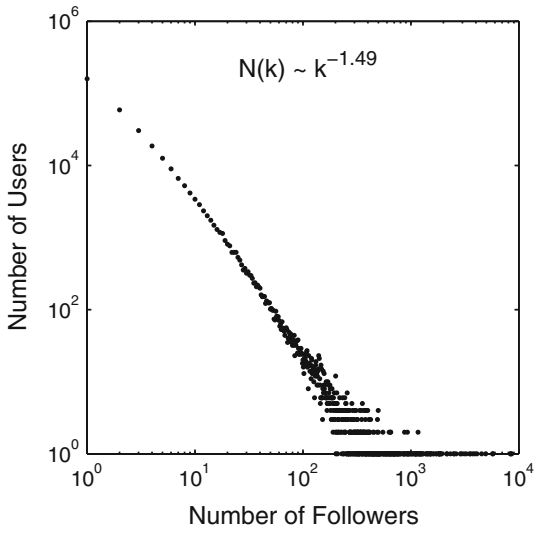

Fig. 7 Indegree distribution of GitHub

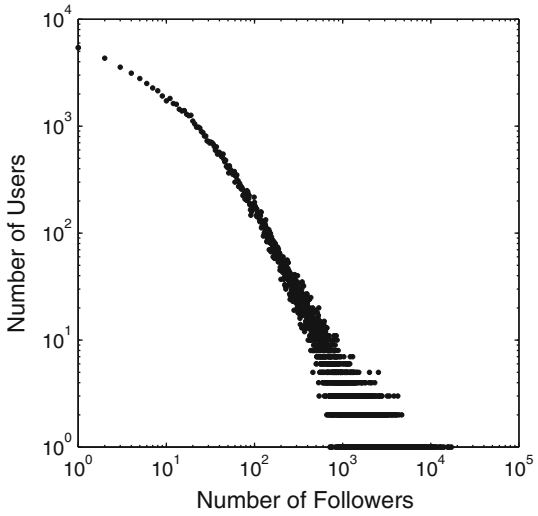

Fig. 8 Indegree distribution of GooglePlus

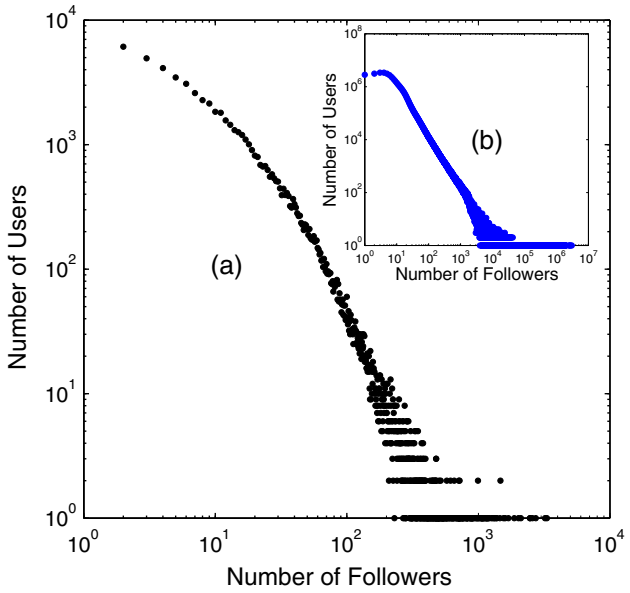

Fig. 9 Indegree distribution of Twitter

$\langle k \rangle = 127$ in this network is much higher than in the GitHub-based follower network, which has only $\langle k \rangle = 1.7$.

A possible explanation for the high differences in the average degree $\langle k \rangle$ is the primary purpose of the platforms. GitHub is a platform for hosting and sharing source code repositories and the 'follow' feature is by many people used to follow top-developers. In GooglePlus, many people follow other people they personally know and use the platform to maintain social relations.

Twitter. The third network is based on a subset of nodes and edges of Twitter (2013). The network was also obtained from the Stanford Large Network Dataset Collection (2013) and is also managed within the Flat File Store. The network counts 81,306 nodes and 1,768,149 edges, thereby making it the smallest network in our experiments. The degree distribution is depicted by Fig. 9.

