LETTER

Community detection in temporal networks via a spreading process

View the article online for updates and enhancements.

Recent citations
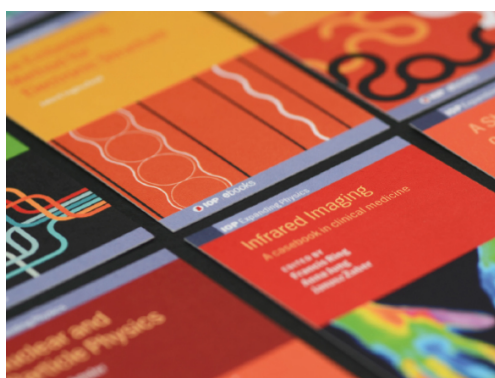
Community detection in temporal networks via a spreading process

PEICAN ZHU[1(a)], XIANGFENG DAI[2,3(a)], XUELONG LI[1,2(b)], CHAO GAO[4], MARKO JUSUP[5(c)] and ZHEN WANG[2,3(d)]

[1] *School of Computer Science and Engineering, Northwestern Polytechnical University - Xi'an, China*
[2] *Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University*
Xi'an, China
[3] *School of Mechanical Engineering, Northwestern Polytechnical University - Xi'an, China*
[4] *College of Information and Computer Science, Southwest University - Chongqing, China*
[5] *World Research Hub Initiative, Institute of Innovative Research, Tokyo Institute of Technology - Tokyo, Japan*

Abstract – Time-evolving relationships between entities in many complex systems are captured by temporal networks, wherein detecting the network components, *i.e.*, communities or subgraphs, is an important task. A vast majority of existing algorithms, however, treats temporal networks as a collection of snapshots, thus struggling with stability and continuity of detected communities. Inspired by an observation that similarly behaving agents tend to self-organise into the same cluster during epidemic spreading, we devised a novel community detection approach for temporal networks based on a susceptible-infectious-recovered-like (SIR-like) spreading process. Specifically, we used a Markov model of the spreading process to characterise each network node with a probability of getting infected, and subsequently recovering, when the infection starts from every other node in the network. This led to a similarity measure whereby nodes that easily infect one another are considered closer together. To account for network time evolution, we used communities from the preceding time step to modulate spreading in the current time step. Extensive simulations show that our technique outperforms several state-of-the-art methods in synthetic and real-world temporal networks alike.

Introduction. – Community or subgraph structure is ubiquitous in many complex systems, such as human contact networks, social networks, and technological networks [1–3]. Intuitively, a community's defining property is that nodes within the community are more tightly connected with each other compared to nodes outside the community. Operationalising this definition and subsequently identifying the community structure of a complex system often deepens the understanding of the system's functioning and improves the control of the system's dynamics. Consequently, community detection has become a topic of great importance and interest in network science [4–6]. Heretofore, however, most algorithms for the purpose of community detection focused on static

networks [7,8], whereas, in practice, system topology often evolves in time [9]. In social networks, for example, communities often merge or split because of the movement of individuals. Investigating community detection in such "temporal" networks is therefore an ongoing challenge for network scientists and a main concern of this manuscript.

One's first impulse in analysing communities in temporal networks may be to simply map the network into a succession of snapshots and handle each snapshot separately. This approach, however, quickly runs into problems because by ignoring evolutionary information, communities may drastically fluctuate from one moment to another. Such problems were first recognised in the context of clustering algorithms, leading to an approach dubbed *evolutionary clustering* [10] that inspired scores of related algorithmic solutions for networks [11–13]. The main focus of these solutions is network topology, yet little attention has been paid to the fact that dynamics in networks

---

(a)These authors contributed equally to this work.
(b)E-mail: li@nwpu.edu.cn
(c)E-mail: mjusup@gmail.com
(d)E-mail: zhenwang0@gmail.com

may be highly revealing of topology. Here, we exploit this fact by first using a dynamical process to extract structural information. Subsequently, we adjust the dynamics to reflect network topology from the previous time step while searching for new structural information in the current time step. We find that this time-evolving dynamical setup catches the community structure of temporal networks accurately.

