Link Prediction and Threads in Email Networks

Qinna Wang
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005 Paris, France.
CNRS, UMR 7606, LIP6, F-75005 Paris, France.
Email: qinna.wang@gmail.com

***Abstract*—We tackle the problem of predicting future links in dynamic networks. For this, we work with the Debian Mailing Lists. In this dataset, a user can post a question to the debian list and other users can reply it by email forming a thread. We show that the number of threads shared in the past between users is a better feature to predict future email exchanges than classical features, like the number of common neighbors. We also show that the structure of a thread do not match the traditional definition of a community, particularly a thread does not have many triangles and has many outgoing connections. While the number of shared (detected) communities is also a better feature to predict future email exchanges than traditional features, is not as good as the number of shared threads. We believe our work should raise interests in characterizing and detecting thread-like structures in dynamic networks.**

I. Introduction

A social network is a social structure modelled as a graph, where nodes (vertices) represent people or other entities embedded in a social context, and edges represent social ties like friendship, kinship, dislike, conflict or trade, among entities. Social networks have been studied in many fields ranging from computer science to biology [4] [19] [13] [22]. These studies have opened unsuspected directions for research and a wealth of applications in understanding the nature of social networks.

Social networks are highly dynamic: over time, people or other entities create and deactivate social ties, thereby altering the structure of the networks in which they participate. How to capture the mechanism of network evolution is still a question that is not well understood. To address this problem, a lot of work has been done [16] [23] [6]. One study [17] introduced the link prediction problem: it estimates the probability that a new link between a pair of unconnected entities that will appear in the future. A new appeared link signifies the appearance of new interaction in the underlying social structure.

The early link prediction model [17] works explicitly on a social network. They estimated the similarity between a pair of vertices by various graph-based similarity metrics and used the ranking on similarity scores to predict the link between two vertices. Later, this work is extended in two ways. First, the external data outside the scope of graph topology is used to improve the prediction result. For example, [25] used friends-of-friends and place-friends to study a large real-world service called Gowalla, where friends-of-friends are all those users that share at least one friend without being directly connected, and place-friends are all those users that have visited at least one common place but are not connected to each other. Second, various similarity metrics as features are used in a supervised learning setup [2] where a binary classification algorithm (such as SVM, Decision Tree, and so on) is applied to predict links.

Recently, the field of relational learning has been extended for predicting link existence [11] [27]. As the main formal approach that extends Bayesian networks to the relational domain, probability relational models (PRMs) incorporate both vertex and edge attributes to model the joint probability distribution of a set of entities and the links that associate them. The advantage of PRMs is that they can incorporate the attributes of the entities to the model.

Besides these, there many approaches concern the evolution of social networks. For instance, [26] used a matrix factorization to estimate the similarity between nodes in real life social networks such as Facebook and MySpace. [28] showed that considering the time stamp of the previous interactions significantly improves the accuracy of the link prediction model. Additionally, there are surveys of link prediction algorithms [3] [18].

Our study uses a different approach that uses temporal topologies. The network is described by a link stream, where each interaction is temporal. In this aspect many link prediction approaches focus on the attributes based on the static topology of social network, while our investigations directly use temporal topologies to address the link prediction problem.

This paper focuses on one particular temporal structure called thread. In Debian Mailing Lists, users can post questions to the Debian List and developers can reply by email forming threads. We use different measures to investigate the basic properties of these threads (Section II). A link predictor based on the threads is then proposed (Section III). We studied whether a thread matched to the traditional definition of community. The analysis shows that a thread does not have many triangles and has many outgoing connections. Moreover, we propose another link predictor which is based on community structure (Section IV). Section V shows the discussion of our work. The conclusion is given by the last section.

II. Dataset

The data used for our analysis are email messages. They are extracted from Debien Mailing List from January 01, 2010 to December 31, 2013. This dataset has been used in [8] for analysing some properties at three different levels: the thread level, the labelled thread level, and the interaction network itself.

A. Description of Threads

Each email message has exactly one header, which is structured into fields such as:

- From: The email address, and optionally the name of the author(s).

- Date: The local time and date when the message was written.
- Message-ID: Also an automatically generated field; used to prevent multiple delivery and for reference in In-Reply-To: (see below).
- In-Reply-To: Message-ID of the message that this is a reply to. Used to link related messages together. This field only applies for reply messages.