The average degree is given as $\langle k \rangle = 21.7$. In addition, we show the degree of a much larger Twitter-based network in Fig. 9 in the top-right corner to show how the presented network subset relates to the large network. The large network was obtained in July 2009 by (Kwak et al. 2010) and counts roughly 5×10^7 nodes and 1.5×10^9 edges. Both networks follow a similar distributional shape. In our experiments, only the smaller network has been used. In combination with the other larger networks (GitHub and GooglePlus), the smaller Twitter-based network provides a sufficient basis to compare link prediction metrics.

Table 3 Configuration settings for link prediction

Configuration	GH	GP	TW
$\lvert U^{\mathrm{P}} \rvert$	110,515	107,614	81,306
$\lvert E^{\mathrm{V}} \rvert$	29,759	2,709,731	333,577
$\lvert U^{\mathrm{R}} \rvert$	100	100	100

5.2 Configuration

Here, we discuss the link prediction configuration settings that were used to perform experiments. Experiments have been performed using the three previously introduced datasets: GitHub (**GH**), GooglePlus (**GP**), and Twitter (**TW**). Table 3 lists the prediction user set size $\lvert U^{\mathrm{P}} \rvert$, the validation set size $\lvert E^{\mathrm{V}} \rvert$ and the root set size $\lvert U^{\mathrm{R}} \rvert$.

For GitHub, the prediction user set U^{P} consists of 10 % of the users which are connected through 148,796 edges. From those edges, we sampled 29,759 random edges (20 %) and added them to E^{V}. The root set U^{R} is populated with 100 nodes. The size of the prediction target set U^{T} can be easily calculated as $U^{\mathrm{T}} = U^{\mathrm{P}} - U^{\mathrm{R}}$. In GooglePlus, we use the entire user base for U^{P}, which also results in approximately the same size of U^{P} as for GitHub's prediction user set. E^{V} consists of 2,709,731 edges (20 %) and the root set U^{R} consists also of 100 nodes. The Twitter-based dataset has the smallest number of nodes and, in the same manner as for GooglePlus, we also select all nodes for U^{P}. E^{V} has 333,577 edges (again 20 %) and the same root set size as for GitHub and GooglePlus is applied.

Using the configuration settings in Table 3, we obtain the graph $G^{\mathrm{P}}(U, E^{\mathrm{P}})$ upon which triad pattern mining and prediction is performed. The relative frequency of each triad pattern in G^{P} is listed in Table 4. The total number of triad patters in each network is 423,034,532 (GitHub), 14,402,457,330 (GooglePlus), and 296,075,286 (Twitter). The patterns at the top in Table 4 are open triads T01–T09

