# Centrality Robustness and Link Prediction in Complex Social Networks

Davidsen, Søren Atmakuri; Ortiz-Arroyo, Daniel

*Published in:*
Computational Social Networks

*DOI (link to publication from Publisher):*
10.1007/978-1-4471-4048-1_8

*Publication date:*
2012

*Document Version*
Early version, also known as pre-print

Link to publication from Aalborg University

*Citation for published version (APA):*
Davidsen, S. A., & Ortiz-Arroyo, D. (2012). Centrality Robustness and Link Prediction in Complex Social Networks. In A. Abraham, & A.-E. Hassanien (Eds.), *Computational Social Networks: Tools, Perspectives and Applications* (pp. 197-224). Springer. https://doi.org/10.1007/978-1-4471-4048-1_8

# Chapter 8
# Centrality Robustness and Link Prediction in Complex Social Networks

**Søren Atmakuri Davidsen and Daniel Ortiz-Arroyo**

**Abstract** This chapter addresses two important issues in social network analysis that involve uncertainty. Firstly, we present an analysis on the robustness of centrality measures that extends the work presented in Borgatti et al. using three types of complex network structures and one real social network. Secondly, we present a method to predict edges in dynamic social networks. Our experimental results indicate that the robustness of the centrality measures applied to more realistic social networks follows a predictable pattern and that the use of temporal statistics could improve the accuracy achieved on edge prediction.

## Introduction

Complex networks are networks which are neither random nor regular. This type of networks are found in many diverse areas such as social networks, transportation airlines, biological networks, etc. This chapter focuses on analyzing some of the properties of complex social networks.

Social Network Analysis (SNA) studies social networks with regard to its structure, functionality, and efficiency in diffusing information. SNA provides the link between sociology and graph theory. In SNA, social networks are represented as graphs, depicting the relations and ties among social actors. The connection between social networks and computational studies is laid out in [20]. SNA has recently attracted attention given the popularity of social network sites on the Internet, the recent worldwide epidemic outbursts, financial fraud scandals, and the coordinated actions performed by international criminal organizations. The study of these social networks helps us to understand the dynamics, structuring, and functioning of social relations.

S.A. Davidsen (✉) • D. Ortiz-Arroyo
Computational Intelligence and Security Laboratory, Department of Electronic Systems,
Aalborg University, Esbjerg, Denmark
e-mail: soren@cislab.org,do@cislab.org

A. Abraham and A.-E. Hassanien (eds.), *Computational Social Networks: Tools, Perspectives and Applications*, DOI 10.1007/978-1-4471-4048-1_8,

Finding central nodes is an important task in SNA that helps analysts to understand how information is diffused in the network, how command control is structured, and what is the effect on the network's structure when such nodes are removed. However, determining the full structure of a network may not be feasible in some cases. For instance, criminal organizations try to hide the structure and hierarchy of their networks from outsiders. Additionally, errors may be introduced when the network is constructed. Errors produce uncertainty with regard to the true structure of the network. Two types of uncertainty may be found during network construction: (1) *node uncertainty*, i.e., uncertainty of a node existence and (2) *edge uncertainty*, i.e., uncertainty about the relation between two nodes.

In [5] Borgatti et al. examined the effect that random errors introduced during network construction have on the performance of some centrality measures. Borgatti essentially found that centrality measures are robust in the presence of errors. However, the networks considered in [5] were random Erdős-Rényi networks. In this chapter, we have extended that work to study complex social networks that are not random.

Networks commonly studied in SNA are static, i.e., the models assume that their structure will not change. Contrarily to static networks, dynamic social networks change with time. In dynamic networks, nodes and edges are periodically added or deleted. An example of a dynamic social network is the network of personal relations that is created during the lifetime of a person. The analysis of dynamic social networks is a relatively new area of research in SNA.

Dynamic networks can be modeled using *temporal graphs*. In this chapter, we use Tang et al.'s *temporal network model* [22] and extend it to predict when a new edge is likely to be added to a network. Our experiments show that there is correlation between: (a) the age of a node and the creation of new edges and (b) last edge creation time and the creation of new edges. We have applied this knowledge to the problem of predicting edges in a temporal graph. Our results indicate that when temporal information is available, it can be used to improve prediction accuracy.

Section "Network Models" introduces the network concepts used in this chapter and section "Related Work" describes most relevant related work. In section "Robustness of Centrality Measures in Complex Social Networks," we examine the robustness aspect and present experimental results. Section "Edge Prediction in Temporal Social Networks" examines temporal networks and presents experimental results on the accuracy of our edge prediction method. Finally, in section "Conclusions and Future Work" we conclude our work and discuss future directions.

## Network Models

The most common model of a network is an undirected weighted graph $G(V, E, w)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices, $E = \{e_1, e_2, \ldots, e_m\}$ a set of edges, each edge being a tuple of the two connecting nodes $e_i = (s, t)$, and $w(v, u) \rightarrow [0, 1]$ a function that maps each edge into a weight. The unweighted network is a special case where $\forall (v, u) \in E : w(v, y) = 1$.

While networks in some domains can be described as directed graphs, in this chapter, we will consider only undirected graphs (i.e., for social network analysis, we consider actors to be reciprocally associated). Formally this means $(v, u) \in E \Leftrightarrow (u, v) \in E$.

For simplicity, in the rest of this chapter, we will use indistinctly the terms network and undirected graph, additionally to edge and link to mean the same concepts.

The *order* of a graph is the number of vertices in the graph, $|V|$. The *size* is the maximal number of possible edges in the graph $|E|$. The number of edges is $\frac{|V|(|V|-1)}{2}$. The *density* is the proportion of current edges in a graph to all possible edges that the graph may contain, $\frac{2|E|}{|V|(|V|-1)}$. A graph is *complete* if its density is 1. Low-density graphs are called *sparse*. Nodes are called *adjacent* if there is an edge connecting them.

A graph has a *path* from $s$ (the source node) to $t$ (the target node) if there is a sequence of edges connecting the nodes, $(s, u_0), (u_0, u_1), \ldots (u_{n-1}, t)$ through a finite set of interconnected nodes. The minimum-length path between two nodes $s$ and $t$ is called the *shortest* or *geodesic* path, which we denote $p(s, t)$ and its length $s(s, t)$. A graph is *connected* if all nodes are joined by a path, $\forall v, j \in V : s(v, u) < \infty$. If a graph is not connected, each maximal set of nodes for which a path exists between all of them is denoted a *component*. A *walk* is a path that allows the same edge to be traversed more than once. If a node $v$ has $d$ adjacent nodes, we denote $d$ the *degree* of $v$, $D(v) = d$. A *triad* is a triangle of three nodes connected to each other by direct edges. The neighbors of a node we call the *neighborhood*, $N(v) = \{u|(v, u) \in E\}$.

An adjacency matrix $A^{n \times n}$ is a useful representation of a network in which each element represents an edge (see Eq. 8.1). The adjacency matrix for an undirected graph is symmetric around the diagonal.

$$A_{vu} = \begin{cases} w(v, u), & \text{if } (v, u) \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{8.1}$$

A *temporal graph* is a graph for which there is an ordered sequence of edges $(e_1, e_2, e_3, \ldots, e_t)$, the index of each edge is the edge's time-stamp, hence the $t$th edge we denote $e_t$. $t(v)$ is a function that gives time-stamp when a node $v$ joined the network, and $t(e)$ is the time-stamp when edge $e$ joined the network. We use $T$ to denote the set of time-stamps.

In this chapter, we use Tang et al.'s *temporal network model* [22]. The model describes the temporal network as a sequence of states in which each state contains time-stamps of the events that happen in the network.

The set of observed network states is denoted $T$, where $T = \{t_0, \ldots, t_{\max}\}$. Each state (represented by its time-stamp $t$) produces a temporary graph to which nodes and edges have been added or deleted. See Fig. 8.1 for a visual example.

A function $t(e)$ yields the time-stamp of the state at which an edge was added to the network. The final network $G(V, E)$ is the union of edges from each state $E_0 \cup \ldots \cup E_t$.

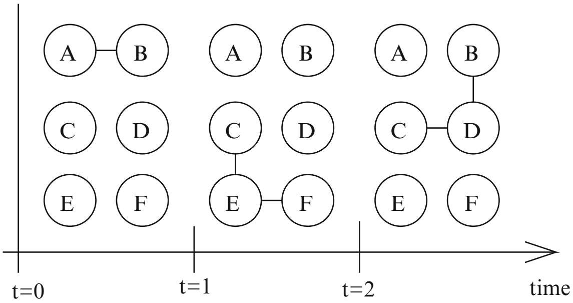


**Fig. 8.1** Prototype visualization of the Tang temporal network model

In Tang's model, all nodes are known a priori, i.e., the model does not consider new nodes entering the network. To model new nodes entering a network, we propose to augment Tang's model with a function $t(n)$ similar to $t(e)$, which will allows us to keep track of the time when a node enters the network. We will also use a subscript $t$ for each of the available functions/sets to indicate that network is at state that has a time-stamp $t$, for example, $s_t(v,u)$ denotes the length of the geodesic between $v$ and $u$ at time $t$.

## Related Work

The study of complex networks is today a well-established field. Early work that defined the *small-world* network effect [17] and the more recent discovery of *scale-free* structures in [1] laid out the groundwork for the study of complex networks. Several recent surveys on complex networks are available, see for instance [4, 10, 18].

Only little work on robustness has been done. Borgatti et al. in [5] generate random graphs to study the performance of centrality measures in the presence of errors. Borgatti classifies errors into two groups: (1) sampling errors and (2) misinformation errors. Sampling errors occur when the observed network is only a partial sample of the true network, i.e., the network misses either nodes or edges. Misinformation errors occur when a network has additional information compared to the real network, i.e., the network may have additional nodes or edges.

