Identifying Over-represented Temporal Processes in Complex Networks

Ursula Redmond* and Pádraig Cunningham

School of Computer Science and Informatics,
University College Dublin,
Ireland

Abstract. Temporal networks encode interactions between entities as well as the time at which the interactions took place, allowing us to identify systematic processes within the network. We can identify sub-processes or *temporal motifs* that recur frequently across a large network. In this paper, we present a strategy that allows us to identify which of a given set of temporal processes are *over-represented*. This highlights peculiarities of behaviour in the network. Our strategy involves constructing a set of interesting temporal processes, counting their embeddings in the network through subgraph matching, and then comparing this against counts in a temporally random version of the network. The network is randomized by shuffling the time-stamps in the original network. We present an evaluation on data from Prosper.com, a peer-to-peer lending website. Prosper.com was closed for regulatory reasons in 2009 and our evaluation shows interesting differences between the pre- and post-closure networks. In particular, temporal motifs indicating arbitrage are over-represented pre-closure and under-represented afterwards.

1 Introduction

Increasingly, temporal information is included with complex network data sets. Thus, instead of examining a set of static interactions between individuals, a finer-grained understanding of those interactions is now possible. Temporal networks have been used to represent a wide variety of social phenomena, from person-to-person communication to contagious disease spread by physical contact between people. The notion of spreading in a network can be more accurately identified when the times at which interactions took place are recorded.

When analyzing the processes that give a complex network its structure, recurring patterns of interaction often come to light. These frequent patterns are referred to as motifs, and are considered the building blocks of networks [16]. When temporal information is incorporated into the search for motifs, the results can have a clearer interpretation. For example, the initiator of a contagion may be easier to identify, since the first interaction in the contagion would

* This work was supported by Science Foundation Ireland [08/SRC/I1407, SFI/12/RC/2289].

have originated from that individual. Likewise, the potential reach of a piece of information in a communication network may be easier to isolate, given that propagation is time-dependent in the network.

The network we analyze in this paper comes from Prosper.com [20], a peer-to-peer lending platform. Members of the website can register to borrow and lend, and act without a bank as an intermediary. The loans among members are unsecured, so there is a risk that a member to whom you lend may default on their repayments. Temporal information is available with the data, which allows us to examine the structure and temporal dimension of some interesting motifs.

Given the wide range of credit ratings that the members have, and the ability for members to both borrow and lend, the opportunity for arbitrage arises. Members with a good credit rating can borrow money at a low interest rate, and lend the same amount at a high interest rate to members with lower credit ratings, aiming to profit from the difference in rates. The website may also provide an opportunity for members to engage in money laundering. In a simple example, money could be lent to from one member to another, and the borrowing member could default, hence completing the transfer of funds without the regulation of a bank. More complicated examples could also be imagined, involving intermediate members. In both scenarios, the network structure representing the behaviour must be composed of time-respecting paths, in which interactions occur in a non-decreasing temporal order.

The purpose of our current study is to examine the extent to which these time-dependent behaviours occur in the Prosper network to a greater extent than might be expected. To do this, we first count the embeddings of a set of time-respecting network patterns that represent this behaviour, using a subgraph matching algorithm. It is important to note that the patterns are not mined automatically - rather they are specified *a priori* and sought in the network. Then, we repeatedly re-assign the time-stamps on the interactions randomly, counting the embeddings again each time. It turns out that the presence of the time-respecting patterns is highly dependent on the timing of the interactions in the real network. This demonstrates the importance of temporal analysis for understanding behaviour in networks.

The paper is organized as follows. Section 2 presents related work, in the areas of temporal network analysis, subgraph matching and modularity. Section 3 introduces our methods for performing the matching and temporal analysis. Our results are discussed in Section 4. Section 5 concludes the paper and suggests future work.

2 Related Work

In order to asses the frequency of temporal motifs in networks, this paper draws on work from the fields of temporal network analysis and subgraph matching. To asses the significance of certain motifs in a network, we use methods from the area of network modularity.