Table 4 Ratio of triads in different social networks

ID	Pattern	GH (%)	GP (%)	TW (%)
9	$u \leftarrow z \rightarrow v$	53.23	9.97	7.04
6	$u \rightarrow z \leftarrow v$	39.09	18.79	39.96
8	$u \leftarrow z \leftarrow v$	1.50	12.71	5.95
4	$u \rightarrow z \rightarrow v$	1.37	12.71	5.95
7	$u \leftarrow z \leftrightarrow v$	1.22	5.69	4.36
2	$u \leftrightarrow z \rightarrow v$	1.14	5.69	4.36
5	$u \leftrightarrow z \leftarrow v$	0.63	3.93	6.21
3	$u \rightarrow z \leftrightarrow v$	0.63	3.93	6.21
1	$u \leftrightarrow z \leftrightarrow v$	0.23	1.44	4.07
28	$u \leftarrow z \leftarrow v \rightarrow u$	0.10	2.74	0.86
29	$u \leftarrow z \rightarrow v \rightarrow u$	0.10	2.74	0.86
19	$u \leftarrow z \rightarrow v \leftarrow u$	0.10	2.74	0.86
26	$u \rightarrow z \leftarrow v \rightarrow u$	0.10	2.74	0.86
16	$u \rightarrow z \leftarrow v \leftarrow u$	0.09	2.74	0.86
14	$u \rightarrow z \rightarrow v \leftarrow u$	0.09	2.74	0.86
25	$u \leftrightarrow z \leftarrow v \rightarrow u$	0.05	0.62	0.70
39	$u \leftarrow z \rightarrow v \leftrightarrow u$	0.05	0.62	0.70
13	$u \rightarrow z \leftrightarrow v \leftarrow u$	0.04	0.62	0.70
27	$u \leftarrow z \leftrightarrow v \rightarrow u$	0.04	1.41	0.87
12	$u \leftrightarrow z \rightarrow v \leftarrow u$	0.04	1.41	0.87
36	$u \rightarrow z \leftarrow v \leftrightarrow u$	0.04	1.41	0.87
31	$u \leftrightarrow z \leftrightarrow v \leftrightarrow u$	0.03	0.29	0.97
11	$u \leftrightarrow z \leftrightarrow v \leftarrow u$	0.01	0.21	0.53
33	$u \rightarrow z \leftrightarrow v \leftrightarrow u$	0.01	0.21	0.53
21	$u \leftrightarrow z \leftrightarrow v \rightarrow u$	0.01	0.21	0.53
37	$u \leftarrow z \leftrightarrow v \leftrightarrow u$	0.01	0.21	0.53
32	$u \leftrightarrow z \rightarrow v \leftrightarrow u$	0.01	0.21	0.53
35	$u \leftrightarrow z \leftarrow v \leftrightarrow u$	0.01	0.21	0.53
17	$u \leftarrow z \leftrightarrow v \leftarrow u$	0.00	0.15	0.27
23	$u \rightarrow z \leftrightarrow v \rightarrow u$	0.00	0.15	0.27
22	$u \leftrightarrow z \rightarrow v \rightarrow u$	0.00	0.15	0.27
38	$u \leftarrow z \leftarrow v \leftrightarrow u$	0.00	0.15	0.27
15	$u \leftrightarrow z \leftarrow v \leftarrow u$	0.00	0.15	0.27
34	$u \rightarrow z \rightarrow v \leftrightarrow u$	0.00	0.15	0.27
18	$u \leftarrow z \leftarrow v \leftarrow u$	0.00	0.10	0.13
24	$u \rightarrow z \rightarrow v \rightarrow u$	0.00	0.10	0.13

followed by closed triads T11–T36. The patterns are sorted in descending order by the column **GH**.

The most common triad pattern in GitHub with 53.23 % is where both u and v are followed by z but neither u nor v follow z. This pattern has a relative frequency of 7.04 % in Twitter, thereby being the second most common pattern in Twitter, and a relative frequency of 9.97 % in Google-Plus, thereby being the fourth most common pattern in GooglePlus. The second most common pattern in GitHub, and the most common pattern in GooglePlus and Twitter, is the pattern where both u and v follow z but z follows neither of them. Triadic Closeness is defined as the

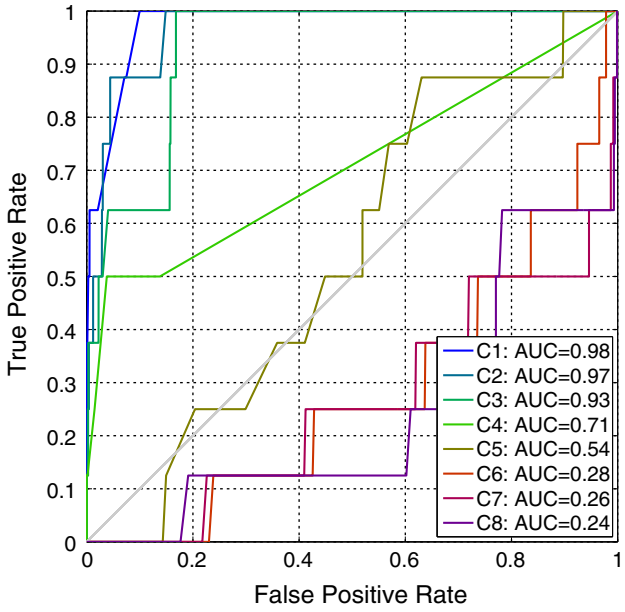

Fig. 10 ROC curves for GitHub-based results

Table 5 AUC Classes for GitHub-based results

Class	Definition	AUC
C1	TC	0.98
C2	RA	0.97
C3	AA	0.93
C4	CN	0.71
C5	HP	0.54
C6	SA	0.28
C7	LHN	0.26
C8	JA, SO, HD	0.24

likelihood that a given open triad (T01–T09) will be closed in a given social network. Thus, the triad patterns T01–T09 are seen in relation with the closed triads to determine the closeness between two nodes.

In the following, the prediction results are presented by plotting ROC curves and calculating AUC for each metric. TC is based on the frequencies of respective triads in Table 4.