The experiments cited in [5] show that as expected errors degrade the performance of the centrality measures employed. However, degradation occurs in a predictable way. Hence, if the proportion of errors in a random network could be estimated, it will be possible to predict the accuracy of the results produced by the centrality measures.

In relation to link prediction, the discovered properties of complex networks lead to proposals of generative models that explain the dynamics of network evolution. For instance, in [23] a generative model for small-world networks was proposed using random rewirings of a lattice. Moreover, in [2] the *preferential attachment* model was suggested to explain how a scale-free structure could emerge.

Many other properties of complex networks have since then been examined. Some of them are *assortative mixing* [18] where types of nodes are taken into account when selecting edges, the *giant component* [4, 18] behavior in a network, *community structures* [18], and more recently a *densifying model* described in [15].

Some approaches include in their generative models the use of dynamic information such as node age or lifetime. Node age was used to predict the clustering coefficient of nodes in [12]. In [7] node lifetime is analyzed to propose a model with decaying degrees. Other research such as those in [8, 9] include using the visualization of changes to calculate network centralities.

The problem of edge prediction has received recently some attention. The problem is described with some detail in [16, 24]. In these works, [16] evaluates current state-of-art methods and [24] creates a classification of the different approaches. The general classes of link prediction considered in [24] are: (a) Class-1, node-wise similarity, where similarities between nodes are determined based on their features, (b) Class-2, topological patterns where similarities between nodes are determined from the structure of the network (locally or globally), and (c) Class-3, probabilistic models, where compressed networks of interactions are learned and used for prediction.

Class-1 has roots in clustering/classification tasks, where we wish to discover hidden relations between a set of otherwise unrelated nodes. Link prediction for this class depends on the availability of a feature vector for each node $f(v) = (f_1, f_2, \ldots, f_n)$, and a similarity measure to determine the vectors' similarity $\text{sim}(f(v), f(u))$. Classical measures such as cosine similarity and the Euclidean distance could be employed. These methods predict a missing edge in the network, when two nodes with high similarity are found. In network analysis, however, this approach is not always applicable, but only in models that include latent spaces, or where physical distance does not matter.

Class-2 employs topological patterns in the network. Some approaches in this class are based on a local heuristic such as "your close neighbors influence you". This is the case described in [16] where a node-proximity measure is developed to find new neighbors and in [6] where a combination of clustering coefficient and hierarchical clustering is used.

Class-3 is a classification task, where examples of network interactions are learned and used for prediction. This approach is described in [11] where a learning algorithm is applied to select neighbors.

While Class-1 depends on node features being available and Class-3 depends on having available examples of node interactions, Class-2 depends only on the network itself. This is a desirable feature, but has some limitations.

## Robustness of Centrality Measures in Complex Social Networks

This section describes a method for evaluating the robustness of the centrality measures described in Table 8.1 when applied to non random social networks.

**Table 8.1** Commonly used centrality measures and their definition

| Measure | Definition | Notes |
|---|---|---|
| Degree centrality | $C_D(v) = \frac{D(v)}{n-1}$ | |
| Betweenness centrality | $C_B(v) = \sum_{u \in V} \sum_{z \in V} g_{uz}(v)$ | $g_{uz}(v) = 1$ if $v \in p(u,z)$ |
| Closeness centrality | $C_C(v) = \sum_{u \in V} s(v,u)$ | |
| Eigenvector centrality | $x_i = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} x_j$ | rewritten as $\lambda x = Ax$ |
| Entropy centrality | $C_E(v) = -\sum_{u \in V'}^{n} \gamma(u) \times \log_2 \gamma(u)$ | $\gamma(v) = \frac{\text{paths}(v_i)}{M}$. $V'$ is $V$ without the node $v$. |

In addition to replicate the results produced by Borgatti in [5] using ER random networks, we want to analyze the robustness of centrality measures on more realistic social networks. One way to do this is by synthesizing network structures that are general models of social networks. Our synthetic networks were selected from three categories described in the literature. Firstly, we generated *scale-free* networks that are used to model collaboration networks using Barabasi's [3] *preferential attachment* model. Secondly, we used the Krebs' community structures described in [14] to generate *Cliques* that model friendship networks and *Core/periphery* networks that can be used to model cohesive social groups.

Section "Synthesizing Complex Networks" describes the methods used to generate the three types of complex networks and section "Experimental Tests" discusses the results obtained in our experiments.

## Synthesizing Complex Networks

Clique networks are networks in which several cliques of tightly connected nodes are interconnected loosely. Clique networks are synthesized using a parametric algorithm which groups the nodes of the network into $c$ cliques and then makes random edge creation dependent on whether two nodes are in the same clique or not as shown in Eq. 8.3.

$$re(p) = \begin{cases} 1, & \text{if rand} < p, \\ 0, & \text{otherwise,} \end{cases} \tag{8.2}$$

$$A_{ij} = \begin{cases} re(p_c), & \text{if } C(v_i) = C(v_j), \\ re(p_b), & \text{otherwise,} \end{cases} \tag{8.3}$$

$C(v)$ is a function that returns the clique of the node $v$. $re(p)$ is a function that returns 1 (indicating an edge in the adjacency matrix) given a uniform random probability distribution $p$. $A$ is the adjacency matrix representing the network. It is expected that centrality measures in clique networks will behave similarly as when applied to ER networks. The intuitive reason for this is that the centrality measures will pick up nodes that are on the edges of many cliques as the most central. Additionally a clique can be seen as a single node in the network.

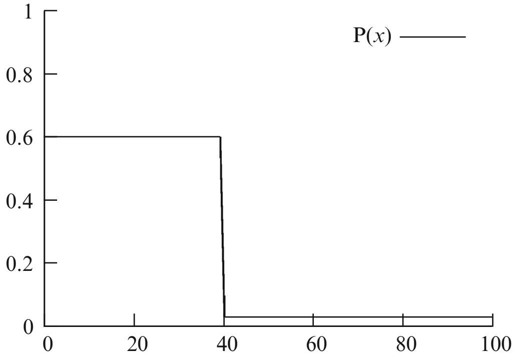


**Fig. 8.2** A sample function for $P(x) = 0.6$ if $x < 40, 0.03$ otherwise

Core/periphery networks are networks in which there is a core of closely connected nodes and a periphery of loosely connected nodes. Core/periphery networks were synthesized as a generalization of ER random networks using a given probability density function $P$ instead of a single probability value $p$. This is shown in Eq. 8.4 where $A$ is an adjacency matrix. A sample function for $P$ can be seen in Fig. 8.2.[1]

$$A_{ij} = \begin{cases} 1, & \text{if rand} < \min(P(i), P(j)), \\ 0, & \text{otherwise,} \end{cases} \tag{8.4}$$

When errors are introduced into core/periphery networks, it is expected that the removal of nodes and edges will have less effect on the centrality measures, i.e., the dense core will provide alternative paths for the periphery nodes. However, when errors are added to the core, they could have some impact on the centrality measures.

A simple method to synthesize scale-free networks is given in [21]. The method consists in attaching new nodes at random to previously existing nodes, using the degree of these previously existing nodes as their probability of attachment. This method is described by Eq. 8.5.

$$Pv_i = \frac{D(v_i)}{D_{\max}} \tag{8.5}$$

$Pv_i$ is the edge creation probability function on node $v_i$, $D$ is the degree of a node $v_i$, and $D_{\max}$ is the highest node degree found in the graph.

Intuitively it is expected that centrality measures will perform better in scale-free networks, given that in its construction few nodes will be very well connected. These few nodes will likely be picked as central nodes by the centrality measures. Hence, the probability of introducing errors in these few nodes is small given the "long tail" of other insignificant nodes. Figure 8.3 shows examples of synthesized networks.

[1] It should be noted that this function is dependent on the size of network that has to be generated.

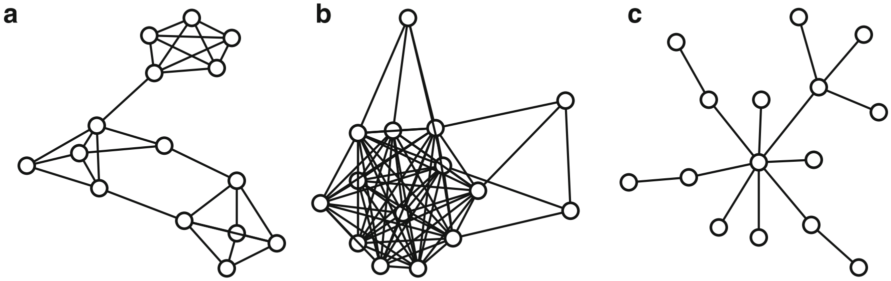


**Fig. 8.3** Examples of small, $n = 15$, synthesized networks. (**a**) Clique network. (**b**) Core/periphery network. (**c**) Scale-free

### *Experimental Tests*

Creating synthetic networks is very convenient for experimental purposes. However, it is also important to use real social networks and see how they perform in the presence of errors when compared to synthetic networks. The terrorist network collected by Krebs [13] was chosen as an example of a real network in our experiments. The network has 62 nodes and 155 edges with a density of 0.08 and is shown as Fig. 4 in [13]. The errors introduced to the network are randomly chosen at each iteration.

We performed three types of experiments to determine: (1) the robustness of centrality measures on random complex networks, (2) the robustness of centrality measures using a real network, and (3) the robustness of entropy as a centrality measure

In [5] five different measures are used to determine the accuracy of the measurements obtained in the network containing errors when compared to an error-free (true) network. These measures are:

1. Top 1 – considers only the top node, i.e., the most central node in the true network is also the most central in the network with errors introduced.
2. Top 3 – considers the top 3 nodes, i.e., the most central node in the true network is among the three most central in the network with errors introduced.
3. Top 10% – considers the top 0.1 of nodes, i.e., the most central node in the true network is in the top 10% of the most central nodes in the network with errors introduced.
4. Overlap – considers Jaccard's similarity measure between top 10% nodes, i.e., the overlap between the top 10% nodes in the true network and the top 10% nodes in the network with errors introduced.
5. $R^2$ – considers Pearson correlation, i.e., the correlation between the centralities of the true network and the centralities of the network with errors introduced.