2.1 Temporal Network Analysis

Given the prevalence of temporal information available with network data, ideas associated with static networks are being revised to take this new aspect into account. A comprehensive review [8] details these concepts. A fundamental concept in this paper is that of a *time-respecting* path, defined as a sequence of contacts which occur at non-decreasing times [10].

In a *reachability graph*, there must be a time-respecting path between nodes i and j for a directed edge to exist between them. Reachability graphs reveal the nodes which are reachable from a single root node [17]. Analysis of the reachability graph within a dating network of high-school students reveals interesting behaviour in relationships [2]. A *time-respecting subgraph* [22] is a generalization of a reachability graph, since it does not require a root node, but insists on reachability along each directed path.

The lifespan of a piece of information in a temporal communication network may be specified by a time window [25], which measures the time between the end of one communication and the beginning of the next. The closer in time the contacts take place, the higher the likelihood that the subject is the same. Similarly, the *relay time* of an interaction captures the time taken for a newly infected individual to spread the infection further via the next interaction they participate in [11]. The spread of information through a temporal network can also be modeled by a cascade. The structure of cascades can reveal spreading and community development [7]. The importance of *time-constrained* cascades is emphasized for understanding contagion [1].

Temporal motifs, as defined by Kovanen *et al.*, are connected subgraphs composed of similar event sequences, where similarity is measured in terms of the topology and temporal ordering of the events [12]. All adjacent events in a temporal motif must occur within time Δt of each other, and the events connected to a node must be consecutive in time. So if a node n in a temporal motif participates in events at times t_0 and t_2, then if an event exists involving n at time t_1, it must also be included in the motif so that the motif is valid. This is distinct from a flow motif, in which directed events that meet head-to-tail must be consecutive in time. Kovanen *et al.* propose an algorithm to find temporal motifs, which do not have the flow requirement. In contrast, the aim of our approach is to efficiently find subgraphs in which interactions occur within a specified time of each other, and in which events meeting head-to-tail are consecutive in time. In subsequent work, Kovanen *et al.* explored temporal motifs in a mobile communication network [13]. By including other attributes of the data, interesting mechanisms were found such as gender-related differences in communication patterns, and a tendency for similar individuals to communicate more often than might be expected.

2.2 Graph and Subgraph Isomorphism

The subgraph isomorphism problem determines whether a given graph contains a subgraph which has the same topological structure as another given graph.

Subgraph isomorphism is an NP-complete problem [6], so the time complexity of brute force matching algorithms increases exponentially with the size of the graphs and query graphs to be matched. This makes the problem prohibitively expensive to solve for large graphs.

Algorithms were developed which restrict the topology of the graphs and hence constrain the complexity. Such methods include the enforcement of planarity [9] or bounded valence [14]. Other approaches depend on deriving associated graphs, and on topological features such as strong regularity [5]. Another type of derived graph used is the canonical form of the graph, as in the Nauty algorithm [15].

Ullmann proposed a backtracking approach to solve the graph and subgraph isomorphism problems [24]. In an extension to the popular algorithm, the search space is pruned based on the degree of nodes in the graphs to be matched. An algorithm by Schmidt *et al.* [23] also employs backtracking, but uses the distance matrix representation of a graph to inspire the pruning steps.

The VF algorithm of Cordella *et al.* presents a depth-first search strategy for graph and subgraph isomorphism [3]. The matching process is described by a state space representation, in which each state of the process is associated with a partial solution. The partial solution includes the elements of the two graphs which match each other so far. The algorithm tries to extend each partial solution based on neighbouring nodes in the query graph and the network graph which maintain the match. The speed of the algorithm compares favourably with Ullmann's popular backtracking approach. An enhanced version, VF2, provides further performance gains [4] by substantially reducing memory requirements.