Modern algorithms that incorporate dynamics to infer structure predominantly rely on random walks, whereby walkers move from one node to a randomly chosen neighbouring node in a single time step. Similarity between nodes is then measured in terms of distance between them as perceived by walkers [14]. Two nodes are close to one another if they can be reached with similar probability irrespective of the starting point in the network. Once distances between nodes are established, the problem of community detection turns into a clustering problem [14] because clustering algorithms typically try to minimise pairwise distances within clusters, while maximising distances between clusters. Herein, inspired by a recent work [15] in which a node's influence in a network is measured using a spreading process, we replaced random walks by a suitable spreading process that, in our case, quantifies mutual node similarity rather than a node's influence. We then assessed the performance of this spreading-based similarity measure for the purpose of community detection, especially in situations where network topology evolves in time.

Spreading dynamics in networks are arguably the main topic of network epidemiology [16], wherein recent efforts increasingly aim to combine the usual susceptible-infectious-recovered (SIR) paradigm with the aspects of human behaviour [17], such as awareness dynamics [18] and vaccination strategies [19,20]. We, by contrast, used the SIR paradigm to devise a similarity measure between network nodes, and thereafter turned this similarity measure into a foundation for a novel dynamic community detection technique. The co-evolution of spreading dynamics and topology, in particular, enabled robust community detection in temporal networks such that wild community fluctuations from one moment to another are avoided. We tested the proposed technique on both synthetic and real-world temporal networks. Computational results indicate that our SIR-based technique generally outperforms several state-of-the-art competitors for subgraph detection.

Preliminaries. – Given a set of time steps $\{1, 2, \ldots, T\}$, a temporal network is a set of sequential graphs, $\mathcal{G} = \{G^{[1]}, G^{[2]}, \ldots, G^{[T]}\}$, where $G^{[t]} = (V, E^{[t]})$ and $V = \{v_1, v_2, \ldots, v_N\}$ is a set of nodes, while $E^{[t]} \subseteq V \times V$ is a set of links between nodes at time t. An alternative way of representing $G^{[t]}$ is by means of an $N \times N$ adjacency matrix, $A^{[t]}$, where the matrix element $A_{ij}^{[t]} = 1$ if $(v_i, v_j) \in E^{[t]}$, and $A_{ij}^{[t]} = 0$ otherwise.

Given a temporal network, $\mathcal{G}$, the community structure $C^{[t]}$ at time t is formally a partitioning of the node set V, *i.e.*,

$$C^{[t]} = \left\{ C_1^{[t]}, C_2^{[t]}, \ldots, C_k^{[t]} \subseteq V \mid \right.$$
$$\left. C_i^{[t]} \cap C_j^{[t]} = \emptyset \wedge \bigcup_{i=1}^{k} C_i^{[t]} = V \right\},$$

such that, informally, nodes within a community are more tightly linked among themselves than with nodes outside the community. This informal part of community definition can be operationalised in various ways, but perhaps the commonest is via modularity Q as defined in [7]. We herein adopted the same approach, too.

Community detection technique. – Our novel technique for community detection in temporal networks comprises three components: i) a spreading process to quantify similarity between any pair of nodes in the network, ii) a greedy clustering algorithm to partition the network into communities, and iii) an extension to account for the network's temporal evolution in an effective manner. Hereafter, each of these three components is described in some detail.

Spreading. We adopted a traditional SIR model in which each individual might be in one of the following three states: susceptible S, infectious I, or recovered R. S individuals may get infected due to the physical contacts with their I neighbours, and this occurs with infection probability λ. Furthermore, I individuals may recover with probability μ. Given an initial number of I individuals, the spreading process has a natural end when there are no more I individuals left in the network. Our interest, however, is not necessarily in this final state, but rather states that best quantify node similarity.