In this chapter, we will use the same measurements of accuracy.

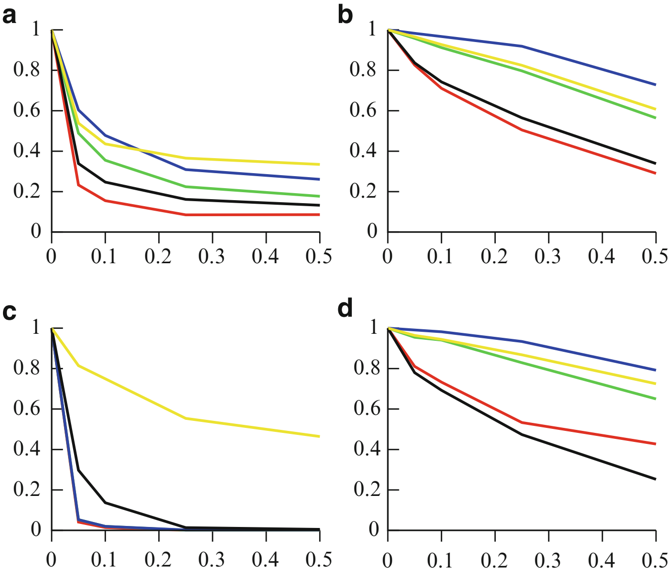


**Fig. 8.4** Scatter plots of the average betweenness accuracy as a function of error. Betweenness measures for core/periphery network of size 50, and a density of 0.40. Legend: 🟥 top 1, 🟩 top 3, 🟦 top 10%, ⬛ overlap, 🟨 $R^2$. (**a**) Edge addition. (**b**) Edge removal. (**c**) Node addition. (**d**) Node removal

## *Experiments with Synthetic Complex Networks*

Synthetic complex graphs were created and tests were conducted with each of the complex topologies previously mentioned. Due to lack of space, we only report the results of our experiments using graphs of size $n = 50$, but we have performed other tests with graph sizes of 10, 25, 50, and 100 nodes, obtaining similar results.

Additionally, in all of our experiments, we used only the betweenness centrality measure. This measure was chosen because it is a global measure and because it was also used in [5]. However, we have conducted experiments using betweenness, closeness, and eigenvector centrality obtaining similar accuracies. It is important to remark that contraily to other measures, degree centrality is easily affected by local changes in the graphs; hence, it is a less predictive measure.

Finally, networks were compared using the same values in density and size. However, in the case of the real network and a random scale-free network, the size and density of the networks used were not the same.

### Core/Periphery Networks

To generate core/periphery networks, the $P$ function shown in Fig. 8.2 was used. Figure 8.4 and Table 8.2 show that the accuracy obtained for this kind of network

**Table 8.2** Arithmetic mean of the difference between measures of an ER network and a core/periphery network, calculated for betweenness centrality, on graphs with 50 nodes, and 0.40 density

| Measure | Edge add | Edge rem | Node add | Node rem |
|---|---|---|---|---|
| Top 1 | −0.309 | 0.093 | −0.398 | 0.069 |
| Top 3 | −0.370 | 0.048 | −0.576 | 0.040 |
| Top 10% | −0.358 | 0.046 | −0.633 | 0.037 |
| Overlap | −0.241 | 0.109 | −0.302 | 0.054 |
| $R^2$ | −0.278 | 0.085 | −0.123 | 0.085 |

has a distinctive difference to that obtained with random ER networks. As expected, in the case of misinformation (edge and node addition), the accuracy drops quickly (having an average within $[-0.633, -0.123]$), while in the case of sampling (edge and node removal), the accuracy is slightly better (within $[0.037, 0.109]$).

### Clique Networks

To generate clique networks, the required parameters were set to the following values: $s = 0.28$, $p_c = 0.60$, and $p_b = 0.03$. This means that tests were conducted on networks with three to four cliques. The values for $p_c$ and $p_b$ were found experimentally to produce easily visible clique networks. The density of graphs generated with these settings was found to be 0.20 for comparison with ER networks of equal density.

As expected Fig. 8.5 and Table 8.3 show that accuracy is similar to that obtained with random ER networks. These figures show that adding extra nodes or edges has a negative impact on the accuracy, while in the case of sampling accuracy improves. However, this variation is small compared to the case of core/periphery networks. The reason for this could be that an odd size of cliques was used, which could bias one clique to be more central than others.

### Scale-Free Networks

To generate scale-free networks no special parameters are needed. The density of the networks is shown in Eq. 8.6.

$$\text{density}(n) = \frac{2(n-1)}{n(n-1)} \tag{8.6}$$

Previous figures show that scale-free networks are very robust to errors. According to Barabási and Bonabeau [3], we should expect that scale-free networks be more robust than random networks, and the experimental results in Fig. 8.6 and Table 8.4 support this assumption. This indicates that centrality measures applied to real networks that have scale-free structure will perform well in the presence of errors.

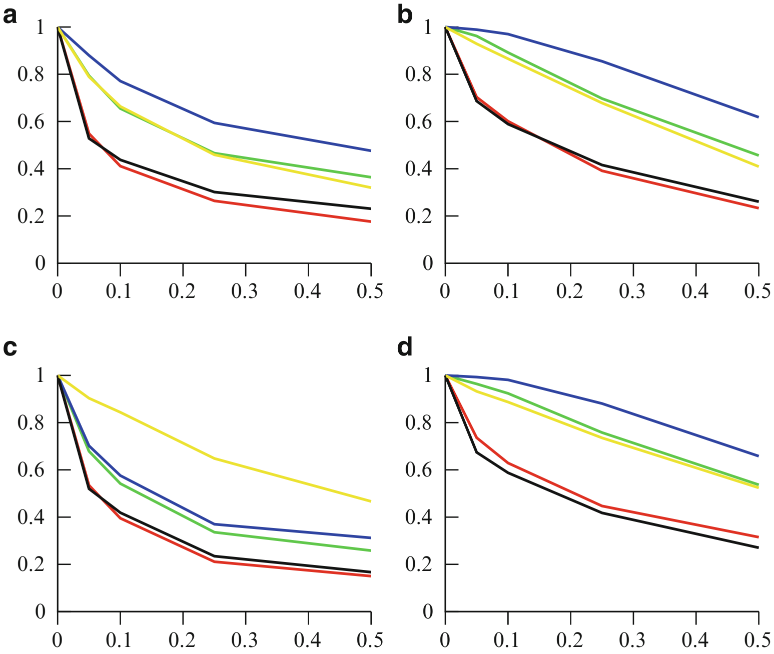


**Fig. 8.5** Scatter plots of the average betweenness accuracy as a function of error. Betweenness measures for clique network of size 50, a density of 0.20. Legend: 🟥 top 1, 🟩 top 3, 🟦 top 10%, ⬛ overlap, 🟨 $R^2$. (**a**) Edge addition. (**b**) Edge removal. (**c**) Node addition. (**d**) Node removal

**Table 8.3** Arithmetic mean of the difference between measures of an ER network and a clique network, calculated for betweenness centrality, on graphs with 50 nodes, and 0.20 density

| Measure | Edge add | Edge rem | Node add | Node rem |
|---|---|---|---|---|
| Top 1 | −0.092 | 0.043 | −0.145 | −0.021 |
| Top 3 | −0.116 | 0.032 | −0.235 | 0.006 |
| Top 10% | −0.104 | 0.033 | −0.268 | 0.002 |
| Overlap | −0.082 | 0.026 | −0.122 | 0.003 |
| $R^2$ | −0.132 | 0.024 | −0.071 | −0.001 |

### *Krebs Terrorist Network*

As an example of a real-world network, we have used the Krebs Terrorist Network. As with the synthesized networks, we have introduced random errors in this network, and compared with random errors in an ER network of same size and density.

As expected Fig. 8.7 shows that the accuracy obtained in this network is better compared to that obtained with the ER random network in Fig. 8.8. Since it is difficult to generate a scale-free network with equal size and density as the Krebs network, the comparison with a scale-free network was made using the results presented in Fig. 8.6. Table 8.5 shows that the accuracies are very similar, within $[-0.103, 0.078]$.