Consequently, each message m (denoted by the value in "Message-ID" field) in our dataset can be labelled with the author $A(m)$, the date $T(m)$, the father $F(m)$ and the root $R(m)$. Here, $A(m)$ is given by the email address in the "From" field, $T(m)$ is the value in the "Date" field and the father $F(m)$ is derived from the Message-ID in "In-Reply-To" field. If m has no father defined this way (it is not an answer to any other message) then we put as a convention that $F(m) = m$. Moreover, this leads to the definition of the root. The root of a message m is either m itself if $F(m) = m$, or else it is the root of $F(m)$ such that $R(m) = R(F(m))$.

Our data is a set of messages $M = \{m_i\}_{i=1,2,...}$, where each message

$$m_i = (A(m_i), T(m_i), F(m_i), R(m_i)).$$

We now define the thread $\mathbf{T}$ which is a set of messages such that all messages in the set have the same root and no other does. Notice that a thread always contains exactly one root, which we denote by $r(\mathbf{T})$, and each root r defines exactly one thread. We define a thread defined by a root r as $\mathbf{T}(r) = \{$ for all $m_i \in M$ such that $R(m_i) = r\}$. The duration of a thread is denoted by $\overline{\mathbf{T}} = \arg\max_{m_i \in \mathbf{T}} T(m_i) - \arg\min_{m_i \in \mathbf{T}} T(m_i)$.

In our experiment, the data is filtered corresponding to the thread information and structural topology. As we selected the email messages from January 1, 2010 to December 31, 2013. We set the beginning time $t_{\min} = 1262304000$ and the end time $t_{\max} = 1388534399$. The former describes the time 00:00:00 UTC on January 1, 2010 and the latter corresponds to the time 23:59:59 UTC on December 31, 2013. A time threshold for filtering the root is defined as the middle of the selected period such as $t_{thr} = \frac{t_{\min}+t_{\max}}{2}$. Likewise, a thread duration threshold is set such that $\overline{D}_{thr} = \frac{t_{\max}-t_{\min}}{2}$.

Then, for all used messages, their root should occur in the period $[t_{\min}, t_{\text{thr}}]$ such that for all $m_i \in M$, $t_{\min} \leq T(R(m_i)) \leq t_{\text{thr}}$. The threads that these messages belong to should have the duration equal to or less than the duration threshold such that $\{\mathbf{T} \subseteq M : \overline{\mathbf{T}} \leq \overline{D}_{thr}\}$.

B. Link Stream Representation

Here, we use the link stream [29] to represent the structure of threads. This representation can be used to characterize the temporal and topological features of threads.

A stream L is a sequence of time-ordered triplets: $L = (l_i)_{i=1,...,n}$, where each link $l_i = (t_i, u_i, v_i)$ represents a connection between node u_i and v_i at time t_i. We call $|L| = n$ its size and $\overline{L} = t_n - t_1$ its duration.

We denote by $V(L)$ the set of all nodes in L, $E(L)$ the set of pairs of connected nodes in L, and $T(L)$ the set of all time instants in L. We define that each message can be represented by a link such that $l(m) = (T(m), A(m), A(F(m)))$. A stream can be constructed for our message set, i.e., $L(M) = (l(m_i))_{i=1,2,...}$ for all $m_i \in M$.

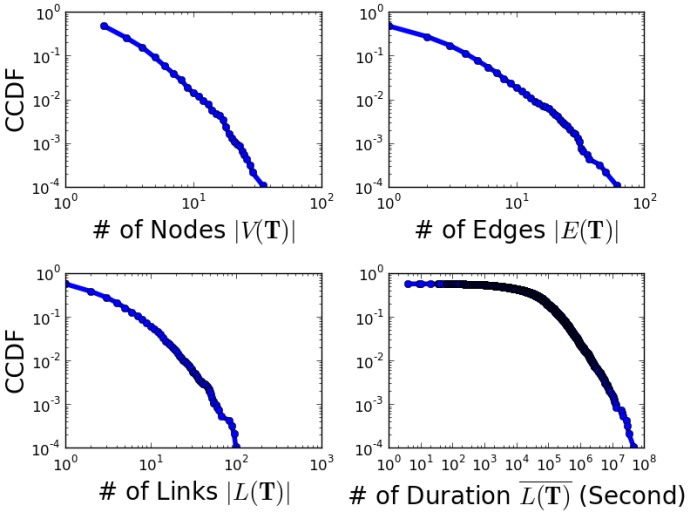


Fig. 1: Complementary Cumulative Distribution Function of the number of Nodes, Edges, Links and Durations of threads