A Markov model for the SIR spreading process is characterised by the following transition probabilities:

$$p_i^{[t]}(S, \tau+1) = p_i^{[t]}(S, \tau) \prod_{j=1}^{N} (1 - \lambda A_{ij}^{[t]} p_j^{[t]}(I, \tau)), \quad (1)$$

$$p_i^{[t]}(I, \tau+1) = p_i^{[t]}(S, \tau)$$
$$\times \left[1 - \prod_{j=1}^{N} (1 - \lambda A_{ij}^{[t]} p_j^{[t]}(I, \tau)) \right]$$
$$+ (1 - \mu) p_i(I, t), \quad (2)$$

$$p_i^{[t]}(R, \tau+1) = \mu p_i^{[t]}(I, \tau) + p_i^{[t]}(R, \tau), \quad (3)$$

where $p_i^{[t]}(s, \tau)$ denotes the probability that node v_i is in state $s \in \{S, I, R\}$ at time τ. We illustrate the meaning of these equations by interpreting the terms in eq. (2). Product $\lambda A_{ij}^{[t]} p_j^{[t]}(I, \tau)$ is the probability that node j will

infect node i, implying that the term in square brackets is the probability that any node in the network will infect node i. For a node to become infectious, it must be susceptible in the first place, which is given by probability $p_i^{[t]}(S, \tau)$. An already infectious node, will stay infectious with probability $1 - \mu$.

To characterise nodes based on the spreading process, we ran N simulations, where in each simulation different node v_j is selected as the origin of the infection, i.e., $p_j^{[t]}(I, 0) = 1$ and $p_i^{[t]}(I, 0) = 0$ for all $i \neq j$. This allowed us to construct row vector $\mathbf{p}_i^{[t]}$ for node v_i with N entries, where the j-th entry is $\mathbf{p}_i^{[t]}(j) = p_i^{[t]}(R, \mathcal{T})$, i.e., the probability obtained from eq. (3) after running the simulation of the spreading process for $\mathcal{T}$ steps with node v_j being the origin of the infection.

Clustering. Upon characterising every network node v_i with corresponding vector $\mathbf{p}_i^{[t]}$, we turned to clustering, first by collecting all the information into an $N \times N$ partitioning matrix, $P^{[t]}$, at time t whose elements are

$$P_{ij}^{[t]} = \begin{cases} \mathbf{p}_i^{[t]}(j), & i \neq j, \\ \dfrac{1}{N-1} \displaystyle\sum_{j \neq i} \mathbf{p}_i^{[t]}(j), & i = j. \end{cases} \quad (4)$$

Next we quantified the node similarity by means of the Pearson correlation

$$\mathrm{Cor}_{\mathrm{Pearson}}(v_i, v_j) = \frac{1}{N} \sum_{k=1}^{N} \frac{(P_{ik}^{[t]} - \mu_i^{[t]})(P_{jk}^{[t]} - \mu_j^{[t]})}{\sigma_i^{[t]} \sigma_j^{[t]}}, \quad (5)$$

where

$$\mu_i^{[t]} = \frac{1}{N} \sum_{k=1}^{N} P_{ik}^{[t]},$$

$$\sigma_i^{[t]} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (P_{ik}^{[t]} - \mu_i^{[t]})^2}.$$

Finally, we defined a distance function between nodes by

$$d(v_i, v_j) = 1 - \mathrm{Cor}_{\mathrm{Pearson}}(v_i, v_j). \quad (6)$$

Equipped with a distance function, we essentially turned the community detection problem into a clustering problem.

