
Domain-Specific Language

- ▶ GUESS

Dual Scaling

- ▶ Correspondence Analysis

Dynamical Graphs

- ▶ Temporal Networks

Dynamic Communities

- ▶ Dynamic Community Detection

Dynamic Communities Analysis

- ▶ Models for Community Dynamics

Dynamic Community Detection

Rémy Cazabet¹ and Frédéric Amblard²

¹National Institute of Informatics (NII), Tokyo, Japan

²IRIT – Université Toulouse 1 Capitole, Toulouse Cedex 9, France

Synonyms

Community detection; Dynamic communities; Evolution of communities; Evolving communities; Evolving networks; Temporal networks

Glossary

Evolving Network A network that changes along time. Nodes and edges can be added to or removed from the network. In weighted networks, weights can also evolve

Snapshot of a Network A static network corresponding to all nodes and edges alive at a given time in an evolving network

Community In this context, a community corresponds to a structure of a network, composed of nodes densely connected together and more sparsely connected to the rest of the network

Definition

Temporal community detection is the process of finding the relevant communities corresponding to each step of evolution of a network that changes along time.

Introduction

Community detection is one of the most popular topics in the field of network analysis. Since the seminal paper of Girvan and Newman (2002), hundreds of papers have been published on the topic. From the initial problem of graph partitioning, in which each node of the network must belong to one and only one community, new aspects of community structures have been taken into consideration, such as overlapping communities and hierarchical decomposition. Recently, new methods have been proposed, which can handle temporal networks. The communities found by these algorithms are called dynamic communities.

Key Points

Definition of dynamic communities

Operations on dynamic communities

Different approaches for dynamic community detection

Temporal Community Detection

The Support of Temporal Communities: Networks Evolving Along Time

The first specificity of temporal community detection is that it is not applied on traditional static networks but on temporal ones. These networks, which evolve along time, can originate from many different domains. They can derive, for instance, from a daily, weekly, or monthly collection of data. For each of these data collections, a static network is created, and the evolution of the network follows naturally the combination of all these snapshots. Another way to obtain an evolving network consists in gathering information as soon as it appears, in real time. For example, in the large network composed of all Facebook users and their so-called friendship relations, we can add or remove a node each time a user creates or deletes his/her account and, similarly, create or remove an edge each time a user adds another one in his list of contacts. More formally, we associate to each node and each edge a sequence of intervals during which these entities are present. Networks modeled this way are called temporal networks (Holme and Saramäki 2012) and are now a very popular research topic.

However, it is possible to transform a sequence of snapshots into a temporal network and a temporal network into a sequence of snapshots; therefore, the process of detecting temporal communities is equivalent in both categories.

Defining Dynamic Communities

Temporal communities are simply communities that can change or evolve along time. To give a simple example of what it means, let's imagine a social network, and some communities identified on it. One of these communities corresponds to the players of a football team. The static community corresponds to all individuals playing in this team at a given time. But as time goes by, some players leave the team, while newcomers join in. After a long enough period, none of the initial players will still be in the team; however, the corresponding community still exists, without any interruption.

While static communities are simply defined as sets of nodes and edges among these nodes, defining dynamic communities is a bit more complex.

As networks can be modeled either by sequences of snapshots or by temporal networks, a temporal community can be seen in two different ways:

- As a sequence of static communities
- As an initial static community and a sequence of modifications on this community, namely, nodes integration and nodes exclusion

As for temporal networks, these two representations are in fact semantically equivalent.