5.3 Prediction results

In all our experiments, we use the metrics as defined in Table 1 and the introduced triadic closeness TC. The configuration settings for the experiments are given in Table 3. As a general note, an AUC value above 0.5 indicates that a prediction algorithm performs better than pure chance. Higher AUC indicates better prediction accuracy. With regards to ROC curves, we group metrics into a single *class* if their AUC values are identical.

Table 6 HitRatio (%) for GitHub-based results

Metric	HitRatio@10	HitRatio@30	HitRatio@50	HitRatio@100	HitRatio@1000	HitRatio@5000
CN	12.5	12.5	12.5	12.5	12.5	50.0
SA	0.0	0.0	0.0	0.0	0.0	0.0
JA	0.0	0.0	0.0	0.0	0.0	0.0
SO	0.0	0.0	0.0	0.0	0.0	0.0
HP	0.0	0.0	0.0	0.0	0.0	0.0
HD	0.0	0.0	0.0	0.0	0.0	0.0
LHN	0.0	0.0	0.0	0.0	0.0	0.0
AA	12.5	12.5	12.5	12.5	37.5	62.5
RA	0.0	0.0	12.5	12.5	37.5	87.5
TC	12.5	12.5	12.5	37.5	62.5	75.0

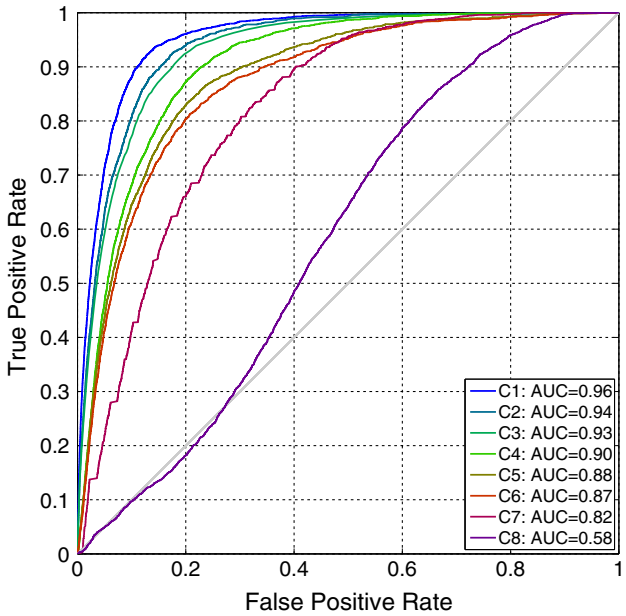

Fig. 11 ROC curves for GooglePlus-based results

GitHub As a first step, we present the prediction results of GitHub-based experiments. The ROC curves are depicted by Fig. 10. The metrics and class correspondence is established in Table 5. We provide the AUC values along with the curves in Fig. 10 and also in Table 5 for easier readability. In GitHub, TC corresponds to C1 and an AUC value of 0.98. Thus, TC outperforms the other methods and delivers the highest accuracy. RA delivers also very good results with an AUC of 0.97. Metrics below 0.5 such as SA, LHN, JA, SO, and HD are not suitable prediction methods for the GitHub-based social networks. The HitRatio at different thresholds is shown in Table 6. The HitRatio until HitRatio@50 is identical for CN, AA and TC. For HitRatio@100 and HitRatio@1000, TC outperforms other methods. RA performs best at HitRatio@5000.

Table 7 AUC Classes for GooglePlus-based results

Class	Definition	AUC
C1	TC	0.96
C2	RA, AA	0.94
C3	CN	0.93
C4	SA	0.90
C5	JA, SO	0.88
C6	HD	0.87
C7	HP	0.82
C8	LHN	0.58

GooglePlus Next, we present the prediction results of GooglePlus-based experiments. The ROC curves are presented in Fig. 11 and the class correspondence is given in Table 7. TC has an AUC value of 0.96 followed by RA and AA with 0.94. Also, the simple CN metric delivers quite good results with an AUC of 0.93. LHN performs worst but still has an AUC above 0.5 thereby delivering acceptable results but with low accuracy.