For clustering purposes, we employed the k-means algorithm [21]. One problem in this context was that the number of clusters (i.e., communities) that maximises modularity Q is unknown *a priori*. We therefore performed searches starting with $k^{[t]} = 2$ clusters, examining values $Q(k^{[t]} - 1)$, $Q(k^{[t]})$, and $Q(k^{[t]} + 1)$, and typically increasing $k^{[t]}$ until we found a local modularity maximum such that $Q(k^{[t]} - 1) \leq Q(k^{[t]})$ and $Q(k^{[t]}) \geq Q(k^{[t]} + 1)$. We termed this simple procedure the greedy k-means. Although it is certainly possible to search for the global modularity maximum because the number of communities is

limited by N, we found that a local modularity maximum represents a good compromise between the quality of community detection and computational performance.

Generalization to temporal networks. Aiming to apply our proposed method to the investigation of temporal networks, we started by calculating the first partitioning matrix, $P^{[1]}$, and then performing clustering as described above to obtain community structure $C^{[1]}$. If the consecutive realisations of a temporal network were independent, then we could treat the whole temporal network as a set of static networks, and proceed by finding any $P^{[t]}$ independently of any other $P^{[s]}$. Typically, however, network topology and properties vary gradually from one moment to another which is why we wanted the community structure from a previous time step to inform us about the current community structure. For $t > 1$, therefore, we took $C^{[t-1]}$ and $G^{[t]}$ as inputs and modified the spreading process to more likely infect nodes from within the same community than nodes between communities, i.e., if $v_i \in C_x^{[t-1]}$, $v_j \in C_y^{[t-1]}$, and $x \neq y$, then the probability of v_i infecting v_j and vice versa was determined by $\gamma \lambda$, $\gamma \leq 1$, instead of just λ, which is the within-community infection probability. The lower the value of γ, the stronger the influence of the previous community structure. With $\gamma = 1$ we could recover the above-mentioned limit of independent networks.

Numerical experiments. – We set off the discussion here by illustrating our technique on a static network (fig. 1). To this end, we chose a network with a total of 128 nodes, the degree of each node is 16, and the nodes comprise four communities, where the ratio of within-community and between-community links is 10:1. Our algorithm is clearly capable of detecting the four communities (fig. 1, right panel), but more important is the basis on which this detection rests (fig. 1, left panel). Namely, shown are four instances of vectors $\mathbf{p}_i$ (we suppressed the temporal index because the example is static), each of which corresponds to a node from a different community. When the spreading process originates from within-community (respectively, between-community) nodes, the probability of getting infected is high (respectively, low). Vectors $\mathbf{p}_i$ thus differently characterise nodes from different communities, and, therefore, serve as a solid basis for community detection.

Shifting the discussion to temporal networks, we compared the algorithm proposed herein with several state-of-the-art counterparts, i.e., FacetNet [11], DYNMOGA [13], and PisCES [22]. We ran numerical experiments on three synthetic benchmark datasets and five widely used real-world networks. The values of model parameters were $\lambda = 1/\langle d^{[t]} \rangle$, $\mu = 1$, and $\gamma = 2/3$, where $\langle d^{[t]} \rangle = \sum_{i=1}^{N} d_i^{[t]}/N$ was the average node degree at time t. We found that running 4–5 iterations of the system of eqs. (1)–(3) produces optimal results. In fact, we used five (respectively, four) iterations for all synthetic (respectively, real-world) networks.
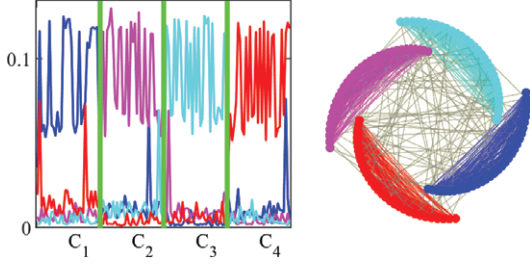
Fig. 1: Spreading process is a good basis for detecting communities in networks. The left panel exemplifies four row vectors $\mathbf{p}_i$ (temporal index suppressed) such that each row vector characterises a node from a different community. It is evident that the probability of getting infected, and then recovering, is high when the spreading process starts from within the community as opposed to out of the community. The right panel shows the final division of the network into four communities that closely correspond to the true network structure.
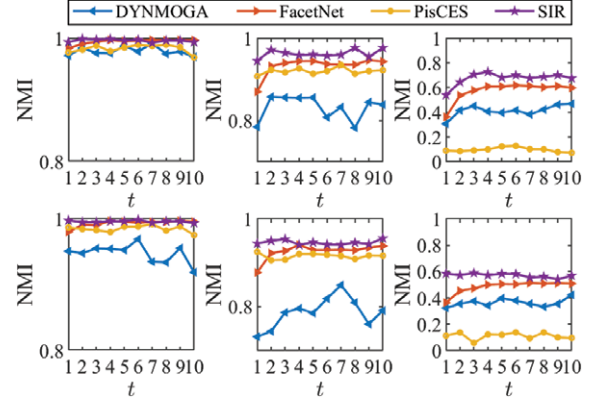