2.3 Network Motifs

Network motifs are patterns of connected nodes that occur at higher frequencies in real networks than in randomized networks [16]. Detecting network motifs gives insight into the processes that networks encode. Milo *et al.* discovered that classes of networks which performed similar functions had similar network motif profiles. For example, information processing networks from such different application areas as biomolecules within a cell and synaptic connections between the neurons in *Caenorhabditis elegans* were comprised of similar network building blocks.

To make this finding, the authors computed the occurrence frequency of a collection of motifs in a network. The structure of the network was then randomized, although each node in the randomized network maintained the same in- and out-degree as in the original network. The motifs were counted again in the randomized network. This randomization and counting was performed repeatedly, and the mean of the motif occurrences was computed. When the number of embeddings of a given motif is much lower in a randomized network, its frequency in the original network is therefore indicative of the functionality encoded by that network.

In contrast to the work of Milo *et al.*, we aim to unearth significant temporal structures of the network, rather than structural properties in isolation.

To achieve this, we count the time-respecting embeddings of query graphs that we specify. We then randomize the temporal information associated with the network, following a methodology described in Section 3. We then count the time-respecting embeddings in the randomized network. After repeating this step a number of times, we compute the average number of embeddings in the randomized networks. This reveals an interesting set of structures whose prevalence depends on processes encoded in the original version of the network.

3 Methods

This section describes our problem framework. We present our methodology for matching time-respecting subgraphs and identifying their prevalence in randomized versions of real temporal networks.

3.1 The Problem Framework

To find subgraphs embedded in a network which match the query graphs we specify, we must solve the subgraph isomorphism problem in the context of temporal networks. The definition of subgraph isomorphism for static networks may be presented as follows [24]:

Definition 1. *A graph $G2$ is isomorphic to a subgraph of a graph $G1$ if and only if there is a one-to-one correspondence between the node sets of this subgraph and of $G2$ that preserves adjacency.*

Instead of referring to an "edge" between two nodes, we use the term "interaction" to specify a triplet, made up of two nodes and the time of their contact. We define a directed temporal graph as follows:

Definition 2. *A directed temporal graph G consists of a set V of nodes and a set E of three-tuples denoting interactions. An interaction $e_i \in E$ is represented by $e_i = (u_i, v_i, t_i)$, in which u_i is the source node, v_i is the target node and t_i is the initiation time of the interaction.*

In order for a flow of information or a disease contagion to take place in a temporal network, adjacent interactions must be time-respecting.

Definition 3. *Let e_i and e_j be interactions in a directed temporal graph. The interactions are time-respecting if they are adjacent and $0 \leq |t_j - t_i| \leq d$, for some threshold d. If the interactions do not share a source node or a target node, then either $v_i = u_j$ and $t_i \leq t_j$, or $v_j = u_i$ and $t_j \leq t_i$ must be true.*

Time-respecting paths describe a non-decreasing sequence of interactions [19]. A path can be thought of as a mechanism for passing information from a source, along a sequence of intermediaries, to a target. We aim to find subgraphs composed of these paths in temporal networks.

With traditional time-slicing, the specified time window determines the interactions examined, between a minimum and maximum interaction time. However, a time-respecting path has no such bounds in reality. In fact, given the right connectivity and timing of interactions, a path might be initiated when the network is first created, and continue until the latest point in the data set. Under such circumstances, time-slicing can lose a lot of important context and information.

We define a time-respecting subgraph in terms of time-respecting interactions. We seek query graphs that are connected, so we require that the embedded subgraphs are connected.

Definition 4. *A time-respecting subgraph $S = (V', E')$ of a temporal graph $G = (V, E)$ is composed of a set of nodes $V' \subseteq V$, from which any pair of nodes is connected via a set of interactions $E' \subseteq E$ such that the nodes comprising interactions in E' are in V', and every adjacent interaction pair is time-respecting.*

In our implementation, embedded subgraphs are induced. So, given any pair of nodes in an embedded subgraph, all interactions between them are included in the embedding. So, if a potential embedding includes more interactions than specified by the query graph, the embedding will not be returned.