Similarly, a sub-stream can be used to represent a thread, which is denoted by $L(\mathbf{T})$. Each message m_i is represented by a link $l_i = (t_i, u_i, v_i)$, where t_i is its timestamp, u_i and v_i denote its author and father respectively. A sub-stream $L(\mathbf{T})$ has a set V of nodes and a set E of edges, where the set V represents the set of authors and the set E corresponds to the set of pairs of authors which send at least one email to each other.

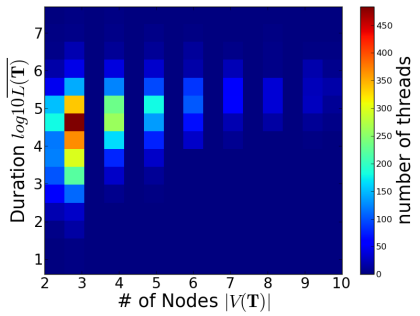
Furthermore, some messages may result self-loops in the structural topology. In our following testing, each sub-stream that represents a thread only contains a set of no self-loop links such that for all $l_i \in L(\mathbf{T}), u_i \neq v_i$.

C. Basic Characteristics of Threads

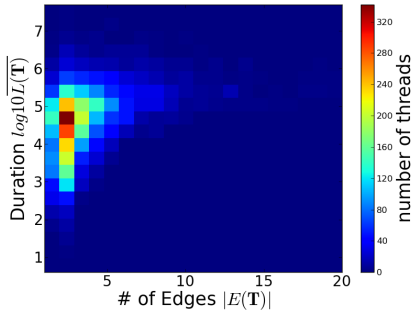
In this section, we present a statistical characterization of the structure of such threads using stream presentation. Fig. 1 shows the distribution of the thread size, i.e., the number of nodes, edges and link, and thread durations. We observe that the distribution of thread size has one tail: only 1.4% of threads have more than 10 nodes; nearly 1.8% of threads have more than 10 edges; and 94% of threads have less than 10 links. Our results show that most of threads do not have large size. On average, each thread has 3.2 nodes, 3.4 edges and 3.6 links.

Different from the number of nodes, edges and links, the distribution of thread duration with the average value 10^5 seconds has two tails. It indicates that there are very few threads whose duration is as short as 15 minutes while there exist some threads whose duration is longer than 1 day.

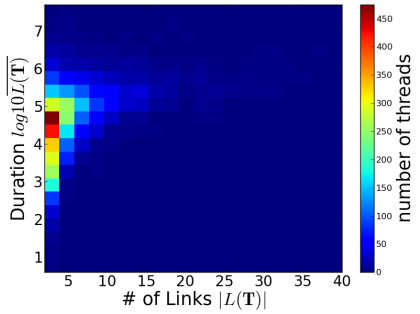
Next, we study the correlation between thread size and thread duration. From Fig. 2, we observe a small correlation: the threads that have short duration typically have small size. For instances, the threads that have the duration less than 10^4



(a)



(b)



(c)

Fig. 2: Heat maps reflect the relationship between thread size and thread duration. (a) thread nodes (b) thread edges (c) thread links.

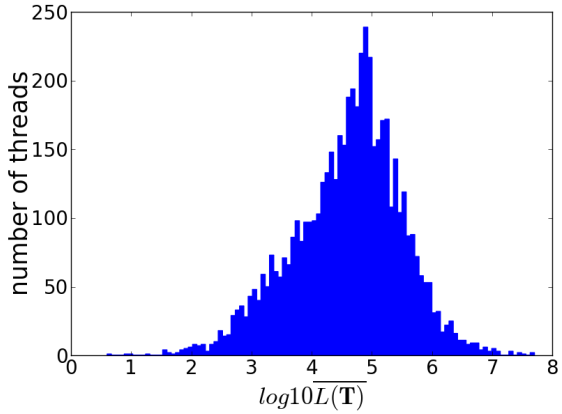


Fig. 3: The histogram of thread duration in $log10\overline{L(\mathbf{T})}$ unite.