Fig. 2: SIR-based community detection outperforms other algorithms on Synthetic dataset #1. The result holds irrespective of the variability (upper row: $n_\mathrm{C} = 10\%$; lower row: $n_\mathrm{C} = 30\%$) and the fuzziness of the network structure (columns, left to right, $z = 2, 3$, and 4).

Evaluation standards. When the ground-truth community structure was known, we quantified the performance using the normalised mutual information, NMI, as defined in [23]. If the ground-truth community structure is $C = \{C_1, C_2, \ldots, C_k\}$, while the algorithm returns $C' = \{C'_1, C'_2, \ldots, C'_l\}$, then NMI is

$$\mathrm{NMI}(C, C') = \frac{2\mathrm{MI}(C, C')}{H(C) + H(C')}, \tag{7}$$

where $\mathrm{MI}(C, C')$ is the mutual information as defined in [24], while $H(C)$ and $H(C')$ are the entropies of partitions C and C', respectively. Quantity NMI attains values between zero and one, with the latter indicating a perfect reconstruction. In all tests performed with synthetic datasets, NMI values are the averages of 20 realisations; here, stochasticity stems from the time evolution of temporal networks and the application of the k-means algorithm. In the real-world networks, by contrast, the community structure is mostly unknown, disqualifying NMI as a performance measure. Instead, we used higher modularity Q, as defined in [7], to indicate a better partitioning into communities. We recorded the maximum Q over the course of 20 realisations; here, stochasticity stems solely from the application of the k-means algorithm.

Synthetic dataset #1. The first synthetic dataset we worked with is a variant of the classical benchmark proposed in [25] as adapted by [11]. This dataset is a generalisation of our static example (fig. 1), *i.e.*, the network consists of 128 nodes with an average degree equal to 16, where the nodes are divided into 4 communities with 32 nodes in each community. The number of between-community links, z, is a tunable parameter that modulates the fuzziness of the network structure. The network furthermore evolves in time over the course of $T = 10$ steps such that, in each time step, fraction n_C of randomly chosen nodes leaves their community to join another one. Parameter n_C is also tunable and controls the variability of

network structure. In numerical experiments, we tested $z = 2, 3$, and 4, and $n_\mathrm{C} = 10\%$ and 30%. Our SIR-based community detection outperforms other algorithms on Synthetic dataset #1 irrespective of network variability or fuzziness (fig. 2). Variability, in fact, seems to have a smaller effect on performance (cf. rows in fig. 2) than on fuzziness (cf. columns from left to right in fig. 2).

Synthetic dataset #2. The second synthetic dataset we worked with was proposed in [12] and comprised two types of temporal networks. The first type contains a fixed number of communities (hereafter SYN-FIX), whereas in the second type this number varies over time (hereafter SYN-VAR). SYN-FIX is in large part similar to Synthetic dataset #1, also totalling 128 nodes of average degree 16, divided into 4 communities with 32 nodes each. Our interest in this temporal network was twofold; first, as additional evidence that success on Synthetic dataset #1 is more than luck and, second, as an intermediate form between Synthetic dataset #1 and SYN-VAR. We ran simulations in SYN-FIX with $z = 6, 7$, and 8, while keeping $n_\mathrm{C} = 3/16$ fixed. SIR-based community detection again outperforms other competing algorithms (fig. 3, upper row). Even with $z = 8$, when half of all links are between-community, SIR-based detection is almost flawless.