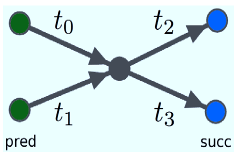

Fig. 1. An example of a time-respecting subgraph. Here, $t_0 \leq t_1 \leq t_2 \leq t_3$. All interactions which are incident to the same node must occur within time d of each other. Thus, we require that $|t_3 - t_0| \leq d$. All incoming edges to a node n must precede all outgoing edges from node n. So, we must have that $t_0 \leq t_2$, $t_0 \leq t_3$, $t_1 \leq t_2$ and $t_1 \leq t_3$.

3.2 The Matching Algorithm

We retain the notation used in the description of the recursive VF2 algorithm by Cordella *et al.* [4]. The matching process is described by a state space representation, in which each state s of the process represents a partial mapping solution. In a state s, a portion of the query graph $G2$ matches a portion of the network graph $G1$. The portion of $G1$ in the mapping is induced. So, given a set of nodes in the mapping, any interactions between them are also present in the mapping.

Given such an intermediate state s, the mapping is extended by first computing candidate node pairs (one node each from $G1$ and $G2$). The candidate node from $G2$ is selected from the set of neighbours of the nodes in $G2$ that are currently in the mapping. This guarantees that the node is connected to the portion of the query graph currently matched. The candidate node from $G1$ is selected in the same way, from the neighbours of the nodes currently matched in the embedding from $G1$, so the embedding will be connected. Once the new nodes are included in the mapping, all interactions between them are also included. The two new, extended portions in the mapping must be graph isomorphic in order to be considered a feasible match. If they are not graph isomorphic, the candidate nodes are discarded as a matching pair, and the process then continues with a new node pair.

If a topological match is confirmed, a semantic match is considered. In our setting, we utilize the dates on which the interactions occur in $G1$. Given an embedding of the subgraph $G2$ in the graph $G1$, we don't require that the dates on each paired interaction match each other, but rather that the partial embedding of $G2$ in $G1$ is time-respecting.. Since we are interested in the actual times at which interactions occurred in the network data, only the semantic feasibility of $G1$ is checked.

The memory requirements of the VF2 algorithm are constrained through the use of data structures which are maintained at each recursion level. We keep track of both topological and temporal information in the same way. A map data structure named *core_1* contains the nodes in the current mapping from $G1$ to $G2$. This provides an efficient way for us to test that a candidate node for inclusion in the mapping will maintain the time-respecting property we require for all of the induced edges.

Before testing the legitimacy of a candidate node $G1\_node$, we construct a set of data structures. The list *pred* contains the predecessors of $G1\_node$ in $G1$ which are also in *core_1*, and thus part of the current mapping. Analogously, *succ* contains the successors of $G1\_node$ in $G1$ which are also in *core_1*. The lists *pred_dates* and *succ_dates* contain the dates, in increasing order, on which connections between $G1\_node$ and the relevant predecessor or successor nodes, respectively, were made. The list *dates* combines these dates, sorted in increasing order.

As described in Definition 4, a pairwise comparison of adjacent interactions must ensure that each pair is time-respecting. Accordingly, a candidate node must fulfil these criteria when included in a potential embedding of $G2$ in $G1$.

3.3 Re-assigning Time-stamps

We aim to discover the extent to which the number of embeddings of a query graph in the network is uniquely a property of the temporal aspect of the network. To ascertain this, we repeatedly re-assign the time-stamps on the interactions, and count the number of embeddings again each time. The re-assignment is performed by first stripping all of the time-stamps off the interactions. We then shuffle the order of this time-stamp collection using the shuffle algorithm from Python's built-in *random* module. We then iterate over the entire set of interactions, assigning a time-stamp to each interaction. Thus, the re-assignment is global in scale.

4 Results

To find out whether certain types of time-respecting subgraph are characteristic of real networks, we constructed a temporal network and a set of query graphs in order to perform our experiments. This section details the network data used, the query graphs, our analysis and the results.