Features	$W = 7$ days	$W = 30$ days
$\|V^t\|$	126.97 ± 11.57	307.5 ± 20.72
$\|E^t\|$	191.90 ± 30.85	742.88 ± 86.93
$\|L^t\|$	314.94 ± 59.08	1339.625 ± 195.39
N	61.87 ± 17.39	309.86 ± 47.49

TABLE I: The basic features of snapshots with different time windows. The numbers (Mean $\pm$ SD) of nodes, edges, links and changing pairs of nodes N for each snapshot are shown.

seconds have less than 6 nodes. However, it is the case for small threads. They are not representative. As shown in Fig. 3, there is a peak at the duration of 10^5 seconds. It represents that many threads have the duration of nearly 1 day. The effect of thread size to thread duration is not significant.

III. Threads and Link Prediction

A. Link Prediction Method

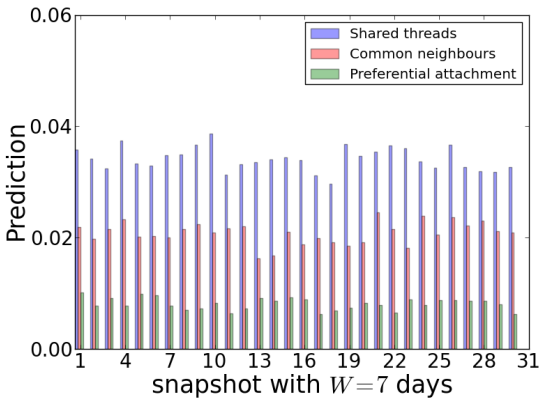
The evaluation of link prediction methods in our experiments is similar that [17] [12]. It measures the goodness of predictors based on a training part and a testing part. For the training part, it contains a set V_{training} of nodes and a set E_{training} of pairs of connected nodes. So does the testing part which has two sets: V_{testing} and E_{testing}. We estimate the probability of new appearing interactions in the set: $E_{\text{p}} = V_{\text{training}} \times V_{\text{training}} - E_{\text{training}}$.

All the predictors assign predicted connection weight $\text{score}(i, j)$ to unconnected pairs of nodes $\langle i, j \rangle$, based on the training part, and then produce a ranked node pair list in decreasing order of $\text{score}(i, j)$, whose value is treated as proportional to the estimated probability of forming a new link between i and j.

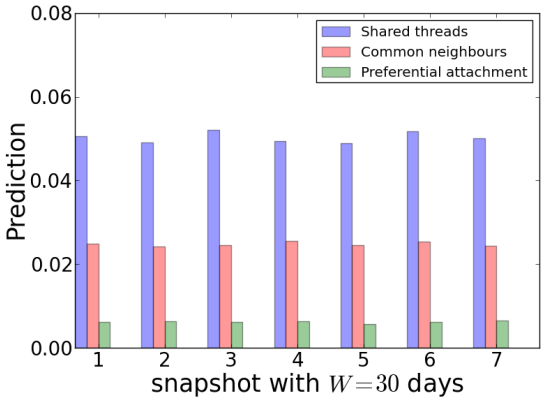
In this way each link predictor outputs a ranked list of node pairs in which would eventually form predicted links. From this list (sorted in decreasing values of scores), the set of first N entries is taken. We denote it by E_p^* such that $\{E_p^* : E_p^* \subseteq E_p\}$ and $N = |E_p^*|$. The goodness of a predictor is measured by the percentage of pairs of nodes in E_p^* that are present in the E_{testing}. This goodness measure is called the *prediction*.

In the following, we apply different predictors in a sequence $(L^t)_{t=1,2,...}$ of snapshots [7] with a chosen length of time window W. Assuming that the sequence $(L^t)_{t=1,2,...}$ starts from the timestamp t_0, each snapshot L^t is a set of links whose timestamps are between $t_0 + W * (t-1)$ and $t_0 + W * t$. Given a snapshot L^t with a set V^t of nodes and a set E^t of pairs of connected nodes, each predictor gives a score to every pair of unconnected nodes, which estimates the probability of creating a new connection at the successive snapshot L^{t+1}. For the evaluation, the value N corresponds to the number of pairs of nodes that are not connected in L^t but have connections in L^{t+1} such that $N = |E^{t+1} \cap (V^t \times V^t) - E^t|$.