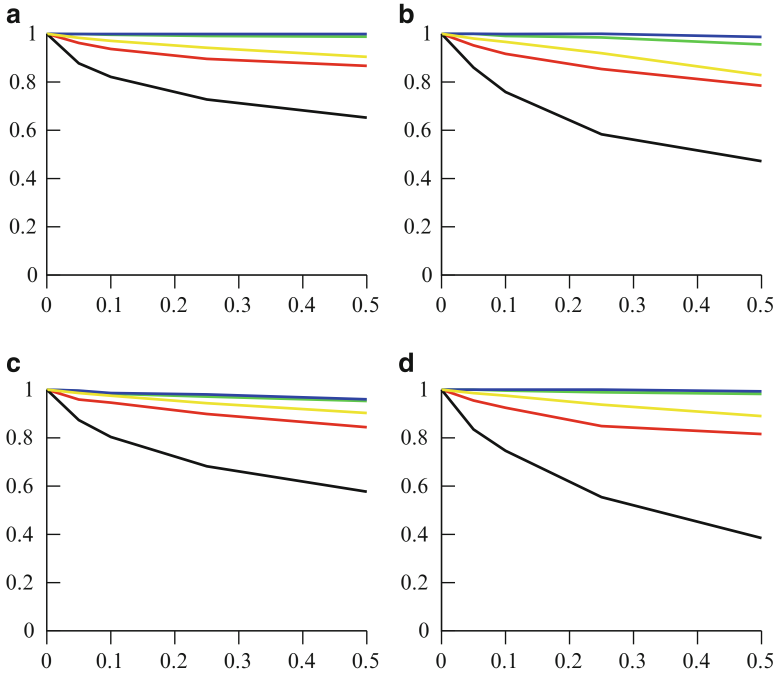


**Fig. 8.6** Scatter plots of the average betweenness accuracy as a function of error. Betweenness measures for scale-free network of size 50, with a density of 0.04. Legend: 🟥 Top 1, 🟩 Top 3, 🟦 Top 10%, ⬛ Overlap, 🟨 $R^2$. (**a**) Edge addition. (**b**) Edge removal. (**c**) Node addition. (**d**) Node removal

**Table 8.4** Arithmetic mean of the difference between measures of an ER network and a scale-free network, calculated for betweenness centrality on graphs with 50 nodes, and 0.04 density

| Measure | Edge add | Edge rem | Node add | Node rem |
|---|---|---|---|---|
| Top 1 | 0.355 | 0.279 | 0.309 | 0.189 |
| Top 3 | 0.251 | 0.224 | 0.227 | 0.099 |
| Top 10% | 0.222 | 0.202 | 0.212 | 0.091 |
| Overlap | 0.295 | 0.235 | 0.267 | 0.148 |
| $R^2$ | 0.217 | 0.243 | 0.196 | 0.161 |

## *Entropy as Centrality Measure*

Our experiments using the entropy measure described in [19] were conducted using the following parameters: (1) network sizes $\in \{10, 25, 50\}$, (2) network densities $\in \{0.05, 0.10, 0.25, 0.50\}$, and (3) error ratios $\in \{0.00, 0.05, 0.10, 0.25, 0.50\}$.

For each combination, 1,000 ER networks were generated and statistics of the entropy measure were collected. Figure 8.9 shows the results from these tests. The experiment shows that the accuracy of entropy under errors declines faster than the other centrality measures, however it still does so in a predictive way.

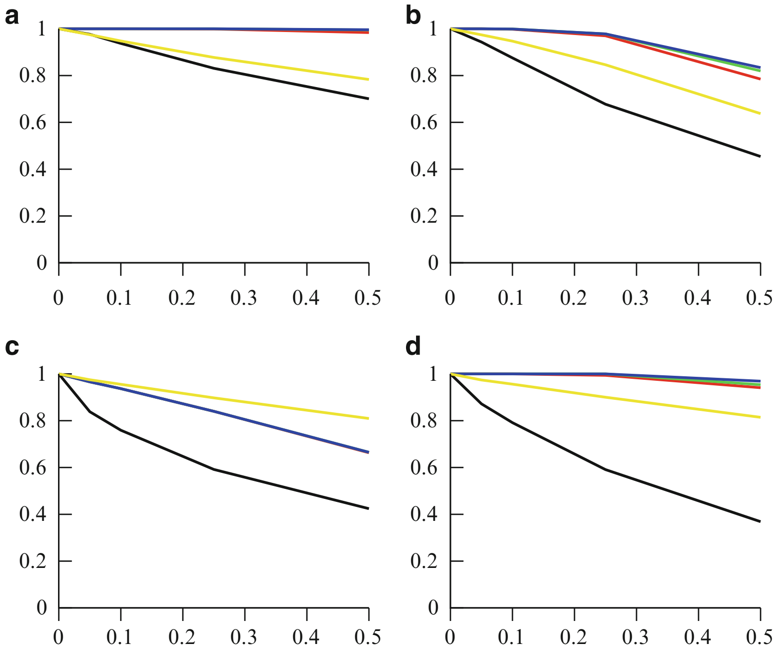


**Fig. 8.7** Scatter plots of the average betweenness accuracy as a function of error. Betweenness measures for Krebs Terrorist network of size 62, with a density of 0.08. Legend: 🟥 top 1, 🟩 top 3, 🟦 top 10%, ⬛ overlap, 🟨 $R^2$. (**a**) Edge addition. (**b**) Edge removal. (**c**) Node addition. (**d**) Node removal

Figure 8.9 shows the performance of the entropy measure when applied to an ER graph with a density of 0.05. Figure 8.10 shows that dense graphs produce less variation in the entropy. At densities larger than 0.05, the variation in centrality entropy becomes so small that results of the measure are inconclusive in deciding the most central node. Variation is calculated as the number of different values of $C_E(v)$ (see Fig. 8.1) found in the test networks.

## Edge Prediction in Temporal Social Networks

The edge prediction problem can be formally describe as: Given a network $G(V, E)$, where $E$ represents the observed edges, how likely is that an unobserved edge $(v, u) \notin E$ may appear between an arbitrary pair of nodes $(v, u)$.

Section "Temporal Datasets" describes the datasets used in our edge prediction experiments. Sections "Static Network Properties" and "Exploring the Dynamics of the Datasets" describe the static and dynamic parameters that we have used to improve prediction accuracy. Finally, section "Temporal Modeling and Experimental Results" describes the results of our experiments.

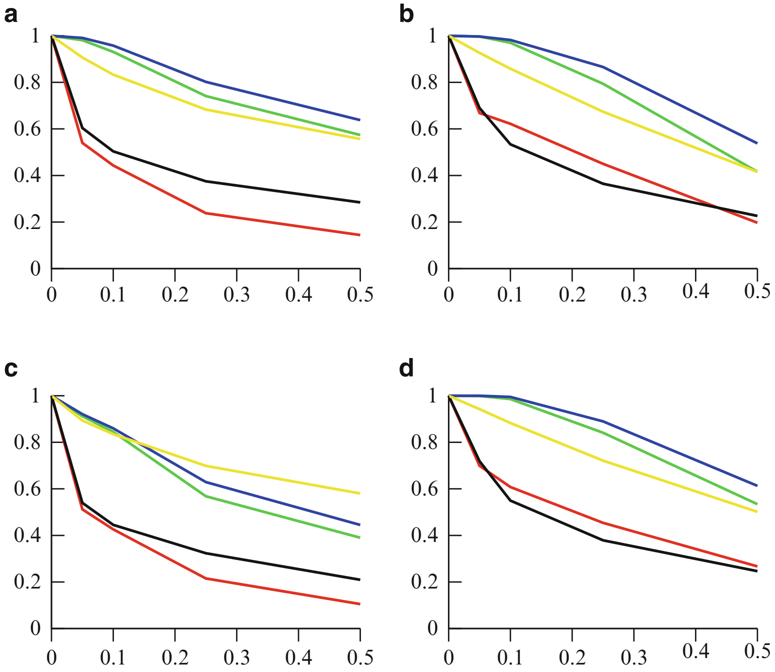


**Fig. 8.8** Scatter plots of the average betweenness accuracy as a function of error. Betweenness measures for ER network of size 62, with a density of 0.08. Legend: 🟥 top 1, 🟩 top 3, 🟦 top 10%, ⬛ overlap, 🟨 $R^2$. (**a**) Edge addition. (**b**) Edge removal. (**c**) Node addition. (**d**) Node removal

**Table 8.5** Arithmetic mean of the difference between measures of a scale-free network with 50 nodes, and a density of 0.04, and the Krebs network, calculated for betweenness centrality

| Measure | Edge add | Edge rem | Node add | Node rem |
|---|---|---|---|---|
| Top 1 | 0.064 | 0.049 | −0.048 | 0.078 |
| Top 3 | 0.004 | −0.027 | −0.099 | −0.003 |
| Top 10% | 0.000 | −0.035 | −0.103 | −0.005 |
| Overlap | 0.073 | 0.055 | −0.065 | 0.020 |
| $R^2$ | −0.043 | −0.058 | −0.034 | −0.029 |

## *Temporal Datasets*

We have selected four datasets for our experiments using the following criteria: (1) temporal information must be available, i.e., $t(e)$ must be known, (2) the dataset must contain a statistically significant amount of nodes and edges, and (3) the dataset should be computationally possible to handle.[2]

Given that the number of datasets publicly available that fit these criteria were quite limited, we decided to create our own datasets. The dataset that we called

[2] Practical memory limit for holding adjacency matrices in our computing system is less than 25,000 nodes.

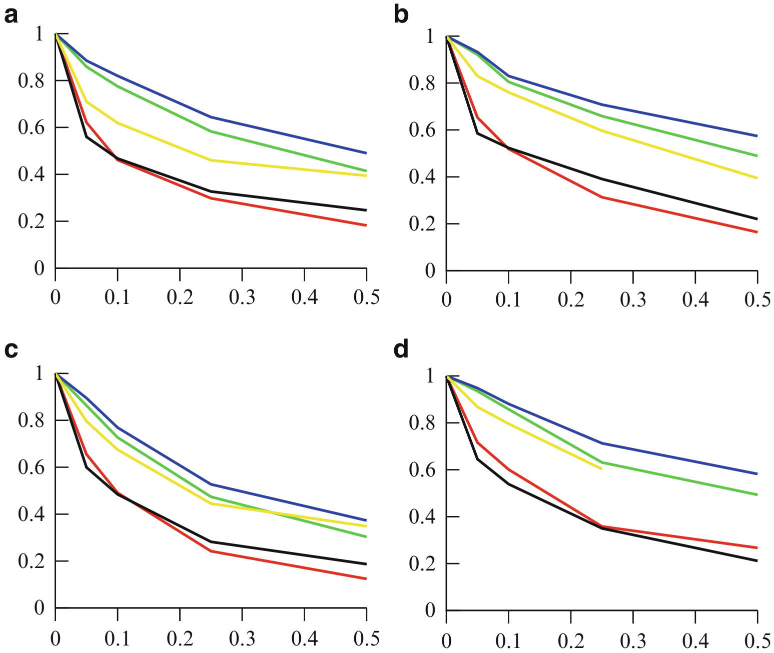
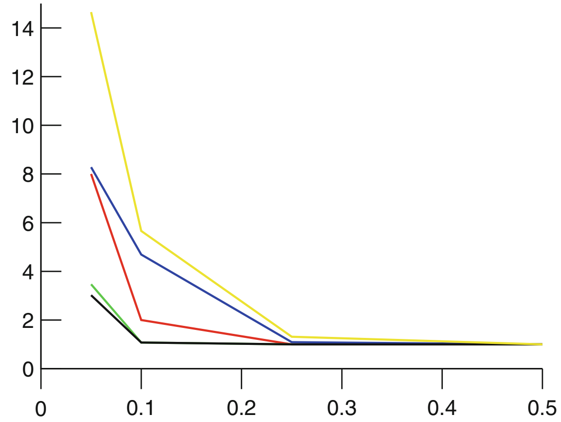