With SYN-VAR, we introduced an additional challenge for community detection algorithms in the form of a changing number of communities over time. We started with 256 nodes divided into 4 communities containing 64 nodes each. To add communities, we randomly chose 8 nodes from each existing community and rewired them to form a new community. Accordingly, if the network contained, say, 4 communities in the previous time step, then the newly formed 5th community in the current time step would contain $8 \cdot 4 = 32$ nodes. Newly formed communities furthermore had a relaxation time of five time steps, after
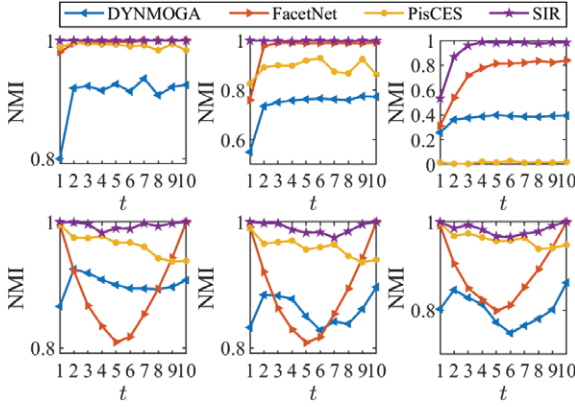
Fig. 3: SIR-based community detection outperforms other algorithms on Synthetic dataset #2. The result holds in both the SYN-FIX network with a fixed number of communities (upper row) and the SYN-VAR network with a changing number of communities (lower row). The fuzziness of network structure (columns, left to right, $z = 6, 7,$ and 8) is sufficient to heavily affect some algorithms in SYN-FIX, but our SIR-based technique remains largely unaffected. The same values of z create less fuzziness in SYN-VAR because here communities are mostly larger than in SYN-FIX.

which the nodes returned to their community of origin. The described procedure yielded 4, 5, 6, 7, 8, 8, 7, 6, 5, and 4 communities over the course of $T = 10$ time steps. The average node degree was set to half the node's community size. To add noise, at each time step, we randomly deleted 16 nodes from the network and then added 16 new ones in their place. We performed the SYN-VAR simulations with $z = 6, 7,$ and 8. Because here z was relatively small compared to the community size, all algorithms perform well (NMI > 0.7), but SIR-based community detection is, once again, a clear winner (fig. 3, lower row).

Synthetic dataset #3. The third and final synthetic dataset we worked with was introduced in [26]. The networks generated here consisted of 1000 nodes with a mean degree of 15 and a maximum degree of 50. The number of communities was between 20 and 50, while the percentage of between-community links was 60%. As the networks evolved over time, four types of events occurred:

– *Birth and death* involved i) rewiring randomly chosen nodes from anywhere in the network in such a way that these nodes ultimately formed three new communities, and ii) purposely rewiring nodes from three preexisting communities until these communities disappeared.

– *Merging and splitting* involved i) three instances of rewiring nodes from two previously separate communities until a new merged community was formed, and ii) three instances of purposely rewiring nodes from a single community in such a way that this community ultimately split into two new communities.
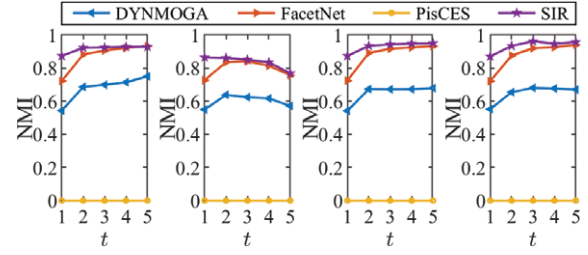

Fig. 4: SIR-based community detection outperforms other algorithms on Synthetic dataset #3. Competing algorithms exhibit very different performances, from PisCES that is almost completely ineffective in this test to FacetNet that is in later time steps almost as effective as our SIR-based community detection. Shown are the results for four types of events; from left to right, respectively, i) birth and death, ii) merging and splitting, iii) expansion and contraction, and iv) intermittency.