The HitRatio for GooglePlus-based results is shown in Table 8. TC performs best at all thresholds and delivers the best results compared with the other methods. RA performs slightly better than AA in terms of HitRatio. LHN performed worst in terms of AUC but slightly better than HP with regards to HitRatio. Overall, only TC, RA and AA are suitable methods to perform link prediction in GooglePlus. All other methods have no correct results at HitRatio@30.

Twitter Finally, we present the prediction results of Twitter-based experiments. The ROC curves are presented in Fig. 12 and the class correspondence is given in Table 9. Again, TC performs best with an AUC value of 0.97. Second ranked are again RA and AA with an AUC of 0.96. Note in this context that all three metrics, TC, RA, and AA, give higher weights to those neighbors who have a lower degree (i.e., 'hub-depressed' behavior). However, by considering triad patterns, TC outperforms all other metrics

Table 8 HitRatio (%) for GooglePlus-based results

Metric	HitRatio@10	HitRatio@30	HitRatio@50	HitRatio@100	HitRatio@1000	HitRatio@5000
CN	0.0	0.0	0.1	0.3	2.3	8.1
SA	0.0	0.0	0.0	0.0	0.1	1.4
JA	0.0	0.0	0.0	0.0	0.5	1.7
SO	0.0	0.0	0.0	0.0	0.1	1.5
HP	0.0	0.0	0.0	0.0	0.0	0.0
HD	0.0	0.0	0.0	0.0	0.1	1.7
LHN	0.0	0.0	0.0	0.0	0.0	0.2
AA	0.0	0.1	0.1	0.4	2.4	8.6
RA	0.1	0.2	0.3	0.6	3.4	12.8
TC	0.3	0.7	0.8	1.1	6.0	21.6

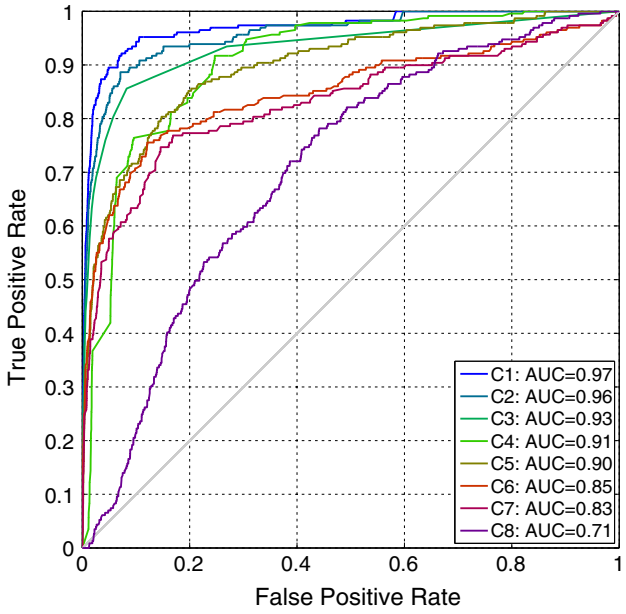

Fig. 12 ROC curves for Twitter-based results

and delivers the most accurate results. Again, LHN ranks last with an AUC of 0.71. None of the metrics have an AUC lower than 0.5. Here, the lowest AUC is 0.71 making all metrics suitable methods for link prediction. As mentioned before, this was not the case for GitHub where many metrics perform below an AUC of 0.5.

The HitRatio for Twitter-based results is shown in Table 10. TC performs best until HitRatio@100. For HitRatio@1000 and HitRatio@5000, RA performs slightly better. However, TC still performs best with respect to AUC and true positive rate. Also, the simple common neighbor (CN) methods provide acceptable results in terms of HitRatio. LHN performs worst with regards to HitRatio (only 5.2 % at HitRatio@5000) and also with regards to AUC.

Table 9 AUC Classes Twitter-based results

Class	Definition	AUC
C1	TC	0.97
C2	RA, AA	0.96
C3	CN	0.93
C4	HP	0.91
C5	SA	0.90
C6	JA, SO	0.85
C7	HD	0.83
C8	LHN	0.71

5.4 Result summary

To provide a brief summary of the main points of our GitHub-, GooglePlus-, and Twitter-based evaluation results and discussions:

- The follower structure of GitHub and GooglePlus or Twitter is very different. Thus, the average degree of GitHub is significantly lower than the average degree in GooglePlus and Twitter.
- As expected, each follower network exhibits distinct triad frequencies. The presented approach helps to give higher weights to triads that are more frequently closed (resulting in closed triangles).
- In general, triadic closeness (TC) outperforms other local similarity-based methods.