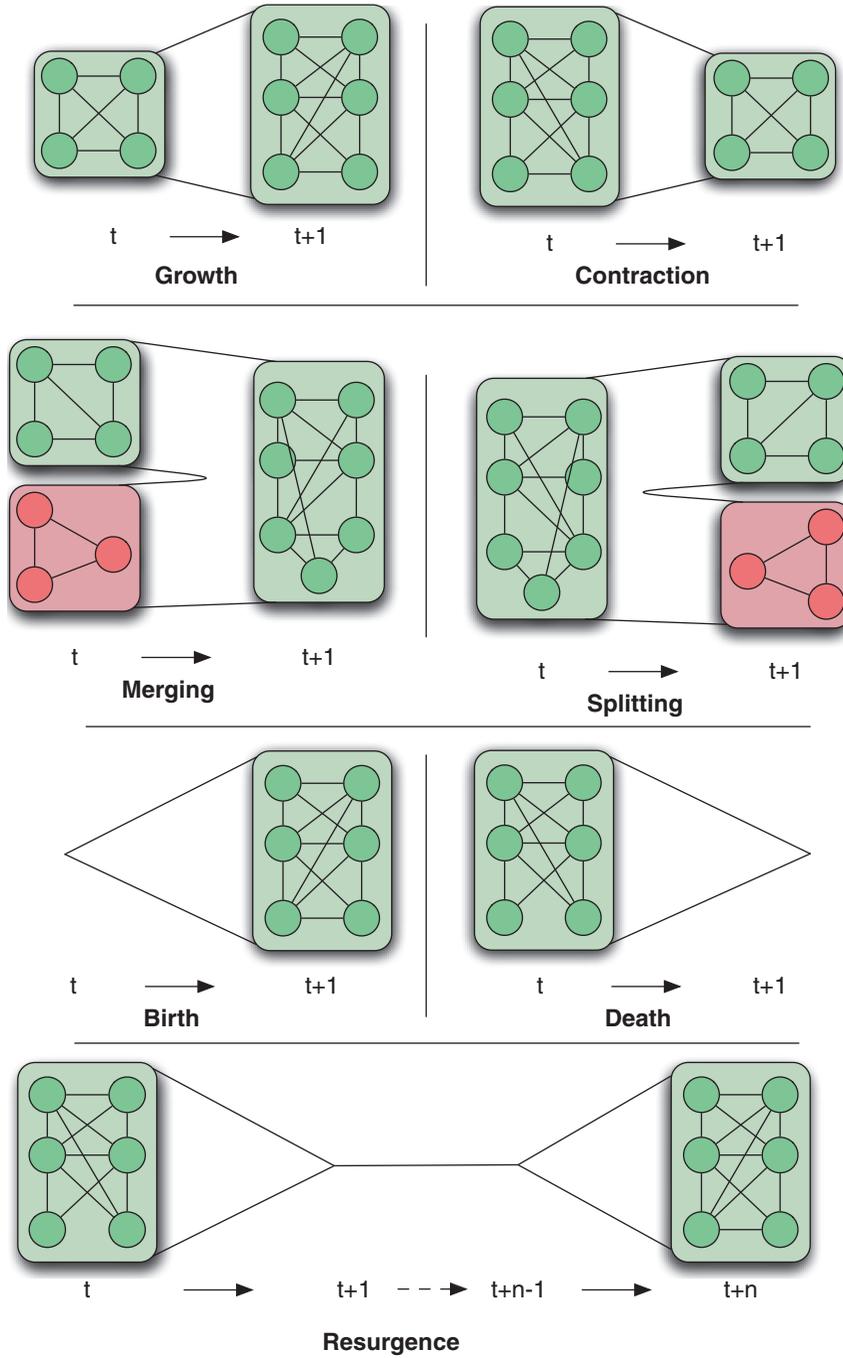
Operations on Dynamic Communities

In an evolving network, independently from its representation, we can define straightforward operations: node apparition, node disappearance, edge apparition, and edge disappearance.

Operations on temporal communities are more complex. They were first introduced in the paper of Palla et al. (2007), which listed six different operations on communities. In the work of Cazabet et al. (2012), a seventh operation is identified.

These operations, illustrated in Fig. 1, are the following:

- Growth: a community can grow by integrating new nodes.
- Contraction: a community can contract by rejecting some of its nodes.
- Merging: two communities or more can merge into a single one.
- Splitting: one community can split into two or more communities.
- Birth: a new community can appear at a given time, composed of any number of nodes.
- Death: a community can vanish at any time. All nodes that were belonging to this community loose this membership.
- Resurgence: a community can only disappear for a period and come back after this period as if it has never stopped. For example, in the case of the soccer team, let's imagine that they stop to play together during the summer vacations, while a snapshot of the network is taken every month. The community



Dynamic Community Detection, Fig. 1 Possible operations on communities

will be present in the snapshots of June and September, and all other months to the exception of the snapshots corresponding to July and August. This can be modeled as a resurgence of the community after two months.

All of these operations, however, cannot be detected by any of the existing algorithms. Growth and contraction are always possible; they are the basic operations necessary for any temporal community. Most algorithms are also able to detect birth and death of communities. But merging and splitting are the more complex operations, and only few methods proposed for the detection of dynamic communities include such operations. In fact, their semantic is not always clear, depending on the application domain, and there are several ways to handle this problem.

In particular, when two communities must be merged, several options are possible. Let's assume that A and B are two communities, which became sufficiently similar to be merged. For the purpose of illustration, we consider two nodes i and j , members of A but not members of B .