It should be noted that the predictors that discussed in [17] [12] can identify $< 10\%$ of new emerging links. It means that most methods to the link-prediction problem give rather poor results.



(a)



(b)

Fig. 4: Prediction results for different methods on Debian Dataset with different time windows: (a) $W = 7$ days. (b) $W = 30$ days. Here the result is obtained by running each method 100 times.

B. Evaluation

In this experiment, we use the data during the period of April 01,2010 and October 31, 2010. When the length of time window is 7 days, there are 30 snapshots and when the length of time window is 30 days, there are 7 snapshots. Their basic features are shown in Tab. I. Based on each snapshot L^t, we use every method to predict future connections on its successive snapshot L^{t+1}. We have compared the results based on the number of shared threads between pairs of unlinked users with the other two standard methods: common neighbours (CN) and preferential attachment (PA) predictor.

Let $\tau(x)$ denote the set of threads that the node x participates in a snapshot L^t. We define the measure $score(x, y) := |\tau(x) \cap \tau(y)|$, the number of threads that the nodes x and y participate in common. For other predictors, let $\Gamma(x)$ denote the set of neighbours of x in a snapshot L^t. The CN predictor defines $\text{score}(x, y) := |\Gamma(x) \cap \Gamma(y)|$. And the preferential attachment corresponds to measure $score(x, y) := |\Gamma(x)| \cdot |\Gamma(y)|$.

To more meaningfully study predictor quality, we use a random predictor, which simply picks randomly selected pairs of nodes that are not linked in the training part.

Feature Used	Prediction	
	$W = 7$ days	$W = 30$ days
Shared Threads	0.034 ± 0.0020	0.050 ± 0.0012
Common Neighbours	0.021 ± 0.0019	0.025 ± 0.0004
PA	0.008 ± 0.0010	0.006 ± 0.0002
Random Links	0.007 ± 0.0113	0.006 ± 0.0044

TABLE II: The average prediction accuracy (mean±SD) obtained by different methods on Debian Dataset with different time windows.

Fig. 4 shows the performance of several different predictors: the number of shared threads, the number of common neighbours and the PA. Considering that some node pairs may obtain the same score values, running one predictor several times can lead to different prediction values. Therefore, we run each predictor 100 times and compute the average prediction. We see that using the number of shared threads consistently outperforms the other methods.

As indicated in Tab. II, almost every predictor performs better than the random predictions (except the predictor by using PA in the snapshots with the length 30 days whose performance is similar as random predictor). The number of shared threads achieves the best performance for different lengths of time windows. Especially when increasing the length of time window, the performance of CN and PA only has subtle changes. The performance of the number of shared threads, however, improves a lot. Its average prediction accuracy is nearly 5%. This value is nearly twice as big as others. This observation suggests to further develop our methods and thread detection.

IV. Thread, Community and Link Prediction

A. Thread and Community

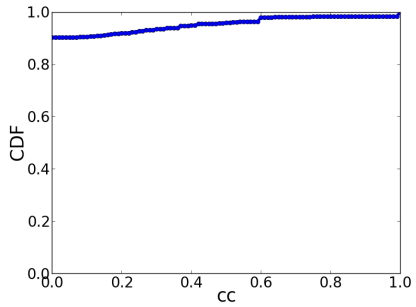
Considering the good performance of threads in link prediction, we are interested in determining how individuals interact and form threads over time. We start from the distribution of the clustering coefficient. Here we use the concept of transitivity ratio to measure the average probability of a tie randomly established between two nodes in each thread.