The final conclusion of this work with an outlook on future work is given in the following section.

6 Conclusions

The prediction of missing links and the prediction of future links is an important task in the domain of social network analysis. The former helps to infer the ‘real’ social network

Table 10 HitRatio (%) for Twitter-based results

Metric	HitRatio@10	HitRatio@30	HitRatio@50	HitRatio@100	HitRatio@1000	HitRatio@5000
CN	2.6	6.1	9.2	13.1	44.1	72.1
SA	0.0	0.0	1.3	5.2	28.4	56.8
JA	1.7	3.1	3.1	7.9	34.1	56.8
SO	0.0	0.0	1.3	4.4	29.3	54.6
HP	0.0	0.0	0.0	0.0	0.4	40.2
HD	0.9	2.6	2.6	2.6	25.3	49.8
LHN	0.0	0.0	0.0	0.0	0.0	5.2
AA	2.6	6.1	9.6	14.4	48.9	78.6
RA	3.1	6.6	8.3	12.2	56.3	86.5
TC	3.9	8.7	11.8	14.8	55.5	86.4

structure, while the latter is used to give friendship as well as following recommendations to users. A wide range of local, global, and semi-local metrics have been proposed by previous work. A large body of existing literature, however, focuses on undirected networks only. This work closes this gap by focusing on directed networks. Here, we propose triad patterns to predict links between nodes in directed graphs. Our approach is called Triadic Closeness. We designed and implemented a link prediction framework that is able to perform predictions in large-scale social networks. The framework's architecture has been presented and discussed in detail. We performed experiments in three different social networks. First, we analyzed the effectiveness of our proposed approach in GitHub; a social coding community. Second, we obtained a subset of the GooglePlus network and third we performed experiments in a subset of the Twitter follower network. The pattern-based prediction approach delivers the best results among the compared local methods. TC consistently outperformed other approaches. Thus, a pattern-based approach is better suited in directed social networks.

At this stage, we have not considered weighted edges in prediction methods. Edge weights may be obtained through interaction analysis in social network. Weighted networks are subject to our future research. An important aspect will also be the extension of our approach towards global and semi-local methods. As an example, the personalized PageRank method could provide the basis for pattern-aware link prediction. We are currently working on the design of this method.

References

Adamic LA, Adar E (2001) Friends and neighbors on the web. Soc Netw 25:211–230

Aiello LM, Barrat A, Schifanella R, Cattuto C, Markines B, Menczer F (2012) Friendship prediction and homophily in social media. ACM Trans Web 6(2):9:1–9:33

Airoldi EM, Blei DM, Fienberg SE, Xing EP (2008) Mixed membership stochastic blockmodels. J Mach Learn Res 9:1981–2014

Alon U (2007) Network motifs: theory and experimental approaches. Nat Rev Genet 8(6):450–461

Backstrom L, Leskovec J (2011) Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the 4th ACM international conference on Web search and data mining, WSDM '11. ACM, New York, NY, pp 635–644

Batagelj V, Mrvar AA (2001) A subquadratic triad census algorithm for large sparse networks with small maximum degree. Soc Netw 23(3):237–243

Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn 30:1145–1159

Brzozowski MJ, Romero DM (2011) Who should i follow? recommending people in directed social networks. In: Adamic LA, Baeza-Yates RA, Counts S (eds) ICWSM. The AAAI Press, Menlo Park, CA

Clauset A, Moore C, Newman MEJ (2008) Hierarchical structure and the prediction of missing links in networks. Nature 453(7191):98–101

Esslimani I, Brun A, Boyer A (2011) Densifying a behavioral recommender system by social networks link prediction methods. Soc Netw Anal Min 1(3):159–172

Facebook. Online: http://facebook.com (last access 22 Feb 2013)

GitHub. Online: http://github.com (last access 22 Feb 2013)

GitHub. Online: http://developer.github.com/ (last access 22 Feb 2013)

GooglePlus. Online: http://plus.google.com/ (last access 22 Feb 2013)

Granovetter M (1973) The strength of weak ties. Am J Sociol 78(6):1360–1380

Holland PW, Laskey KB, Leinhardt S (1983) Stochastic blockmodels: first steps. Soc Netw 5(2):109–137