Several options are possible to merge A and B :

- Consider that one of the communities no longer exists. In this case, it is more precisely a mechanism of absorption of one community by another. Which community will survive is a problem in itself. Among possible choices, we can cite:
 - Keep the oldest/youngest community.
 - Keep the biggest/smallest community.
 - Remove the one with the highest percentage of nodes included in the other.
- Make the two communities A and B die and create a new community C from the nodes of A and B . This solution rises the question of the continuity of the evolution of a community, which can become hard to follow.

Regardless of the solution chosen, another problem must be solved: what must be done with the nodes that do not belong to the intersection of A and B , namely, i and j , in this example?

There are again several options:

- Integrate (or not) these nodes in the community resulting from the merging.
- Evaluate for each node if he must become a member of the resulting community.

- Keep them with their community if it is the one which is kept (A); do not integrate them if it is B .

Different Approaches to the Problem

Now that we have clarified what dynamic community detection is, we will present the different approaches we can take to obtain these communities as well as most of the algorithms available so far in the literature.

Independent Community Detections and Matching

The first approach taken was to work with sequences of snapshots and was composed of two steps:

1. Detect static communities on each snapshot independently.
2. For each snapshot, match the communities detected with the communities found on the previous one.

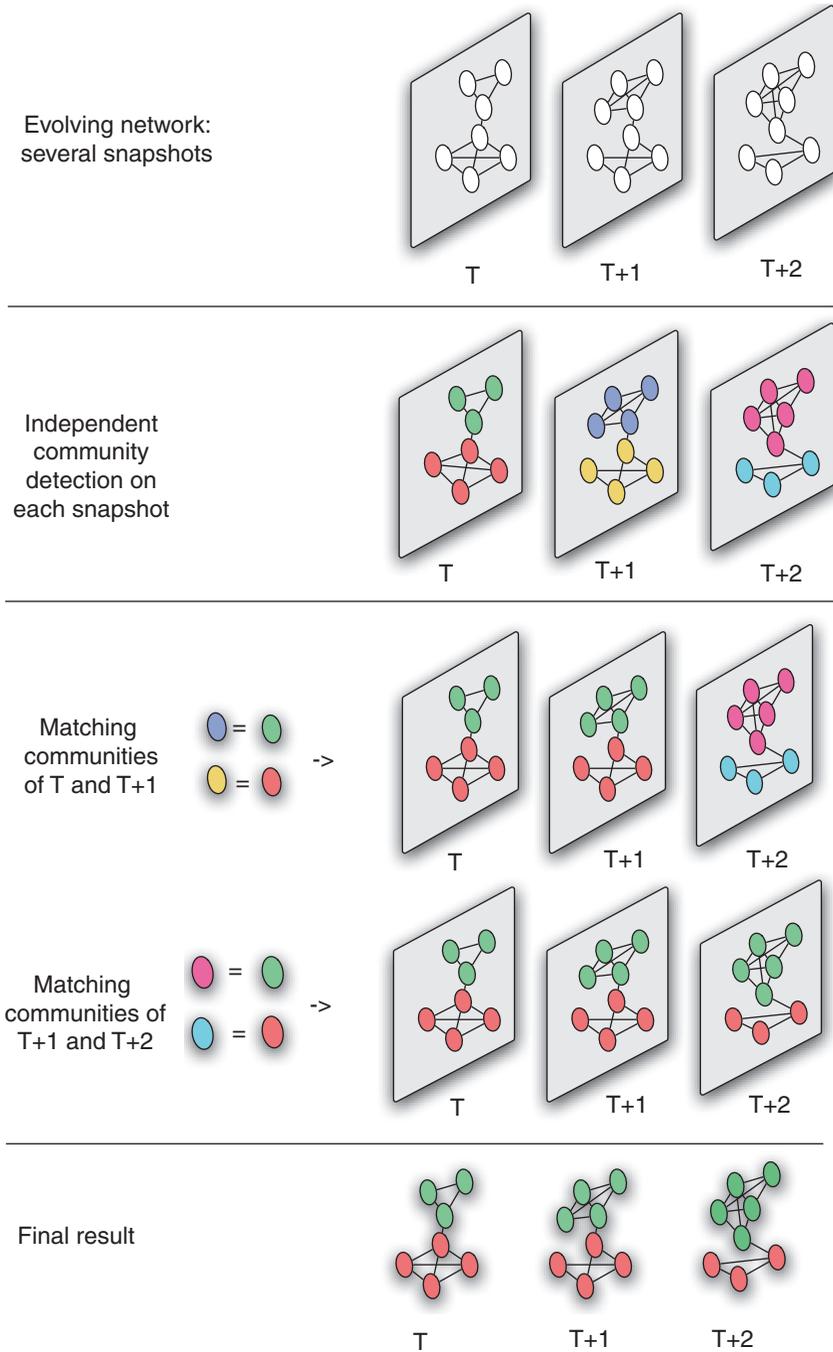
This process is illustrated in Fig. 2.