**Fig. 8.9** Scatter plots of the average entropy accuracy as a function of error. Entropy measures for ER network of size 50, with a density of 0.02. Legend: 🟥 top 1, 🟩 top 3, 🟦 top 10%, ⬛ overlap, 🟨 $R^2$. (**a**) Edge addition. (**b**) Edge removal. (**c**) Node addition. (**d**) Node removal

**Fig. 8.10** Scatter plots of the entropy variation ($y$-axis) as a function of density ($x$-axis) in ER networks of size 50. Legend: 🟥 true, 🟩 node add, 🟦 node remove, ⬛ edge add, 🟨 edge remove



*Version2* was obtained from an online IT-news community. The *ENRON* dataset was constructed from the corpus of emails that were made publicly available on the ENRON scandal case. The *HepPh* dataset is a citation graph of papers from the High Energy Physics domain. Finally, we created a synthetic dataset called *Barabasi*

**Table 8.6** Statistics of datasets, $n$ (number of nodes) and $m$ (number of edges) are values from the final graph

| Dataset | $n$ | $m$ | Start | End | $T$ |
|---|---|---|---|---|---|
| ENRON | 19,211 | 45,967 | 1999-01 | 2002-06 | 1,264 |
| Version2 | 3,390 | 110,147 | 2007-01 | 2010-02 | 1,145 |
| HepPh | 21,627 | 201,259 | 1992-03 | 1999-12 | 2,847 |
| Barabasi | 19,218 | 30,446 | 2010-03 | 2012-05 | 801 |

generated by the scale-free preferential attachment model. Basic statistics obtained from these datasets are shown in Table 8.6. In that table, $T$ is the number of days represented in the dataset.

In the next section, we analyze some network properties that could be used for edge prediction. Firstly, we look at properties obtained from static networks[3] that will be used as a baseline for evaluation. Secondly, we examine properties which are based on temporal information.

## *Static Network Properties*

The static properties that we are interested in are: (1) *Small-world effect* and (2) *Scale-free effect*. The small-world effect states that clustering increases as mean path length decreases. This fact allows us to make edge predictions based on triads and hops in the network. The scale-free effect states that nodes prefer high-degree targets when edges are created. This fact allows us to use node degree as an input to our predictor.

### Small-World Effect

The small-world effect [23] appears when the global clustering coefficient is similar to that of a regular network and the mean geodesic is similar to that of a random network. This means that edges will be created most likely between nodes that have a shorter distance between them and that closing triads will appear very frequently, based on the intuitive principle "friends of my friends are also my friends".

To model this behavior, we propose to use a probability distribution based on the number of shared friends (possible triads that we could close) and the likeliness that two nodes will create an edge through one of the shared friends. This probability distribution is depicted in Fig. 8.11. Note that we have also included in these plots a fitness function graph.

The probability distribution used to generate Fig. 8.11 is shown in Eq. 8.8 where $E_\Delta(x)$ is the set of edges that at time $t-1$ have $x$ neighbors in common, $n_t(u,v)$ is

[3] This approach was used in [16].

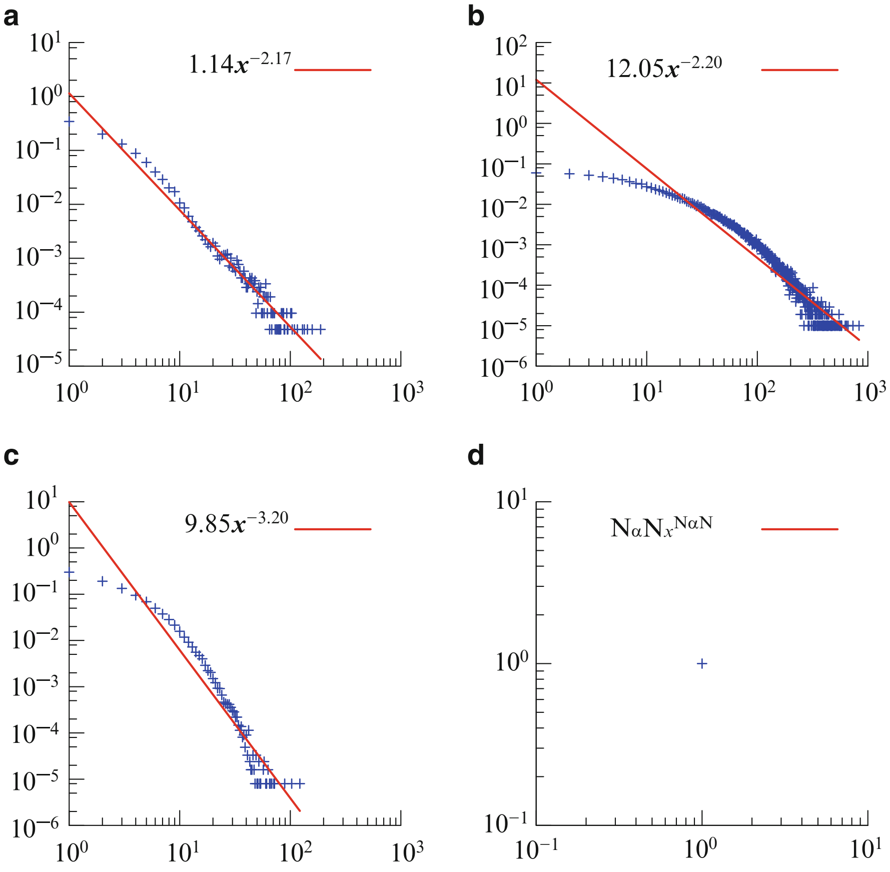


**Fig. 8.11** Probability distribution: Closed triads in proportion to number of shared neighbors, $x$-axis shows the number of shared neighbors, $y$-axis shows the probability that end nodes create an edge. Note: The scale-free dataset was created with only one edge per new node, hence no triads are closed. (**a**) ENRON. (**b**) Version2. (**c**) HepPh. (**d**) Scale-free

the number of common neighbors for two nodes $v$ and $u$ and $P_{\Delta}(x)$ is the probability that two nodes with $x$ neighbors in common will create a new edge.

$$n_t(u,v) = |N_t(u) \cap N_t(v)|, \tag{8.7}$$

$$E_{\Delta}(x) = \{e_t = (u,v)|n_{t-1}(u,v) = x\}, \quad P_{\Delta}(x) = \frac{|E_{\Delta}(x)|}{\sum_i |E_{\Delta}(i)|} \tag{8.8}$$

A second probability we are interested in represents the frequency with which a node makes contact with nodes outside the "friend of friends" network. Figure 8.12 shows the probability distribution of the frequency with which relations to far-away nodes in the network are created. We have plotted the probability of linking vs.

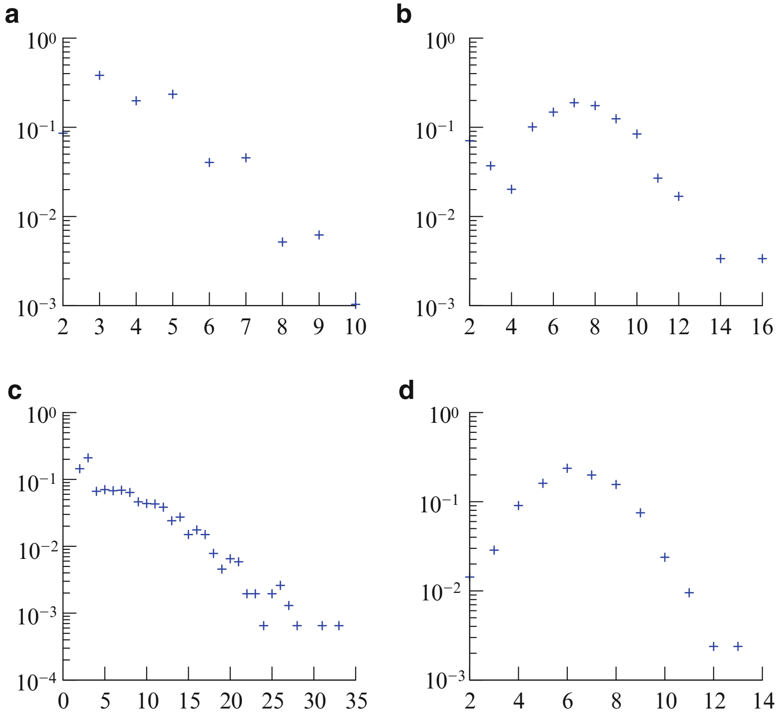


**Fig. 8.12** Probability distribution: Network distance between two nodes before a direct edge is created. $x$-axis shows distance, $y$-axis shows the probability of creating new edge at this distance. (**a**) ENRON. (**b**) Version2. (**c**) HepPh. (**d**) Barabasi

distance between nodes. In Eq. 8.9, $E_h(x)$ represents a set of edges $e_t$, which at time $t-1$ have distance $x$ (represented by $s_{t-1}(v,u)=x$). $P_h(x)$ is the probability of creating an edge between two nodes at distance $x$ and $|E|$ is the total number of edges in the graph.

$$E_h(x) = \{e_t = (v,u) \quad | \quad s_{t-1}(v,u) = x\}, \quad P_h(x) = \frac{|E_h(x)|}{|E|} \tag{8.9}$$

It can be noticed from previous figures that the ENRON and HepPh datasets both follow an exponential decaying function, and the Version2 and Barabasi follow something that appears to be a bell-curve function. The Version2 dataset shows the most surprising behavior. In the preferential attachment Barabasi dataset, we would expect a bell-curve distribution, since selecting target nodes happens randomly uniformly. However, in the Version2 dataset, it appears that selecting a target also happens randomly uniformly in the network.