4.1 Network Data

The website at Prosper.com [20] provides a forum for prospective borrowers and lenders to connect and exchange funds. Prosper.com allows members to borrow and lend without the presence of a bank. This means that borrowers with low credit-worthiness have a better chance to get loans, since the requirements for being funded are lower. It also gives people a chance to invest smaller amounts, to experiment with lending.

For the purpose of our experiments, we constructed a directed temporal network of lenders and borrowers, connected via loans. An interaction is composed of a source (the lender) and a target (the borrower) and represents the money sent in contribution to a loan request. An interaction also contains the time at which the money was transferred. We set the d-value (maximum time allowed between interactions) to 6 days, to reflect the time-scale at which the network operates.

Since the Prosper.com marketplace closed for a period in 2009 due to regulatory issues, we extracted two portions of the network; one before and one after the temporary closure. This allowed us to compare the social behaviours that occurred in the network as a result of different levels of regulation. The details of these networks are listed in Table 1. An important point to note is that the duration of each network is the same, as is the size of each network. So, when the time-stamps are randomly re-assigned, there is the same amount and variation in the time-stamps.

Network	Start Date	End Date	Order	Size
Pre-closure	1^{st} November 2006	31^{st} December 2006	8,690	72,215
Post-closure	1^{st} September 2009	31^{st} October 2009	7,201	77,026

Table 1. The pre-closure network spans the last three months of 2006, while the post-closure network runs from the beginning of September to the end of October in 2009. Both networks have a similar number of interactions, and occur over the same amount of time. This means that shuffling the time-stamps on the interactions ought to have a similar effect in both networks, since the distribution of time-stamps and interactions is comparable.

4.2 Query Graphs

We enumerate some small directed query graphs that have clear interpretations in the context of the Prosper network, illustrated in Fig. 2. These only encode the topological structure of the patterns we are interested in. When we examine their topological embeddings in the network, we also check that the embeddings are time-respecting, so the notion of non-decreasing activations along the paths in the query graphs is maintained.

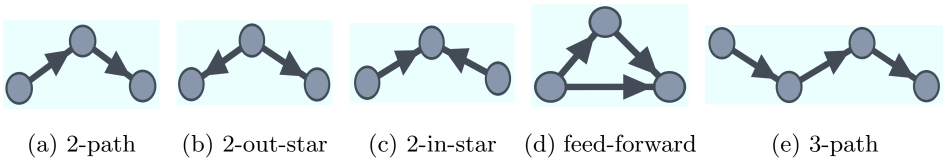

Fig. 2. The query graphs sought in the Prosper network. Each node represents a member of the Prosper marketplace, and each interaction represents a sum of money being transferred via a loan. These queries were chosen since their structure is clear in the context of the network data.

4.3 Analysis

The experiments were performed on a Linux server with a 2 GHz processor, limited to 5GB of physical memory. The algorithms we proposed for performing the subgraph matching and the re-assignment of time-stamps were implemented in the programming language Python [21], using the NetworkX library [18]. The VF2 algorithm is included in this library, and was implemented as part of a project at the Complexity Sciences Center and Physics Department, UC Davis. We extended this implementation to process temporal networks and use temporal information during the matching process. Our implementation can handle directed graphs as well as directed multigraphs (graphs with multiple interactions between nodes).

The results of our experiments are listed in Table 2. In both the pre- and post-closure network, the 2-in-star and 2-out-star queries had the highest number of embeddings. This is likely to be a result of how the Prosper marketplace is used; by members who either exclusively borrow or lend. Borrowers have a high in-degree, since the loans they request are funded from many sources, who all give relatively small amounts. Lenders have a high out-degree, since they need to distribute their lending portfolio over a range of borrowers in order to make a more reliable profit.