The earlier methods proposed were using this approach. However, it is losing popularity due to instability problems, explained below. This approach is notably used by Hopcroft et al. (2004), Palla et al. (2007), Wang et al. (2008), Rosvall and Bergstrom (2010), Chen et al. (2010), and Greene et al. (2010).

Advantages The main advantage of this solution is to offer the possibility to reuse traditional community detection techniques without the necessity to modify them. Matching sets is also a classic problem, for which methods already exist. Finally, the time-consuming process of community detection on all snapshots can be parallelized, speeding it up tremendously.

Drawbacks There is a major drawback with this solution: community detection algorithms are unstable. This means that for two similar networks with tiny modifications in between, the algorithm can provide very different results. In fact, most of the efficient community detection algorithms are stochastic, and two runs on the same network do not necessarily provide the same solution.

As a result, when we observe modifications between two snapshots, we cannot know if they correspond to real structural modifications of the



Dynamic Community Detection, Fig. 2 Temporal community detection by independent static community detections and matching

communities or are simply artifacts of the static community detection algorithm used.

Some solutions tried to improve this issue by considering core community, defined as the most stable part of the community, ignoring nodes that frequently change membership.

Nevertheless, this problem stays very important, and all other approaches try to fix it.

Informed Iterative Community Detections

This approach is still using static community detections, but, this time, snapshots are processed in their natural order, and, for each of them, the preceding result is used in order to stabilize the process. The process follows three steps:

1. Detect static communities on the first snapshot.
2. Detect communities on snapshot $t + 1$ using network at $t + 1$ and communities at time t .
3. As long as all snapshots have not been processed, go back to step 2.

This process is illustrated in Fig. 3.

There are several ways to detect communities that are coherent with the ones detected on the previous snapshots, the two most popular are:

- Initialize the algorithm used at $t + 1$ with the communities at t .
- Create a metric that tries to optimize two objectives: the maximization of the quality of communities at $t + 1$ and the maximization of the likeliness with communities at t .

This approach is notably used by Chakrabarti et al. (2006), Chan et al. (2009), Wang et al. (2008), Xu et al. (2011), Lin et al. (2009), and Lancichinetti et al. (2009).

Advantages This method allows to cope with the instability problem, while staying relatively close to the usual community detection problem.

Drawbacks Traditional community detection methods are no longer directly applicable. It is not possible to parallelize community detection on different snapshots; therefore these methods can be very slow when applied to large networks with many steps of evolution.

Finally, the quality of the stabilization rests only upon the algorithm proposed and might not always be sufficient.

Global Community Detection on All Snapshots

This approach still uses sequences of snapshots, but, this time, all of them are studied simultaneously. A single community detection is run, creating a community which contains not only different nodes from the same snapshot but also nodes from different snapshots.

This process is illustrated in Fig. 4.

There are at least two ways of doing this global community detection:

- One can create a unique network, where nodes are all instances of all nodes in all snapshots and edges can either be usual edges between snapshots or a new type of edges, linking nodes in between different snapshots. Hereafter, on this large network, we can run a community detection algorithm.
- Another solution is to design a metric that can directly be optimized on several snapshots.