### Scale-Free Effect

The scale-free effect means that the degree distribution follows a power-law function $f(x) = ax^-k$, which is true for our datasets with $k = [-0.99, 2.01]$. However, what we are interested in is the probability that will indicate how likely a node $v$ is to be attached to another node with a given degree. This probability is calculated in Eq. 8.12. $P_d(x)$ is the probability that a target node with a degree $x$ will create a new edge. $E_d(x,t)$ is the set of edges created where the target node has degree $x$ at time $t$; $V_d(x,t)$ is the set of nodes available with a degree of $x$ at time $t$; and $M_d(x)$ is the number of edges created where the target node has degree $x$ in proportion to how many nodes are available in the network with a degree $x$. Finally, $M_D$ is the sum of all $M_d(x)$ for all $x$.

$$E_d(x,t) = \{e_t = (u,v) | D_{t-1}(v) = x\}, \tag{8.10}$$

$$V_d(x,t) = \{v_t | D_t(v) = x\}, \tag{8.11}$$

$$M_d(x) = \frac{\sum_t |E_d(x,t)|}{\sum_t |V_d(x,t)|}, \quad P_d(x) = \frac{M_d(x)}{M_D} \tag{8.12}$$

We observe from Fig. 8.13 that all datasets show heavy bias toward creating edges in high-degree nodes. For the ENRON and Version2 datasets, our intuition is that few people tend to communicate very frequently, while the HepPh dataset has few seminal papers. The Barabasi dataset shows the expected behavior given by its definition.

## *Exploring the Dynamics of the Datasets*

In [18] and other related works, the dynamics of static networks are thoroughly examined. Properties such as scale-free preferential attachment, small-world clustering, and density are learned and generative models are proposed to create synthetic networks with similar properties. However, little work has been reported in the literature regarding the dynamics of temporal networks. In this section, an analysis was performed in relation to the temporal aspect of the networks. More specifically, we have explored these aspects:

1. *At what rate are new edges created?* In our temporal network model, a node joins the network, and creates new edges over time. Does this happen in uniformly over time?
2. *Is there a lifetime of nodes?* As a node joins the network and creates edges, it is also deleted at some point. The creation of new edges is uniformly distributed over time.

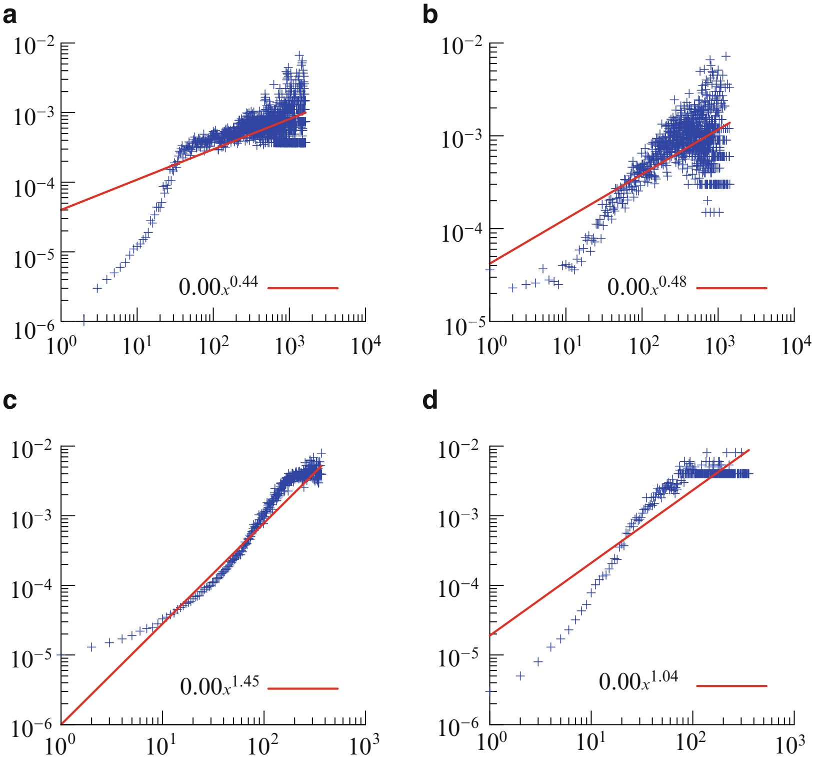


**Fig. 8.13** Probability distribution: Degree attachment. $x$-axis is the degree $d$ and the $y$-axis the probability of edge creation for the degree. (**a**) ENRON. (**b**) Version2. (**c**) HepPh. (**d**) Scale-free

3. *Do older nodes create more edges?* The preferential attachment model indicates that old nodes are the ones with the highest degree, and therefore are likely to create most edges over time.

### Last Edge Creation Time

Last edge creation time is the time that has elapsed from the time a node has created an edge, until the node is connected to a new edge. The intuition used here is that this property will follow the preferential attachment model where nodes with many edges will tend to create new edges more often, and nodes with less edges will wait longer before creating new edges. Figure 8.14 shows the probability distribution described by Eq. 8.14. In Eq. 8.14, $g$ is the time gap since the last edge was created, counted as the number of edges at each $t$, which had a gap $g$ at $t - 1$.

In Eq. 8.14, $g$ is the time gap since the last edge was created, $E_g(x)$ is the set of edges that had a time gap $x$ at $t - 1$, and $E_G$ the set of edges with all different time gaps at $t - 1$.

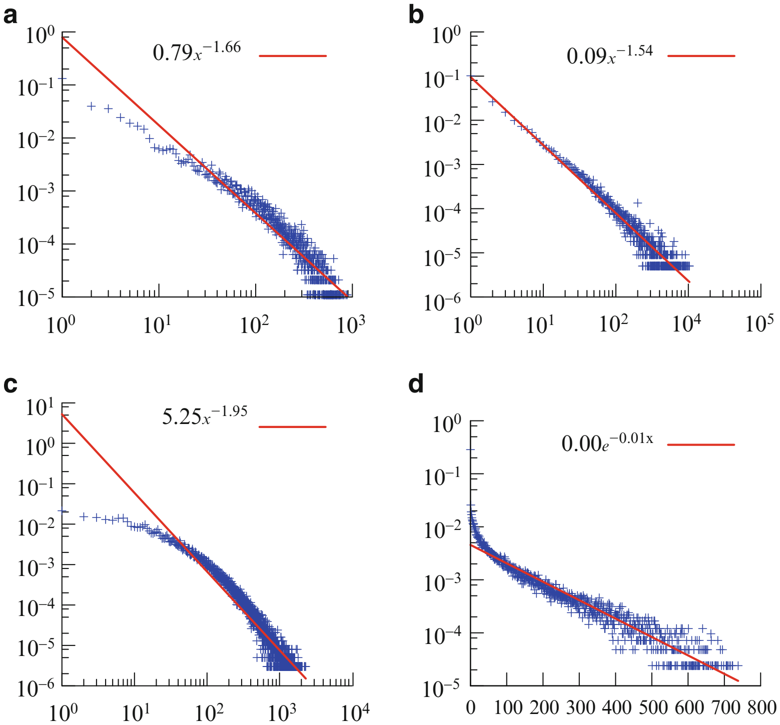


**Fig. 8.14** Probability distribution: Gap between edges, $x$-axis is the gap in $t$, increasing from left to right, $y$-axis is the number of edges with the gap, as described by $P_g(x)$. (**a**) ENRON. (**b**) Version2. (**c**) HepPh. (**d**) Scale-free (note, single-log scale)

$$g_t(v) = t - \max\{t(e)|e = (v, u)\}, \tag{8.13}$$

$$E_g(x) = \{e_t = (u, v)|g_{t-1}(u) = x\}, \quad P_g(x) = \frac{|E_g(x)|}{|E_G|} \tag{8.14}$$

We can see that all datasets, except the synthetic Barabasi dataset, conform to a power-law distribution, where nodes that create edges often will likely continue doing it and the longer time elapses without new edges are being attached to a node, the less likely this will happen. This is similar to the way degree distribution behaves. Therefore, it is not expected that using this effect will give much better predictions when degree distribution has been already used.

## Node Age Edge Creation

The preferential attachment model relies on node degree calcualtions when deciding when to create new edges. Here we examine if there is also a temporal aspect in this

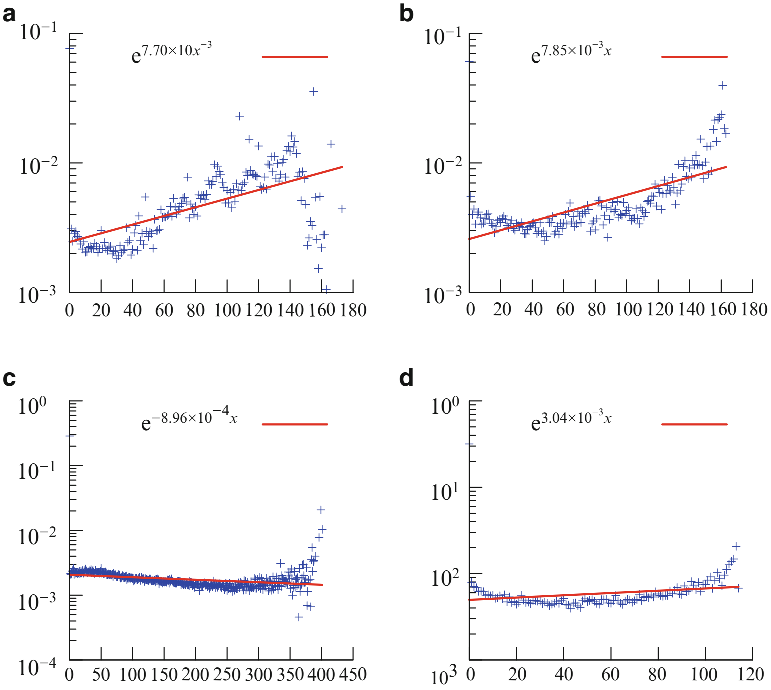


**Fig. 8.15** Probability distribution: Mean edges created in proportion to node age, $x$-axis is the node age $a$ (counting weeks), $y$-axis is the mean number of edges $P_a(x)$. (**a**) ENRON. (**b**) Version2. (**c**) HepPh. (**d**) Barabasi

type of attachment. The intuitive idea is that old nodes have more "experience" within the network and therefore they are more likely to build new "relations."