When the time-stamps are randomly re-assigned in the pre-closure network, the number of embeddings of the query graphs drops between 20% and 89.1%. This strongly suggests that the actual timing of the interactions was important

Network	Structure	Count	Mean	Std. Dev.	Decrease
Pre-closure	2-path	15,061	11,717.8	166.4	22.2%
	2-out-star	1,083,154	819,451.4	3,826.1	24.3%
	2-in-star	5,209,926	1,083,642.2	2,215.9	79.2%
	feed-forward	1,130	123.0	12.5	89.1%
	3-path	1,584	1,267.5	181.5	20.0%
Post-closure	2-path	6,237	6,411.4	179.8	-2.8%
	2-out-star	1,064,034	786,908.3	2,784.4	26.0%
	2-in-star	17,900,180	3,806,272.3	9,061.4	78.7%
	feed-forward	1,516	217.0	20.7	85.7%
	3-path	825	847.0	219.7	-2.7%

Table 2. Comparing the number of embeddings found in the pre- and post-closure networks. The count lists the number of embeddings with the original time-stamps in place. The mean shows the average over 100 separate counts of the number of embeddings after randomly re-assigning the time-stamps every time. In the pre-closure network, embedding counts dropped after shuffling the time-stamps. The same was true in the post-closure network, except for the path query graphs, indicating that their presence in the network is not necessarily a property of the original network.

for the processes to take place. The greatest drop in the number of embeddings occurs with the queries containing a higher in-degree. This makes sense, since a borrower needs to get funds from multiple lenders at around the same time for a loan to go ahead. If the time-stamps are shuffled, this condition may not be met. This demonstrates the effectiveness of our strategy; real social behaviour in the network is shown to be dependent on interaction timing.

The most interesting results relate to the path queries in the post-closure network. An intermediate node in a path may represent an arbitrageur. An arbitrageur aims to profit from the difference in interest rates between the loan taken on and the loans given to borrowers. The timing of this sequence of loans is important for the arbitrage to be successful. The different results for the pre- and post-closure network indicate the influence of greater regulation within the marketplace. Specifically, the number of embeddings of path queries does not decrease when the time-stamps are re-assigned. Thus, the existence of the time-respecting paths in the original network does not reveal a process that is unique to the network. This is consistent with the fact that stronger regulation may have discouraged arbitrage. In the case of money laundering, the existence of intermediate individuals is also a possibility. So, this result also indicates that if attempts at money laundering occurred in the pre-closure network, it was discouraged by greater regulation.

5 Conclusions and Future Work

The primary aim of this work is to evaluate the importance of temporal information in a network for identifying the processes that underly the network topology. Specifically, we examined the network from Prosper.com to see if the existence of some suspicious patterns was dependent on the time at which the interactions which made up the pattern took place. To do this, we specified some query graphs to search for in the network and counted their time-respecting embeddings. Then, we randomly re-assigned the time-stamps and counted the embeddings again. We did this latter step 100 times, and took the average of the counts. Almost all the counts dropped in comparison with the actual network.

Since Prosper.com closed due to regulatory issues in 2009, we compared a portion of the pre- and post-closure network to see if there was a change in behaviour. The query graphs associated with arbitrage or potentially money laundering behaviour were prevalent in the pre-closure network, but not so in the post-closure network. This was revealed by the fact that, after temporal randomization, the number of embeddings dropped in the pre-closure network, but did not change in the post-closure network. This is likely to be an effect of increased regulation on the lending platform.

The time at which interactions take place is a key component of network formation, and can help to explain many types of emergent social behaviour. In future, we aim to apply these methods to other networks which contain temporal information, especially networks which operate at a finer temporal grain. This will help to validate the performance of our algorithm, and may give an insight into which processes play a significant role in the networks in question. Given our prior knowledge of the Prosper network, our validation of what constitutes an interesting pattern is intuitive. In future, a method for automatically extracting over-represented patterns would overcome this dependency and potentially yield unforeseen network behaviour.