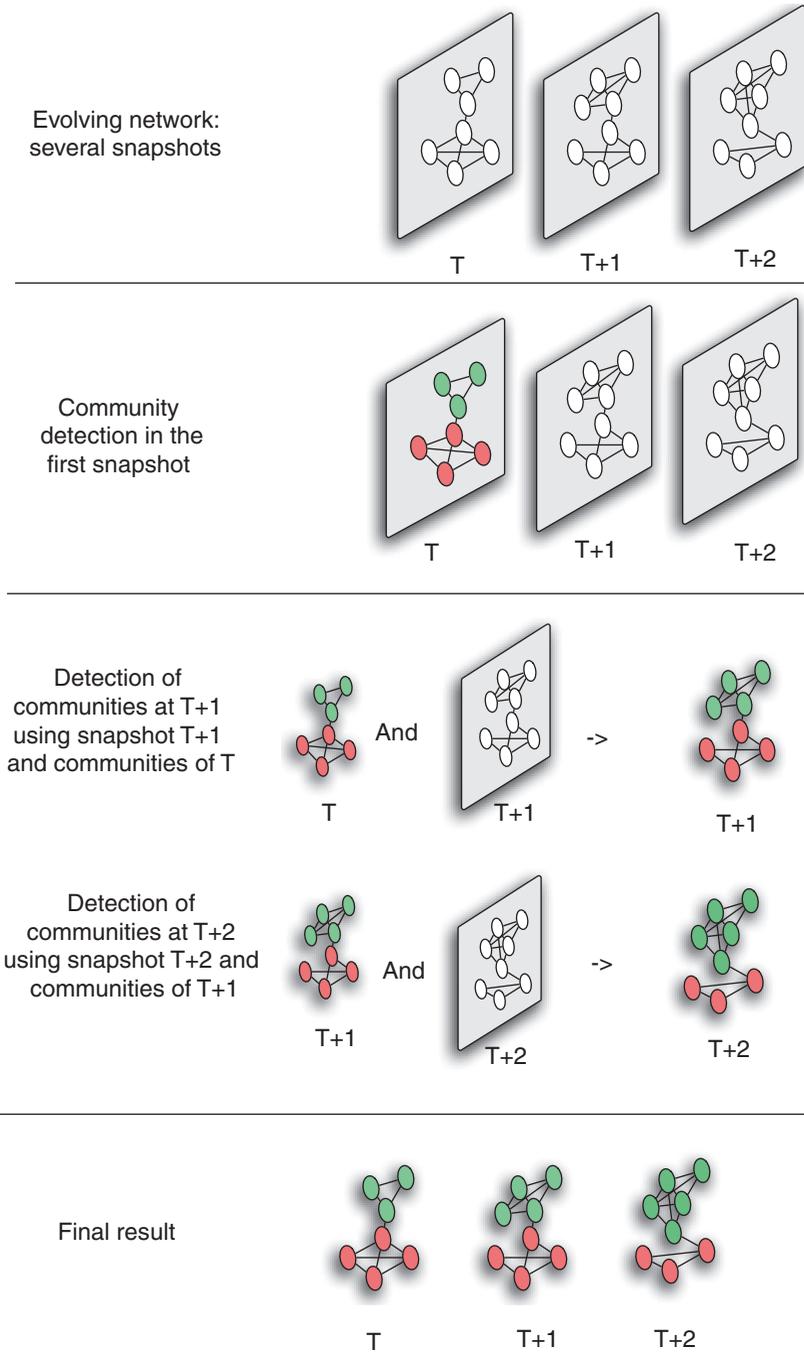
This approach is notably used in Tantipathananandh et al. (2007), Jdida et al. (2007), Mucha et al. (2010), Aynaud and Guillaume (2010), and Yang et al. (2011).

Advantages The main advantage of this solution is to eradicate the problem of instability. We ensure that detected communities are coherent in the long run.

Drawbacks The computational cost of this solution can be high, when applied to a network with many snapshots. Indeed, if only one community detection is attempted, the quantity of data to deal with is equal to the number of snapshots multiplied by the average size of each snapshot.

The other limitation is the difficulty to handle operations on communities such as merging and splitting. While there is no structural impossibility to detect them, none of the existing algorithms proposed using this approach propose to find them.

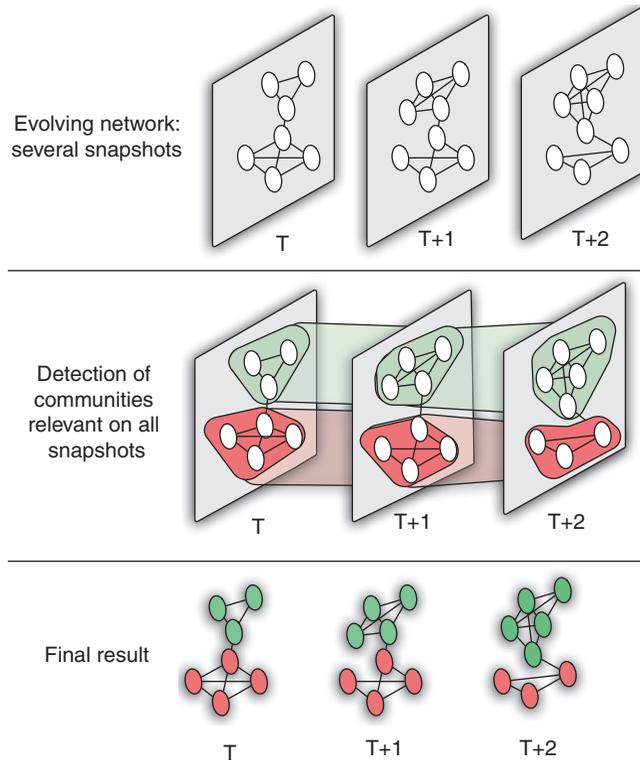
Finally, with this method, it is not possible to ensure coherent temporal community detection on a network evolving in real time. As the solution is not iterative, if we obtain a new snapshot, we cannot keep the previous result and just update the evolution of communities with the last data. This method is only applicable to temporal networks that will not change anymore.