The transitivity ratio is defined by:

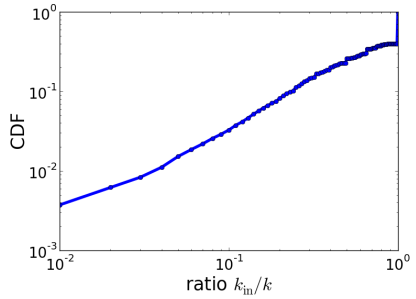
$$CC = \frac{3 \times \text{number of triangles}}{\text{number of connected triples of vertices}}$$

From Fig. 5a, we observe only a small portion of threads that have the clustering coefficient value larger than 0. The value of clustering coefficient corresponds to the number of triangles. Therefore, we learn that the threads in the email network do not have many triangles.

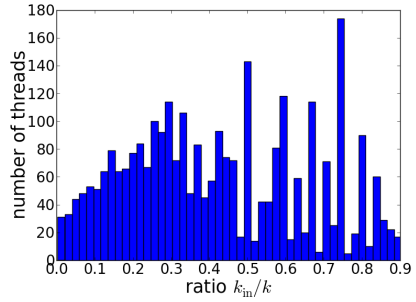
We then measure the ratio k_{in}/k of the number of links within each thread to the total number of links connected to the thread. Given a thread $\mathbf{T}$ which consists of the set V of nodes, the number of links that connected to it is the sum of links that connected to the node set V during the period when the thread $\mathbf{T}$ is active. This ratio measures how community-like is a thread, indeed. A community [10] should be a group



(a)



(b)



(c)

Fig. 5: (a) Cumulative Distribution Function of threads' clustering coefficient. (b) Cumulative Distribution Function of the ratio k_{in}/k, the fraction of the number of links within each thread to the total number of links connected to the thread. (c) The histogram of threads' ratio, where the test threads have at least 3 links and connect to the rest of the network.

of nodes with more intra-connections than inter-connections that connect to the rest of network.

Fig. 5b shows the distribution of the ratio k_{in}/k of threads found in our dataset. We observe that 60% of threads have the value of k_{in}/k equal to 1.0, i.e., have no links outside their threads. This means that a substantial fraction of threads are disconnected. This result is different from the topological graph obtained by our dataset which only has nearly 15 disconnected edges. By plotting the histogram of threads' ratio for threads that have at least 3 links and connect to the rest of the network in Fig. 5c, we observe a large portion nearly 60% of threads that have the ratio k_{in}/k less than 0.5. This means that many threads have high number of outgoing links.

Our above statistical analysis indicate that a thread does not satisfy the traditional definition of a community because a thread does not have many triangles and has many outgoing connections.

B. Community Structure and Link Prediction

Next, we study the community structure. Here, we first compare the structure of threads and communities, and then we study the predictive power of communities.

For the comparison, we detect the community which is a set of closely interrelated links. Most of existing community detection algorithms [5] [21] [15] are designed for discovering communities constituted by nodes. We use line graph [9] [1] to represent a stream. Such a line graph I is constructed by connecting all pairs of links $\langle l_i, l_j \rangle$ if they share a common end point and the intercontact time is less than or equal to Δ. In our test, we set $\Delta = 86400$ seconds (1 day) which matches to human behaviour cycle and corresponds to the duration of many threads. Based on a line graph which combines topological and temporal relations, we are able to detect link communities.

Algorithms for finding communities are quite diverse. In this section, we run Louvain algorithm [5] based on the modularity quality function [20] with a resolution parameter [24] and IOLoCo (Local community identification in social networks) [21]. The Louvain algorithm produces a hierarchical community structure. Here, we select the partition at the lowest level of the hierarchy as it produces the best results. IOLoCo finds the local community for a given node. For each node in I, we find its local community which is used to estimate the predictive scores.

Based on the structures of threads and communities, we measure their normalized mutual information generalized for overlapping community structure [14]. The normalized mutual information quantifies the similarity between two sets of link groups. Its value equals 1 if the sets of groups are identical, whereas it has an expected value of 0 if the sets of groups are independent. From Tab. III, we observe that the similarity between the community structure and thread structure is very low. This result tells us that the thread structure is different from our detected community structure and we can't characterize the threads by our found link communities.

As motivated by our above findings, using the number of shared threads can address the prediction problem, we assess the predictive power by using the number of shared communities. We detect link communities on each training snapshot. Each link corresponds to a temporal contact between a pair of nodes. It allows that each node belongs to at least of one link communities. It also allows us to compute predictive scores for every pair of disconnected nodes by using the number of link communities that they share. Then we numerically rank these candidates according to their score and predict pairs of linked nodes in the testing snapshot.