References

1. Baños, R.A., Borge-Holthoefer, J., Moreno, Y.: The role of hidden influentials in the diffusion of online information cascades. arXiv preprint arXiv:1303.4629 (2013)
2. Bearman, P.S., Moody, J., Stovel, K.: Chains of affection: The structure of adolescent romantic and sexual networks. American Journal of Sociology 110(1), 44–91 (2004)
3. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: Performance evaluation of the VF graph matching algorithm. In: Image Analysis and Processing, 1999. Proc. Intl Conf. on. pp. 1172–1177. IEEE (1999)
4. Cordella,L. P., Foggia,P., Sansone, C., Vento, M.: An Improved Algorithm for Matching Large Graphs. 3^{rd} IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition pp. 149–159 (2001)
5. Corneil, D.G., Gotlieb, C.C.: An efficient algorithm for graph isomorphism. Journal of the ACM (JACM) 17(1), 51–64 (1970)
6. Gary, M.R., Johnson, D.S.: Computers and intractability: A guide to the theory of np-completeness (1979)
7. Ghosh, R., Lerman, K.: A framework for quantitative analysis of cascades on networks. In: Proc. of the fourth ACM Intl. conf. on Web search and data mining. pp. 665–674. ACM (2011)
8. Holme, P., Saramäki, J.: Temporal networks. Physics reports 519(3), 97–125 (2012)

9. Hopcroft, J.E., Wong, J.K.: Linear time algorithm for isomorphism of planar graphs (preliminary report). In: Proc. of the sixth annual ACM symposium on Theory of computing. pp. 172–184. ACM (1974)
10. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. In: Proc. of the thirty-second annual ACM symposium on Theory of computing. pp. 504–513. ACM (2000)
11. Kivelä, M., Pan, R.K., Kaski, K., Kertész, J., Saramäki, J., Karsai, M.: Multiscale analysis of spreading in a large communication network. Journal of Statistical Mechanics: Theory and Experiment 2012(03), P03005 (2012)
12. Kovanen, L., Karsai, M., Kaski, K., Kertész, J., Saramäki, J.: Temporal motifs in time-dependent networks. Journal of Statistical Mechanics: Theory and Experiment 2011(11), P11005 (2011)
13. Kovanen, L., Kaski, K., Kertész, J., Saramäki, J.: Temporal motifs reveal homophily, gender-specific patterns and group talk in mobile communication networks. Proc. of the National Academy of Sciences 110(45), 18070–18075 (2013)
14. Luks, E.M.: Isomorphism of graphs of bounded valence can be tested in polynomial time. Journal of Computer and System Sciences 25(1), 42–65 (1982)
15. McKay, B.D.: Practical graph isomorphism. Department of Computer Science, Vanderbilt University (1981)
16. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
17. Moody, J.: The importance of relationship timing for diffusion. Social Forces 81(1), 25–56 (2002)
18. NetworkX Developers: NetworkX. networkx.github.io (2013)
19. Pan, R.K., Saramäki, J.: Path lengths, correlations, and centrality in temporal networks. Physical Review E 84(1), 016105 (2011)
20. Prosper Marketplace Inc.: Personal Loans and Online Investing – Peer to Peer Lending – Prosper. http://www.prosper.com/ (2013)
21. Python Software Foundation: Python. www.python.org (2012)
22. Redmond, U., Cunningham, P.: A temporal network analysis reveals the unprofitability of arbitrage in the prosper marketplace. Expert Systems with Applications (2012)
23. Schmidt, D.C., Druffel, L.E.: A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. Journal of the ACM (JACM) 23(3), 433–445 (1976)
24. Ullmann, J.R.: An algorithm for subgraph isomorphism. Journal of the ACM (JACM) 23(1), 31–42 (1976)
25. Zhao, Q., Tian, Y., He, Q., Oliver, N., Jin, R., Lee, W.C.: Communication motifs: a tool to characterize social communications. In: Proc. of the 19th ACM Intl conf. on Information and knowledge management. pp. 1645–1648. ACM (2010)