Dynamic Community Detection, Fig. 3 Temporal community detection by informed iterative static community detections

Dynamic Community Detection, Fig. 4

Temporal community detection by global detection on all snapshots



Dynamic Community Detection on Temporal Networks

The last approach is original by the fact that it does not process sequences of snapshots, but rather a temporal network. As a consequence, knowing the network and the communities at time $t - 1$, these algorithms, to find the communities at time t , do not consider the snapshot t but only the modifications of the network between $t - 1$ and t .

The process can be described in three steps:

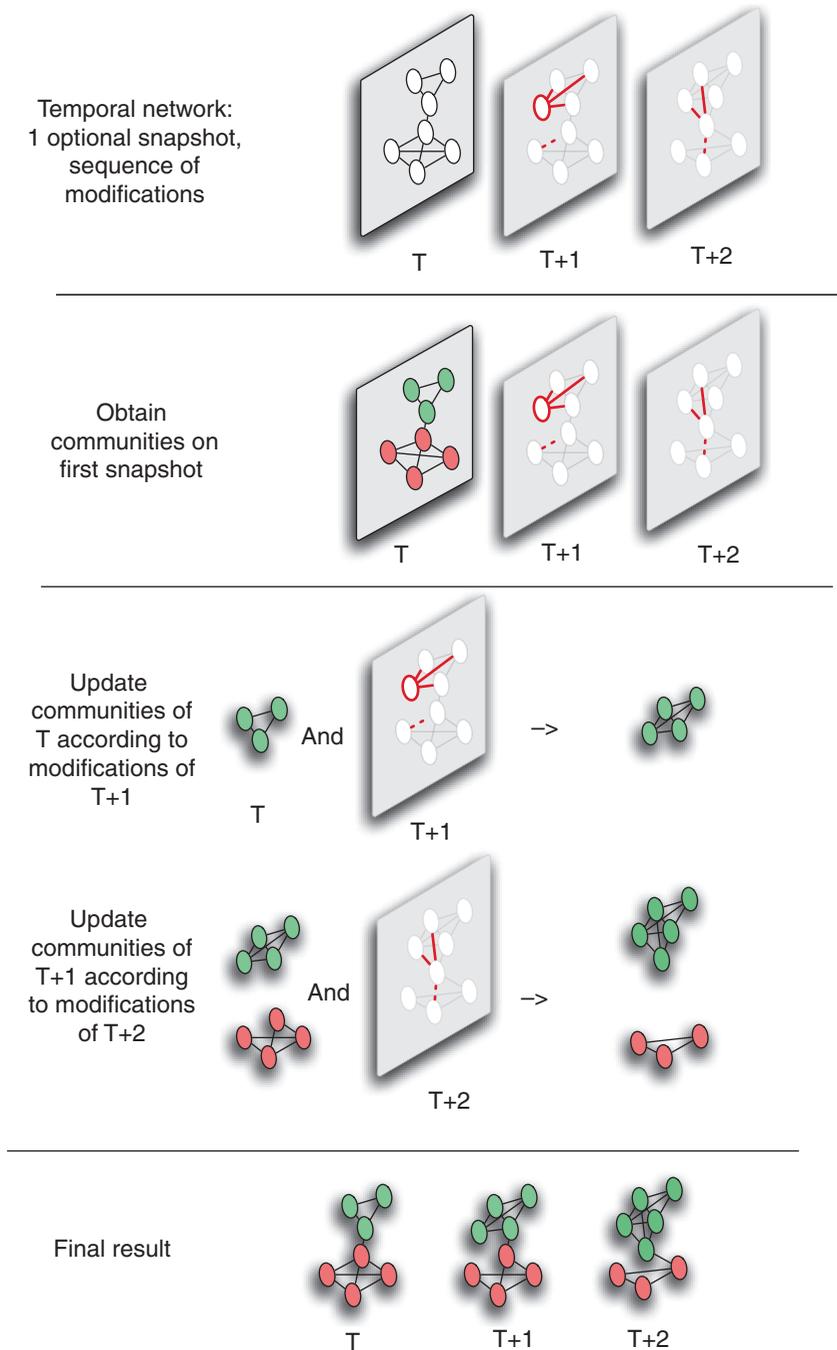
1. Detect communities in the initial state of the network
2. From communities at t and modifications of the network producing $t + 1$, deduce modifications of the communities at $t + 1$.
3. As long as all steps of evolution have not been processed, go back to step 2.