– *Expansion and contraction* involved i) rewiring randomly chosen nodes from anywhere in the network in such a way that three target communities increased their size by 25%, and ii) purposely rewiring nodes from three target communities until these communities reduced their size by 25%.

– *Intermittency* involved temporarily hiding three target communities by severing, and later restoring, links to their nodes.

All four types of events preserved the total number of nodes. We tested SIR-based community detection on four temporal networks over the course of $T = 5$ time steps. One network underwent events of one type. We once again found that SIR-based community detection outperforms its competitors, who display very different performance (fig. 4); from PisCES that is almost completely ineffective to FacetNet that is almost as effective as our technique as the time passes.

Real-world networks. We lastly analysed the performance of community detection algorithms in several real-world networks. These are High school [27], Hospital [28], Hypertext [29], and Rados [30] networks, as well as the temporal evolution of a Java program's source code structure as indicated by commits on the SVN[1]. Except for the Java network, the sampling interval for all these networks was one second, thus yielding large datasets that offered a microscopic perspective on how the networks had changed over time. For the purpose of testing the algorithms used in this study, we assumed a more macroscopic perspective by which a snapshot of High school, Hospital, and Hypertext (respectively, Rados) networks was taken every 12 hours (respectively, 30 days) in such a way that any two nodes (*e.g.*, v_i and v_j) were considered connected (*i.e.*, $A_{ij} = 1$) if they had been connected at any point in time between two consecutive snapshots. SIR-based community detection performs the best in four out of five

---

[1]`https://github.com/gephi/gephi/wiki/Datasets`.

Table 1: SIR-based community detection is the most consistent performer on real-world networks.

Dataset	Nodes	Links	Total time steps	Snapshot period	Total snaps	Time-averaged modularity Q			
						DYNMOGA	FacetNet	PisCES	SIR
High school	327	188508	363560	12 h	9	0.7049	0.7106	0.6272	**0.7129**
Hospital	75	32424	347500	12 h	9	0.1942	0.1776	0.0389	**0.1961**
Hypertext	113	20818	212340	12 h	5	0.3952	0.3634	0.0266	**0.3963**
Java	376	40915	129	–	66	0.6828	0.6471	0.0000	**0.7080**
Rados	167	82927	23430482	30 d	10	0.2134	**0.2571**	0.0759	0.2552

cases (table 1). Even with the Rados network, on which FacetNet performs the best, SIR-based community detection is a very close second. Our technique, therefore, is the most consistent performer that generates the best or nearly the best results in a wide variety of situations.

Conclusion. – We proposed a novel technique to detect communities in temporal networks by analysing an SIR-like spreading process that takes place on these networks. Intuitively, the idea was that two nodes from the same community more easily infect one another than two nodes from different communities do. Furthermore, we used the information on detected communities from the preceding time step to modulate spreading in the current time step, and thus avoid wild community fluctuations from one step to another. Numerical experiments on synthetic and real-world networks show that an SIR-like spreading process can serve as a good basis for defining a similarity measure between nodes, as well as for performing community detection based on such a measure.