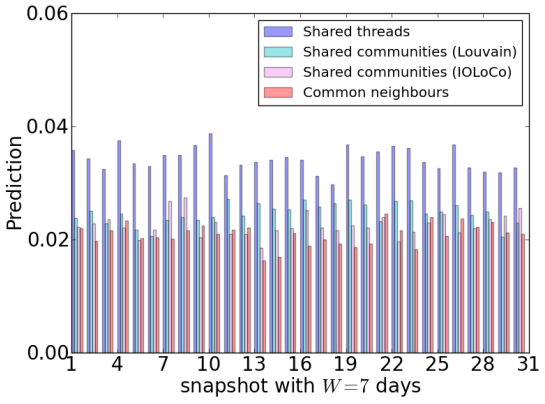
Figure 6 shows predictors' performance on our two sequences snapshots (see Sec. III-B). These predictors use the features including the number of shared threads, shared communities found by Louvain algorithm, shared communities found by IOLoCo and common neighbours. As indicated in Tab. IV, we observe that the communities found by Louvain algorithm and IOLoCo have similar performance in their

Algorithm	NMI	
	$W = 7$ days	$W = 30$ days
Louvain	0.1606 ± 0.0471	0.1634 ± 0.0434
IOLoCo	0.1131 ± 0.0458	0.1273 ± 0.0362

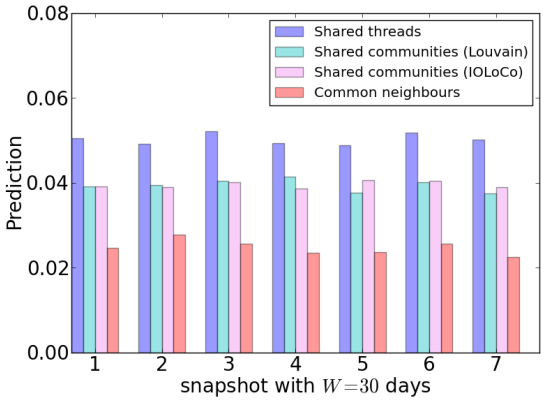
TABLE III: The average NMI obtained by different methods on Debian Dataset with different time windows

Algorithm	Prediction	
	$W = 7$ days	$W = 30$ days
Louvain	0.025 ± 0.0018	0.038 ± 0.0017
IOLoCo	0.022 ± 0.0020	0.040 ± 0.0007

TABLE IV: The average prediction obtained by different methods on Debian Dataset with different time windows



(a)



(b)

Fig. 6: Prediction results for different methods on Debian Dataset with different time windows: (a) $W = 7$ days. (b) $W = 30$ days. Here the result is obtained by running each method 100 times.

predictions. The number of shared communities provide better performance than that of CN but not as good as the number of shared threads. Moreover, the performance of the predictor that uses the number of shared communities is greatly improved when increasing the length of time window from $W = 7$ days to $W = 30$ days.

We can thus conclude that although the detected communities are different from the structure of threads, the predictor by using the number of shared communities also has superior predictive power than traditional predictors such as common neighbours predictor.

V. Discussion

Our results are obtained by analysing an email network, which is a temporal social network. From its available temporal topology, we can known when the links between pairs of individuals occur, crucial for revealing network dynamics. In addition, there exist known particular structure called thread in the email network. These threads allow to study the specific patterns of temporal movement of individuals from one thread to another. Such valuable information can be used to predict new social ties for users who do not have any connection.

Here, we are just beginning to investigate social networks by using link stream. For example, by using sub-streams to describe threads, we observe the different performances between thread duration distribution and thread size distribution. The more detailed work is needed and may provide insight into the connection between individuals and temporal contact patterns. Moreover, we transform a link stream to a line graph and use the algorithm which uncovers the community structure of a static graph to discover the link communities. It shows that some metrics and methods for static graphs can be used in link stream directly. From this point, link stream might be a powerful tool to study the temporal and topological properties of social networks. Predicting future links by using link stream might be important for dynamical individuals.