This process is illustrated in Fig. 5.

This approach is notably used by Falkowski et al. (2008), Cazabet et al. (2010), Cazabet and Amblard (2011), Li et al. (2012), and Shang et al. (2012).

Advantages As communities from t are modifications of existing communities at time $t - 1$, on the short term, the stability is ensured. Another advantage is the complexity: as modifications only concern the changes in the network between snapshots, if snapshots are topologically close to each other, processing communities at $t + 1$ knowing communities at t can be very fast. This property makes this approach very suitable for detecting communities in networks for which we have a large number of steps of evolution.

Drawbacks The main drawback of this solution is the difficulty to ensure a long-term coherence of the dynamic communities. As a complete community detection on a whole static network is never run after the first step, it is harder to certify that the communities found are relevant at the global level. We can also note that this process is quite different from most static solutions, and therefore classic static methods cannot be reused easily.



Dynamic Community Detection, Fig. 5 Dynamic community detection using a temporal network

Conclusion

In conclusion, we have seen that temporal community detection, while issued from static community detection, asks several new questions. Operations on communities are a totally new concept and are not a straightforward problem. Furthermore, we have seen that many approaches can be used to deal with the problem, all of them having strengths and weaknesses.

Key Applications

Temporal community detection, as its static counterpart, has a lot of possible applications. However, while there is nowadays a tremendous quantity of easily accessible static networks, ranging from small social networks to large networks issued from the web, finding temporal networks with more than a handful of steps of evolution stays rare. These networks are though very common, but the process of data collection is not as simple as for static ones.

Nevertheless, applying these methods has produced some interesting results.

Citation and co-authorship networks are a popular topic of application, such as in Rosvall and Bergstrom (2010), Palla et al. (2007), and Jdida et al. (2007). Rosvall and Bergstrom (2010) is especially interesting, as it illustrates the growth of some scientific fields, and can even identify some fusions, as, for example, the creation of the scientific field of neuroscience from neurology, psychology, and some part of molecular and cell biology.

Communication datasets is another topic for which networks can be easily accessible. The Enron mail dataset has been studied in many publications (e.g., Falkowski et al. (2008); Chen et al. (2010)), as it is a relatively small network for which we have access to the complete evolution. Correlation can be seen between communities' birth and death and events happening at Enron, such as employees leaving the company.

Finally, many datasets from the Internet or the web can be studied. For example, a network of multicast routers in Aynaud and Guillaume (2010), blogs networks in

Chakrabarti et al. (2006), and a video sharing network in Cazabet et al. (2012). In this former paper, the analysis is done on more than two years, not with snapshots but on the temporal network, resulting in more than 50,000 steps of evolution. The communities detected correspond to popular topics on the platform, and different categories of events are identified, notably short events corresponding to the release of a new movie or video game and long-lasting events, corresponding to more general topics such as jazz, soccer, and cat videos.

Future Directions

Temporal community detection is still a very young field of research, which will probably evolve strongly in a near future. As we are still in a prospecting phase, we will probably continue to see new methods proposed.

But it is in the domain related to temporal community detection that most work is to be done.

It is, for example, a necessity to be able to compare different solutions between them. As the introduction of the LFR benchmark (Lancichinetti et al. 2008) allowed to compare static community detection algorithms, the conception of a unified method to compare dynamic communities is necessary. But such a work requires first to explore and study the properties of temporal communities.

Another domain, which will probably be explored more deeply, is the processing and visualization of the results. Obtaining temporal communities is a first step, but being able to use this information in order to get new insights into the studied networks is of course the final goal.