Holland PW, Leinhardt S (1970) A method for detecting structure in sociometric data. Am J Sociol 76(3):492–513

Jeh G, Widom J (2002) Simrank: a measure of structural-context similarity. In: Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'02. ACM, New York, NY, pp 538–543

Jeh G, Widom J (2003) Scaling personalized web search. In: Proceedings of the 12th international conference on World Wide Web, WWW '03. ACM, New York, NY, pp 271–279

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

Kwak H, Lee C, Park H, Moon S (2010) What is twitter, a social network or a news media? In: Proceedings of the 19th

international conference on World wide web, WWW '10. ACM, New York, NY, pp 591–600
Leicht EA, Holme P, Newman MEJ (2006) Vertex similarity in networks. Phys Rev E 73:026120
Leskovec J, Huttenlocher D, Kleinberg J (2010) Predicting positive and negative links in online social networks. In: Proceedings of the 19th international conference on World wide web, WWW '10. ACM, New York, NY, pp 641–650
Liben-Nowell D, Kleinberg J (2003) The link prediction problem for social networks. In: Proceedings of the twelfth international conference on information and knowledge management, CIKM '03. ACM, New York, NY, pp 556–559
Liu W, Lu L (2010) Link prediction based on local random walk. Europhys Lett (EPL) 89(5):58007
Lu L, Zhou T (2011) Link prediction in complex networks: a survey. Phys A Stat Mech Appl 390(6):1150–1170
McAuley JJ, Leskovec J (2012) Learning to discover social circles in ego networks. In: Bartlett PL, Pereira FCN, Burges CJC, Bottou L, Weinberger KQ (eds) NIPS. pp 548–556
Meng B, Ke H, Yi T (2011) Link prediction based on a semi-local similarity index. Chin Phys B 20(12):128902
Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: simple building blocks of complex networks. Science 298(5594):824–827
Page L, Brin S, Motwani R, Winograd T (1998) The PageRank citation ranking: bringing order to the web. Technical Report, Stanford University
Ravasz E, Somera AL, Mongru DA, Oltvai ZN, Barabasi AL (2002) Hierarchical organization of modularity in metabolic networks. Science 297(5586):1551–1555
Rettinger A, Wermser H, Huang Y, Tresp V (2012) Context-aware tensor decomposition for relation prediction in social networks. Soc Netw Anal Min 2(4):373–385
Romero DM, Kleinberg JM (2010) The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In: Cohen WW, Gosling S (eds) ICWSM. The AAAI Press, Menlo Park, CA
Salton G, McGill MJ (1986) Introduction to modern Information retrieval. McGraw-Hill, Inc., New York, NY
Sautter G, Bhm K (2013) High-throughput crowdsourcing mechanisms for complex tasks. Soc Netw Anal Min 3(4):873–888
Schall D (2012) Expertise ranking using activity and contextual link measures. Data Knowl Eng 71(1):92–113
Schall D (2012) Service oriented crowdsourcing: architecture, protocols and algorithms. Springer Briefs in Computer Science. Springer, New York, NY
Schall D, Skopik F (2012) Social network mining of requester communities in crowdsourcing markets. Soc Netw Anal Min 2(4):329–344
Snijders TA (2012) Transitivity and Triads. University of Oxford. Online: http://www.stats.ox.ac.uk/snijders/Trans_Triads_ha.pdf (last access 22-Feb-2013)
Sørensen T (1957) A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. Biologiske Skrifter/Kongelige Danske Videnskabernes Selskab 5(4):1–34
Stanford. Online: http://snap.stanford.edu/data/index.html (last access 22 Feb 2013)
Symeonidis P, Mantas N (2013) Spectral clustering for link prediction in social networks with positive and negative links. Soc Netw Anal Min 3(4):1433–1447
Twitter. Online: http://twitter.com (last access 22 Feb 2013)
Wasserman S, Faust K, Iacobucci D (1994) Social network analysis: methods and applications (structural analysis in the social sciences). Cambridge University Press, Cambridge
White HC, Boorman SA, Breiger RL (1976) Social structure from multiple networks. i. blockmodels of roles and positions. Am J Sociol 81(4):730–780
Zhou T, Lu L, Zhang Y-C (2009) Predicting missing links via local information. Eur Phys J B Condens Matter Complex Syst 71(4):623–630