We propose calculating the mean number of edges created in proportion to a node's age, as shown in Eq. 8.15. The numerator is the number of edges created with age $a$.[4] This is then normalized with the number of nodes that have reached all ages $E_A$. Figure 8.15 shows the plots obtained with each of the datasets. Note that $a$ is counted in weeks for this plot, due to the fact that only few very old nodes are available.

$$E_a(x) = \{e = (u, v) | t(e) - t(u) = x\}, \quad P_a(x) = \frac{|E_a(x)|}{|E_A|} \tag{8.15}$$

Figure 8.15 shows that we can fit the data of all datasets with an exponential function with different features. ENRON and Version2 datasets have a slightly increasing curve. The Version2 dataset is consistently increasing. This means that

[4] $t(e)$ is the time-stamp when edge $e$ was created, and $t(u)$ the time-stamp when $u$ joined the network.

older nodes are more likely to create new edges. Contrarily, the ENRON dataset is generally increasing, except in the oldest nodes. This means that the older a node gets, the more likely it will create new edges, except, if the node is among the oldest. The HepPh dataset, on the other hand, shows a slightly decreasing behavior, but with strong bias to oldest nodes. The intuition for this behavior is that in citation networks, a few old seminal papers will be referenced very often, while in general other papers will be forgotten over time (and thus receive less new edges).

The Barabasi dataset shows an almost straight line, which is what we would expect since the model depends only on the degree.

In summary, all the previous graphs show that we can create either linear, power-law, or exponential functions for the following probability distributions:

- $P_{\Delta}(x)$ probability of creating an edge given $x$ possible triads between two nodes, where $x$ for two nodes is calculated as shown in Eq. 8.7
- $P_h(x)$ probability of creating an edge given the distance $x$ between two nodes, where $x$ for two nodes is $s_t(u, v)$
- $P_d(x)$ probability of creating an edge on a given target node's degree, where $x$ for a target node is $D_t(v)$
- $P_g(x)$ probability of creating an edge given the target node's last edge creation $x$, where $x$ is the gap $g_t(v)$ as defined in Eq. 8.13
- $P_a(x)$ probability of creating an edge on a given target node's age $x$, where $x$ is the age of the target $a_t(u) = t - t(v)$

## Temporal Modeling and Experimental Results

In this section, we propose a simple linear regresion model for edge prediction that uses the statistical features obtained from the analysis we have made on the static network and the temporal network discussed in sections "Static Network Properties" and "Exploring the Dynamics of the Datasets", respectively.

The model employs the scoring function shown in Eq. 8.16, where $\mathbb{R}$ is a score that indicates our degree of belief in an unobserved edge $(u, v)$; a higher number indicates higher belief.

$$\text{score}(G_t, v, u) \rightarrow \mathbb{R}. \tag{8.16}$$

Each unobserved edge is described with a triplet $(v, u, w)$, where $w$ is the score and $u$ and $v$ are nodes. The set of predicted edges $E_p$ is then constructed by selecting edges with highest scores.

### Temporal Link Predictor

This section introduces our novel *temporal link predictor* (tep). The main idea of our predictor is to extract a ranking function from each of the probability distributions used and then aggregate the ranking functions to produce a final score.

**Table 8.7** Probability distributions and their fit ranking functions

| | | Actual used ranking function, $\mu(x)$ | |
|---|---|---|---|
| Probability distribution | Ranking function | ENRON | Version2 |
| Triads attachment, $P_\Delta(x)$ | $\mu_\Delta$ | $x^{-2.17}$ | $x^{-2.20}$ |
| Distance attachment, $P_h(x)$ | $\mu_h$ | $e^{-0.65x}$ | 1 |
| Degree attachment, $P_d(x)$ | $\mu_d$ | $x^{0.44}$ | $x^{0.48}$ |
| Edge creation gap $P_g(x)$ | $\mu_g$ | $x^{-1.66}$ | $x^{-1.54}$ |
| Node age activity $P_a(x)$ | $\mu_a$ | $e^{0.01x}$ | $e^{0.01x}$ |

Note: distance attachment ranking function for Version2 is fixed to 1 because of bell-curve shape

**Table 8.8** Scoring functions

| Name | Score-function | Comments |
|---|---|---|
| $\text{tep}_{\text{am}}$ | $\oplus_{\text{am}}(\mu_d, \mu_\Delta, \mu_a, \mu_h, \mu_g)$ | |
| $\text{tep}_{\text{rd}}$ | $\oplus_{\text{am}}(\mu_d + \frac{1}{\text{rand}(n)}, \mu_\Delta, \mu_h, \mu_g)$ | Use randomness and degree in target node selection. |
| $\text{tep}_{\text{a}}$ | $\oplus_{\text{am}}(\mu_a, \mu_\Delta, \mu_h, \mu_g)$ | Use only age instead of degree. |
| $\text{tep}_{\text{ad}}$ | $\oplus_{\text{am}}(\mu_a + \mu_d, \mu_\Delta, \mu_h, \mu_g)$ | Use age or degree. |
| $\text{tep}_{\text{ngr}}$ | $\oplus_{\text{am}}(\mu_a, \mu_d, \mu_\Delta, \mu_h, \frac{1}{\text{rand}(t)})$ | Use randomness instead of last edge creation. |
| $\text{tep}_{\text{tg}}$ | $\oplus_{\text{am}}(\mu_a, \mu_d, \mu_\Delta + \mu_g, \mu_h)$ | Bias towards triads or last edge creation. |
| Adamic/Adar$_2$ | $\sum_{z\in N(v)\cap N(u)} \frac{1}{\log D_t(z)} \mu_a \mu_g$ | Added temporal for target node. |
| Adamic/Adar$_3$ | $\sum_{z\in N(v)\cap N(u)} \frac{1}{\log(D_t(z)\mu_a(a_t(z))\mu_g(g_t(z)))}$ | Added temporal for intermediate node $z$. |

Firstly, for each of the probability distributions that we have used, we extract a linear ranking function, denoted $\mu$.[5] Table 8.7 shows the relation between the probability distributions and the ranking functions for each of the datasets we used for prediction.

Secondly, we aggregated the ranking functions into a set of scoring functions. Table 8.8 lists our proposed functions. $\oplus_{\text{am}}$ is an aggregation function based on the arithmetic mean and rand$(x)$ is a random number generator with a uniform probability distribution. The set contains variations, where special attention was placed to separate features or randomize a feature. Additionally our method makes the following two modifications to the Adamic/Adar metric: (1) we apply temporal information to the intermediate nodes, (2) we apply temporal information to the target node only.

The main motivation of this simple model is to determine if temporal information helps to increase overall predictor's performance.

[5] This function was found using standard linear least squares method, and the pearson correlation as fitness function between power fit and exponential fit.

### Other Predictors

In our experiments, additionally to the temporal link predictor, we use two other models for comparison purposes: (1) a random predictor used as baseline and (2) the Adamic/Adar measure, which received the best scores in [16].

The Adamic/Adar measure is shown in Eq. 8.17. This equation takes into account the common neighbors of two nodes $(v, u)$ and the degree of the common neighbors, weighted by an importance degree function $\frac{1}{\log D(z)}$. We note that $D(z) \geq 2$ and that nodes with few neighbors are favored. $N(v)$ is the set of neighbors for $v$ and $D(z)$ the degree of node $z$ as previously defined.

$$\text{score}_{\text{AA}}(v, u) = \sum_{z \in N(v) \cap N(u)} \frac{1}{\log D(z)} \tag{8.17}$$

In [16] the Adamic/Adar measure performed the best on the chosen datasets, when only graph-structural properties were considered.

### Experimental Results

We evaluated the proposed temporal link predictor on our two datasets. However, since our simple linear regression model was not always able to fit the actual network data, it is expected that for some datasets, its accuraccy will be low.

We consider predicting edges a function $f(G_{t-1}, l) \rightarrow \{(u, v) : \text{to appear in } G_t\}$, where $l$ is the number of new edges to be predicted for $G_t$ and $G_{t-1}$ is the graph before time $t$. The experiments were performed as follows: for each $t$ in $T$, a network $G_{t-1}$ was constructed, then the set of new edges $E_t$ that will be added to $G_t$ was determined. Then, $a$ the size of the set $E_t$ is used as the number of new edges that will be predicted. $E_a$ will represent the set of new edges actually created between $G_{t-1}$ and $G_t$.

Table 8.9 shows the output of our evaluation on edge prediction using the ENRON and Version2 datasets.

Using the *tep* predictor, each feature was disabled one by one, in order to evaluate its final influence on the final results. These results are shown in Table 8.9 as tepnofeat where feat is the name of disabled feature. A lower number here means that a feature has more influence, and a higher number less influence, compared to tep$_{\text{am}}$, which includes all features.