Cross-References

- ▶ Analysis and Mining of Tags, (Micro)Blogs, and Virtual Communities
- ▶ Analysis and Visualization of Dynamic Networks

- ▶ Community Detection, Current and Future Research Trends
- ▶ Community Evolution
- ▶ Community Identification in Dynamic and Complex Networks
- ▶ Models for Community Dynamics

References

- Aynaud T, Guillaume JL (2010) Long range community detection. In: LAWDN- Latin-American workshop on dynamic networks, Buenos Aires
- Cazabet R, Amblard F, Hanachi C (2010) Detection of overlapping communities in dynamical social networks. In: IEEE second international conference on social computing (SocialCom), Minneapolis, pp 309–314
- Cazabet R, Amblard F (2011) Simulate to detect: a multi-agent system for community detection. In: IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (WI-IAT), Lyon, vol 2. IEEE, pp 402–408
- Cazabet R, Takeda H, Hamasaki M, Amblard F (2012) Using dynamic community detection to identify trends in user-generated content. *Soc Netw Anal Min* 2: 361–371
- Chakrabarti D, Kumar R, Tomkins A (2006) Evolutionary clustering. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia. ACM, pp 554–560
- Chan SY, Hui P, Xu K (2009) Community detection of time-varying mobile social networks. *Complex Sci* 4:1154–1159
- Chen W, Liu Z, Sun X, Wang Y (2010) A game-theoretic framework to identify overlapping communities in social networks. *Data Min Knowl Discov* 21(2): 224–240
- Falkowski T, Barth A, Spiliopoulou M (2008) Studying community dynamics with an incremental graph mining algorithm. In: AMCIS 2008 proceedings, Toronto, p 29
- Girvan M, Newman MEJ (2002) Community structure in social and biological networks. *Proc Natl Acad Sci* 99(12):7821–7826
- Greene D, Doyle D, Cunningham P (2010) Tracking the evolution of communities in dynamic social networks. In: International conference on advances in social networks analysis and mining (ASONAM), Odense. IEEE, pp 176–183
- Holme P, Saramäki J (2012) Temporal networks. In: Physics reports. Elsevier, Amsterdam
- Hopcroft J, Khan O, Kulis B, Selman B (2004) Tracking evolving communities in large linked networks. In: Proc Natl Acad Sci U S A 101(1): 5249–5253
- Jdidia MB, Robardet C, Fleury E (2007) Communities detection and analysis of their dynamics in collaborative networks. In: Second international conference on digital information management, ICDIM'07, Lyon, vol 2, pp 744–749
- Lancichinetti A, Fortunato S, Kertesz J (2009) Detecting the overlapping and hierarchical community structure in complex networks. *New J Phys* 11(3): 033015
- Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E* 78(4):046110
- Li J, Huang L, Bai T, Wang Z, Chen H (2012) Cdbia: a dynamic community detection method based on incremental analysis. In: International conference on systems and informatics (ICSAI), Yantai. IEEE, pp 2224–2228
- Lin YR, Chi Y, Zhu S, Sundaram H, Tseng BL (2009) Analyzing communities and their evolutions in dynamic social networks. *ACM Trans Knowl Discov Data (TKDD)* 3(2):8
- Mucha PJ, Richardson T, Macon K, Porter MA, Onnela JP (2010) Community structure in time-dependent, multiscale, and multiplex networks. *Science* 328(5980):876–878
- Palla G, Barabasi AL, Vicsek T (2007) Quantifying social group evolution. *Nature* 446(7136):664–667
- Rosvall M, Bergstrom CT (2010) Mapping change in large networks. *PLoS one* 5(1):e8694
- Shang J, Liu L, Xie F, Chen Z, Miao J, Fang X, Wu C (2012) A real-time detecting algorithm for tracking community structure of dynamic networks. In: SNAKDD workshop, Beijing
- Tantipathananandh C, Berger-Wolf T, Kempe D (2007) A framework for community identification in dynamic social networks. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose. ACM, pp 717–726
- Wang Y, Wu B, Du N (2008) Community evolution of social network: feature, algorithm and model. *Sci Technol*, arXiv:0804.4356
- Xu K, Kliger M, Hero A (2011) Tracking communities in dynamic social networks. In: Social computing, behavioral-cultural modeling and prediction, College Park, pp 219–226
- Yang T, Chi Y, Zhu S, Gong Y, Jin R (2011) Detecting communities and their evolutions in dynamic social networks? A bayesian approach. *Mach Learn* 82(2):157–189