$$* * *$$

REFERENCES

[1] NEWMAN M. E. J., *Nat. Phys.*, **8** (2012) 25.
[2] BU Z., LI H.-J., CAO J., WANG Z. and GAO G., *IEEE Trans. Cybern.*, **49** (2019) 328.
[3] LI H.-J., BU Z., LI A.-H., LIU Z.-D. and SHI Y., *IEEE Trans. Knowl. Data Eng.*, **28** (2016) 2349.
[4] QU Y.-F., SHI W.-R. and SHI X., *EPL*, **116** (2016) 28003.
[5] SUN B.-J., SHEN H.-W. and CHENG X.-Q., *EPL*, **108** (2014) 68001.
[6] ZHANG X.-S. *et al.*, *EPL*, **87** (2009) 38002.
[7] NEWMAN M. E. J. and GIRVAN M., *Phys. Rev. E*, **69** (2004) 026113.
[8] FORTUNATO S., *Phys. Rep.*, **486** (2010) 75.
[9] HOLME P. and SARAMÄKI J., *Phys. Rep.*, **519** (2012) 97.
[10] CHAKRABARTI D., KUMAR R. and TOMKINS A., *Evolutionary clustering*, in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM) 2006, p. 554.
[11] LIN Y.-R., CHI Y., ZHU S., SUNDARAM H. and TSENG B., *Facetnet: A framework for analyzing communities and their evolutions in dynamic networks*, in *Proceedings of the 17th International Conference on World Wide Web* (ACM) 2008, p. 685.
[12] KIM M. and HAN J., *Proc. VLDB Endow.*, **2** (2009) 622.
[13] FOLINO F. and PIZZUTI C., *IEEE Trans. Knowl. Data Eng.*, **26** (2014) 1838.
[14] PONS P. and LATAPY M., *Computing Communities in Large Networks Using Random Walks*, in *Computer and Information Sciences - ISCIS 2005, Lecture Notes in Computer Sciences*, Vol. **3733** (Springer, Berlin, Heidelberg) 2005, pp. 284–293.
[15] HU Y.-Q. *et al.*, *Proc. Natl. Acad. Sci. U.S.A.*, **115** (2018) 7468.
[16] LI H.-J. and WANG L., *New J. Phys.*, **21** (2019) 015005.
[17] WANG Z., ANDREWS M. A., WU Z.-X., WANG L. and BAUCH C. T., *Phys. Life Rev.*, **15** (2015) 1.
[18] GRANELL C., GÓMEZ S. and ARENAS A., *Phys. Rev. Lett.*, **111** (2013) 128701.
[19] LIU Y., DENG Y., JUSUP M. and WANG Z., *J. Theor. Biol.*, **400** (2015) 92.
[20] KUGA K., TANIMOTO J. and JUSUP M., *J. Theor. Biol.*, **469** (2019) 107.
[21] HUANG J., NIE F.-P. and HUANG H., *Spectral Rotation versus K-Means in Spectral Clustering*, in *Proceedings of the 27th AAAI Conference on Artificial Intelligence* (AAAI) 2013.
[22] LIU F.-C., CHOI D., XIE L. and ROEDER K., *Proc. Natl. Acad. Sci. U.S.A.*, **115** (2018) 927.
[23] DANON L., DIAZ-GUILERA A., DUCH J. and ARENAS A., *J. Stat. Mech.: Theory Exp.*, **2005** (2005) 09008.
[24] MACKAY D. J., *Information Theory, Inference and Learning Algorithms* (Cambridge University Press) 2003.
[25] GIRVAN M. and NEWMAN M. E. J., *Proc. Natl. Acad. Sci. U.S.A.*, **99** (2002) 7821.
[26] GREENE D., DOYLE D. and CUNNINGHAM P., *Tracking the Evolution of Communities in Dynamic Social Networks*, in *2010 International Conference on Advances in Social Networks Analysis and Mining* (IEEE) 2010, p. 176.
[27] MASTRANDREA R., FOURNET J. and BARRAT A., *PLOS ONE*, **10** (2015) 1.
[28] VANHEMS P. *et al.*, *PLOS ONE*, **8** (2013) 1.
[29] LORENZO I. *et al.*, *J. Theor. Biol.*, **271** (2011) 166.
[30] RYAN A. R. and NESREEN K. A., *The Network Data Repository with Interactive Graph Analytics and Visualization*, in *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (AAAI) 2015.