The applications above show that some particular temporal structures such as threads and communities can be used to predict future links between individuals. Both temporal community membership and temporal thread membership reveal the likelihood between individuals. This information is crucial for further investigation of temporal null-model construction. Investigating more temporal and topological features of social networks will help to derive a clear and concise summary of a network's temporal structure, to model network dynamics and accurately predict missing and future connections in a wide variety of situations.

VI. Conclusion

In this paper, we explore the connection between link prediction and temporal topology. We use the link stream to describe the temporal topology of social networks and employ particular temporal structures such as threads and communities to address link-prediction problem. In this study, we focus on studying the performance of the number of shared threads in predicting future links. Using the Debian Mailing Lists, we have found that the structure of a thread does not match the traditional definition of community. Another important conclusion is that the number of shared communities is a good

link predictor in the email network but not as good as the number of shared threads.

There are many future work. We are primarily interested in studying and characterizing threads. The structure of threads shows strong implicit relationship between users in email network. We only analysed the 1-mode network whose nodes represent individuals. If we use a 2-mode network whose nodes are distinguished into two classes such as regular individuals and irregular individuals, we may obtain more information about thread structure. Understanding the structure of threads will help us to model the evolution of social networks

In addition, we are interested in identifying link communities to tackle link-prediction problem. From our studies on email network, we know that the number of shared link communities can be used to predict future links. But we only considered the unweighted line graph. A weighted line graph may improve the performance of community detection methods and ameliorate the predictive power of the number of shared detected link communities.

We are also interested in identifying particular structures different from communities, such as thread-like structure. Using the number of shared (detected) threads to predict future social ties between entities in different fields such as sociology, biology, economics, information science and computer science, would be interesting.

REFERENCES

[1] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[2] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: Workshop on Link Analysis, Counter-terrorism and Security*, 2006.

[3] Mohammad Al Hasan and Mohammed J Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

[4] Réka Albert and Albert-László Barabási. Topology of evolving networks: local events and universality. *Physical review letters*, 85(24):5234, 2000.

[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[6] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308, 2006.

[7] Aaron Clauset and Nathan Eagle. Persistence and periodicity in a dynamic proximity network. *arXiv preprint arXiv:1211.7343*, 2012.

[8] Rémi Dorat, Matthieu Latapy, Bernard Conein, and Nicolas Auray. Multi-level analysis of an interaction network between individuals in a mailing-list. In *Annales des télécommunications*, volume 62, pages 325–349. Springer, 2007.

[9] TS Evans and R Lambiotte. Line graphs, link partitions, and overlapping communities. *Physical Review E*, 80(1):016105, 2009.

[10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[11] Lisa Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *The Journal of Machine Learning Research*, 3:679–707, 2003.

[12] Krzysztof Juszczyszyn, Katarzyna Musial, and Marcin Budka. Link prediction based on subgraph evolution in dynamic social networks. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 27–34. IEEE, 2011.

[13] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

[14] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[15] Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, and Santo Fortunato. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011.

[16] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[17] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[18] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[19] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[20] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[21] Blaise Ngonmang, Maurice Tchuente, and Emmanuel Viennet. Local community identification in social networks. *Parallel Processing Letters*, 22(01), 2012.

[22] Jukka-Pekka Onnela, Samuel Arbesman, Marta C González, Albert-László Barabási, and Nicholas A Christakis. Geographic constraints on social network groups. *PLoS one*, 6(4):e16939, 2011.

[23] Walter W Powell, Douglas R White, Kenneth W Koput, and Jason Owen-Smith. Network dynamics and field evolution: The growth of interorganizational collaboration in the life sciences1. *American journal of sociology*, 110(4):1132–1205, 2005.

[24] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1):016110, 2006.

[25] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1046–1054. ACM, 2011.

[26] Han Hee Song, Tae Won Cho, Vacha Dave, Yin Zhang, and Lili Qiu. Scalable proximity estimation and link prediction in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 322–335. ACM, 2009.

[27] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. *Link prediction in relational data*. 2003.

[28] Tomasz Tylenda, Ralitsa Angelova, and Srikanta Bedathur. Towards time-aware link prediction in evolving social networks. In *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, page 9. ACM, 2009.

[29] Jordan Viard and Matthieu Latapy. Identifying roles in an ip network with temporal and structural density. *The Sixth IEEE International Workshop on Network Science for Communication Networks*, 2014.