It can be noticed that in general the performance obtained was poor. However, this is consistent with the results obtained by [16], given that edge prediction is a hard problem. The results obtained by our predictor when compared with those obtained by the Adamic/Adar method are not consistent. For the ENRON dataset, the performance of our method is better and for the Version2 dataset performance is worse.

**Table 8.9** Link prediction results on ENRON and Version2 datasets. *random* is the random predictor, *tep* is the edge predictor presented here. Index is relative to $\text{tep}_{\text{am}}$. The highlighted elements are: (1) best overall for the dataset; (2) best *tep* for the dataset

| | ENRON | | | Version2 | | |
|---|---|---|---|---|---|---|
| Predictor | $E_p \cap E_a$ | % | Index | $E_p \cap E_a$ | % | Index |
| Random | 15 | 0.04 | 0.01 | 968 | 1.41 | 0.14 |
| Adamic/Adar | 720 | 2.13 | 0.59 | **9,230** | **13.46** | **1.31** |
| $\text{tep}_{\text{am}}$ | 1,212 | 3.59 | 1.00 | 7,067 | 10.30 | 1.00 |
| $\text{tep}_{\text{rd}}$ | 766 | 2.27 | 0.63 | 6,941 | 10.12 | 0.98 |
| $\text{tep}_{\text{a}}$ | 369 | 1.09 | 0.30 | 4,027 | 5.87 | 0.57 |
| $\text{tep}_{\text{ad}}$ | 459 | 1.36 | 0.38 | 4,669 | 6.81 | 0.66 |
| $\text{tep}_{\text{ngr}}$ | 344 | 1.02 | 0.28 | 3,162 | 4.61 | 0.45 |
| $\text{tep}_{\text{tg}}$ | 458 | 1.36 | 0.38 | 4,669 | 6.81 | 0.66 |
| $\text{Adamic/Adar}_2$ | 747 | 2.21 | 0.62 | 4,979 | 7.26 | 0.70 |
| $\text{Adamic/Adar}_3$ | **1,381** | **4.09** | **1.14** | 5,707 | 8.32 | 0.81 |
| $\text{tep}_{\text{nonodeage}}$ | 973 | 2.88 | 0.80 | 6,812 | 9.93 | 0.96 |
| $\text{tep}_{\text{notriads}}$ | 951 | 2.82 | 0.78 | 5,474 | 7.98 | 0.77 |
| $\text{tep}_{\text{nodistance}}$ | **1,243** | **3.68** | **1.03** | 7,069 | 10.31 | 1.00 |
| $\text{tep}_{\text{nolastedge}}$ | 1,070 | 3.17 | 0.88 | 7,030 | 10.25 | 0.99 |
| $\text{tep}_{\text{nodegree}}$ | 1,206 | 3.58 | 1.00 | **8,778** | **12.80** | **1.24** |

As a summary our results indicate the following:

- All measures used outperform the random predictor. However, the random predictor performs better on the Version2 dataset.
- There is not big difference in which aggregation operator was used.
- Prediction precision on the Version2 dataset is higher when random features are introduced. Random, $\text{tep}_{\text{rd}}$, and $\text{tep}_{\text{ngr}}$ perform better on the Version2 dataset than the ENRON dataset when random features are used.
- Using the distance feature $\mu_h$ does not show good results. It has little or worse effect in both cases.
- The Version2 dataset is less sensitive to not including some features in our predictor, while the ENRON dataset requires as many features as possible.
- The improvement over the Adamic/Adar method is small. However, the Adamic/Adar method has the highest performance on both datasets.
- For the tests performed with features disabled, we determined that the most important features are (in ranked order): (1) ENRON – triads, node age, last edge creation, degree, and distance; (2) Version2 – triads, node age, last edge creation, distance, and degree.

It can be noticed that in all cases, using temporal information will increase the precision. However, a deeper analysis needs to be done to determine which method should be used on a given network.

## Conclusions and Future Work

In this chapter, we have analyzed the robustness of centrality measures when errors are introduced in the construction of a complex social network. Our results extend the work performed in [5] by considering non random networks. In the replication of Borgatti's et al. experiments, we observed some discrepancies with our results. This may be due to the slightly different methodologies employed. However, we obtained the same linear predictability in the centrality measures that is described in [5].

Our experiments employed generative models of complex social networks and one real social network. One limitation of generative models is that while the methods for producing complex networks allow constructing networks of any size, they do not allow constructing networks with arbitrary density.

In the second contribution of this chapter, several datasets have been analyzed to obtain temporal and static features that were used in our model to predict edges in a dynamic network. Our predictive model was based on simple linear regression. The model was evaluated with two different datasets of real social networks. Our experiments show that incorporating temporal information can improve precision in prediction. This fact indicates that learning other temporal features may improve the prediction furthermore.

Our prediction model only considers predicting new edges, not if some current edges will be deleted. Changing our model to incorporate this type of prediction may improve its performance.

In our experiments, the time $T$ was defined as 1-day intervals, and the effect of changing this parameter was not investigated. We splitted time into time stamps as was done in [16]; half of the time-stamps were used for learning and the other half for prediction. However, given the nature of the datasets at hand it could happen that the initial construction phase of the network (that is used for learning) had different dynamics than the establized phase (where the predictions are made). This issue needs further investigations.

The evaluation of our edge predictor was done under the assumption that the past behavior that was not included in the dataset will not have any effect in predictor's performance. In [15] it was shown that missing part of the past do not affect in predicting the evolution of a network's diameter.

We noticed that the random predictor performs much better in the Version2 dataset used. It is interesting to note that the Adamic/Adar predictor performs also better on this same dataset. Contrarily, our method performs better on the ENRON dataset. This could be an indication that the Version2 dataset is more random in its structure than the ENRON dataset.

The simple model used in our predictor considers few correlations between features. However, in some type of networks, features are dependent, for instance node lifetime and the degree of the node. Other supervised machine learning methods such as naive bayes could also be incorporated in our preditor to improve its accuracy. Finally, we plan to apply our predictor on more datasets to obtain a more complete characterization of its performance.

## References

1. Albert, R., Barabási, A.L.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
2. Albert, R., Barabási, A.L., Jeong, H.: Mean-field theory for scale-free random networks. Physica A **272**, 173–187 (1999). doi:10.1016/S0378-4371(99)00291-5
3. Barabási, A.L., Bonabeau, E.: Scale-free networks. Sci. Am. **288**(5), 50–59 (2003)
4. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: structure and dynamics. Phys. Rep. **424**, 175–308 (2006). doi:10.1016/j.physrep.2005.10.009
5. Borgatti, S.P., Carley, K.M., Krackhardt, D.: On the robustness of centrality measures under conditions of imperfect data. Soc. Netw. **28**(2), 124–136 (2005). doi:10.1016/j.socnet.2005.05.001
6. Clauset, A., Moore, C., Newman, M.E.J.: Hierarchical structure and the prediction of missing links in networks. Nature **453**, 98–101 (2008). doi:10.1038/nature06830
7. Geng, X., Wang, Y.: Degree correlations in citation networks model with aging. Europhys. Lett. **88**, 38002 (2009). doi:10.1209/0295-5075/88/38002
8. Gloor, P.A., Niepel, S., Li, Y.: Identifying potential suspects by temporal link analysis. Technical Reports, MIT CCS (2006)
9. Gloor, P.A., Zhao, Y.: Tecflow – a temporal communication flow visualizer for social network analysis. In: ACM CSCW Workshop on Social Networks, ACM CSCW Conference (2005)
10. Goldenberg, A., Zheng, A.X., Fienberg, S.E., Airoldi, E.M.: A survey of statistical network models. Found. Trends Mach. Learn. **2**(2), 1–117 (2009)
11. Kashima, H., Abe, N.: A parameterized probabilistic model of network evolution for supervised link prediction. In: Proceedings of the 6th International Conference on Data Mining, pp. 340–349. IEEE Computer Society (2006). doi:10.1109/ICDM.2006.8
12. Klemm, K., Eguiluz, V.M.: Highly clustered scale-free networks. Phys. Rev. E **65**(3), 036123 (2002). doi:10.1103/PhysRevE.65.03612
13. Krebs, V.E.: Uncloaking terrorist networks. First Monday **7**(4) (2002)
14. Krebs V., Holley J. : Building Smart Communities through Network Weaving Know the Net. Communities **26**(3), 367–368 (1985). doi:10.1080/00420988920080361
15. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: Proceedings of the 11th ACM SIGKDD international conference on knowledge discovery in data mining, pp. 177–187 (2005). doi:10.1145/1081870.1081893
16. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. **58**(7), 1019–1031 (2007). doi:10.1002/asi.v58:7
17. Milgram, S.: The small world problem. Psychol. Today **2**, 60–67 (1967)
18. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**(2), 167–256 (2003)
19. Ortiz-Arroyo, D., Hussain, D.M.A.: An information theory approach to identify sets of key players. In: Intelligence and Security Informatics, vol. 5376/2008, pp. 15–26. Springer (2008). doi:10.1007/978-3-540-89900-6_5
20. Scott, J.: Social Network Analysis: A Handbook, 2nd edn. SAGE, London (2000)
21. Strogatz, S.H.: Exploring complex networks. Nature **410**, 268–276 (2001). doi:10.1038/35065725
22. Tang, J., Musolesi, M., Mascolo, C., Latora, V.: Temporal distance metrics for social network analysis. In: WOSN '09: Proceedings of the 2nd ACM Workshop on Online Social Networks, pp. 31–36. ACM, New York (2009). doi:10.1145/1592665.1592674
23. Watts, D.J., Strogatz, S.: Collective dynamics of small-world networks. Nature **393**, 440–442 (1998). doi:10.1038/30918
24. Xiang, E.W.: A survey on link prediction models for social network data. Technical Reports, The Hong Kong University of Science and Technology (2008)