

# D

## Dynamic Community Detection

Rémy Cazabet<sup>1</sup>, Giulio Rossetti<sup>2</sup> and  
Frédéric Amblard<sup>3</sup>

<sup>1</sup>LIP6, CNRS, Pierre and Marie Curie University,  
Paris, France

<sup>2</sup>KDD Lab, University of Pisa, Pisa, Italy

<sup>3</sup>IRIT – Université Toulouse 1 Capitole,  
Toulouse, France

## Synonyms

[Community detection](#); [Dynamic communities](#);  
[Evolution of communities](#); [Evolving communi-  
ties](#); [Evolving networks](#); [Temporal networks](#)

## Glossary

Community	In this context, a community corresponds to a subgraph of a network, composed of nodes densely connected together and more sparsely connected to the rest of the network
Evolving network	A network that changes along time. Nodes and edges can be added to or removed from the

Snapshot of a network	In weighted networks, weights can also evolve A static network corresponding to all nodes and edges alive at a given time in an evolving network
-----------------------	---

## Definition

Dynamic community detection is the process of finding relevant communities in 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 dynamic 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

## Dynamic Community Detection

### The Support of Dynamic Communities: Networks Evolving Along Time

The first specificity of dynamic community detection is that it is not applied on traditional static networks but on dynamic ones. These networks, which evolve along time, can originate from many different domains. They can derive, for instance, from a daily, weekly, or monthly snapshot of the current state of the network or an aggregation of all observed interactions on such a period. In both cases, the evolution is characterized by an ordered sequence of static networks. 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 Saramaki 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, at worst at the cost of a discretization in the case of continuous temporal networks. Therefore, the nature of dynamic community detection is independent from the representation of evolution.

### Defining Dynamic Communities

Dynamic communities are communities that can change or evolve over time. To give a simple example of what it means, let us 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.

Static communities are defined as sets of nodes. We define dynamic communities as follows:

*Definition (Dynamic Community).* Given a dynamic network  $G$ , a Dynamic Community (DC) is defined as a set of pairs (nodes, periods).

Note that periods can be discrete or continuous, depending on the adopted representation, without changing the nature of dynamic communities.

### 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 dynamic 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.

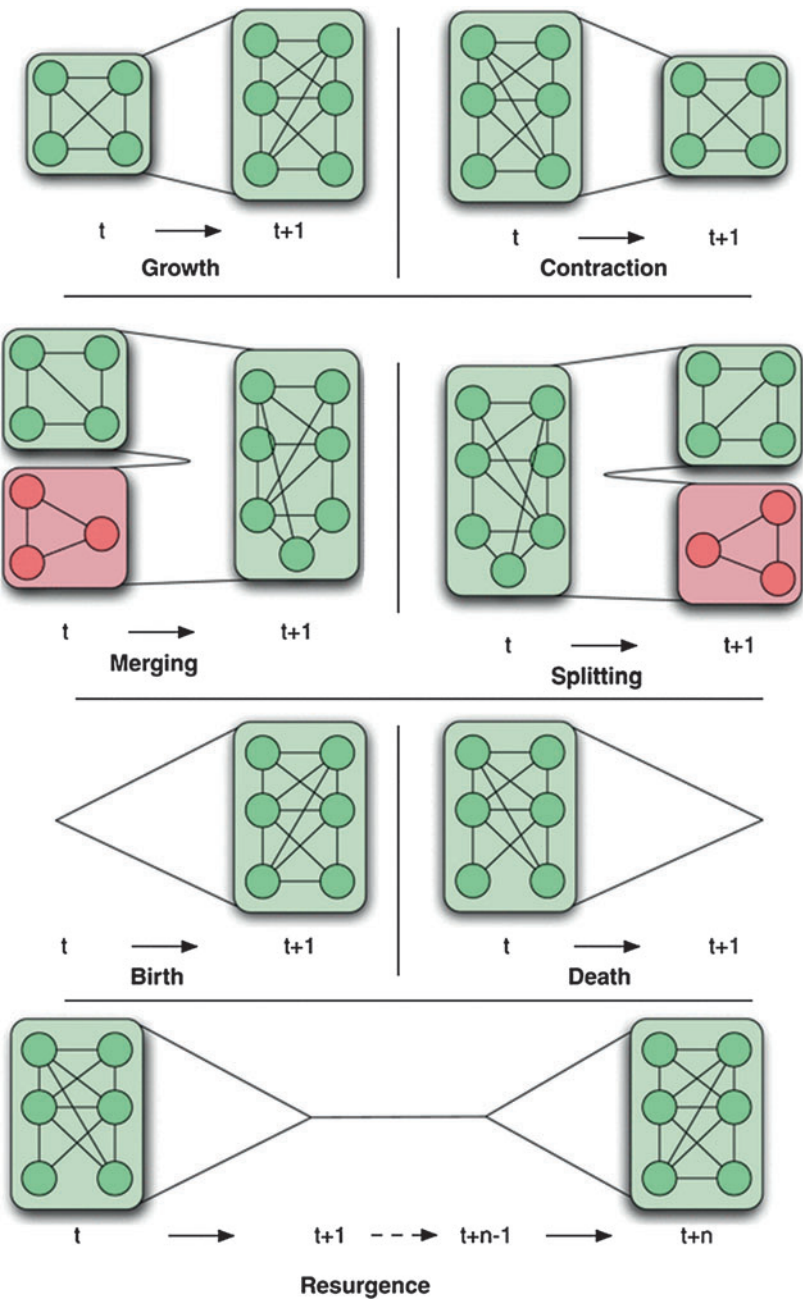
Merging: two communities or more can merge into a single one.

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.

Dynamic Community Detection,

Fig. 1 Possible operations on communities



Death: a community can vanish at any time. All nodes that were belonging to this community lose 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 us imagine that they stop to

play together during the summer vacations, while a snapshot of the network is taken every month. The community 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 2 months.

All of these operations, however, cannot be detected by every existing algorithm. Some algorithms focus on stable communities and do not search for growth and contraction. Most algorithms are able to detect birth and death of communities. But merging and splitting are more complex operations and therefore less frequently considered. 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 us 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 raises 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 that can be taken to obtain these communities as some of the most prominent algorithms published so far.

The three identified approaches differ by their definition of what good dynamic communities are; they are *instant optimal*, *temporal trade-off*, and *cross-time community* detection approaches.

#### Instant Optimal

In this approach, communities identified at time  $t$  must be the most relevant communities considering only the state of the network at time  $t$ .

The typical method adopted to solve this problem is the *two-stage approach*:

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

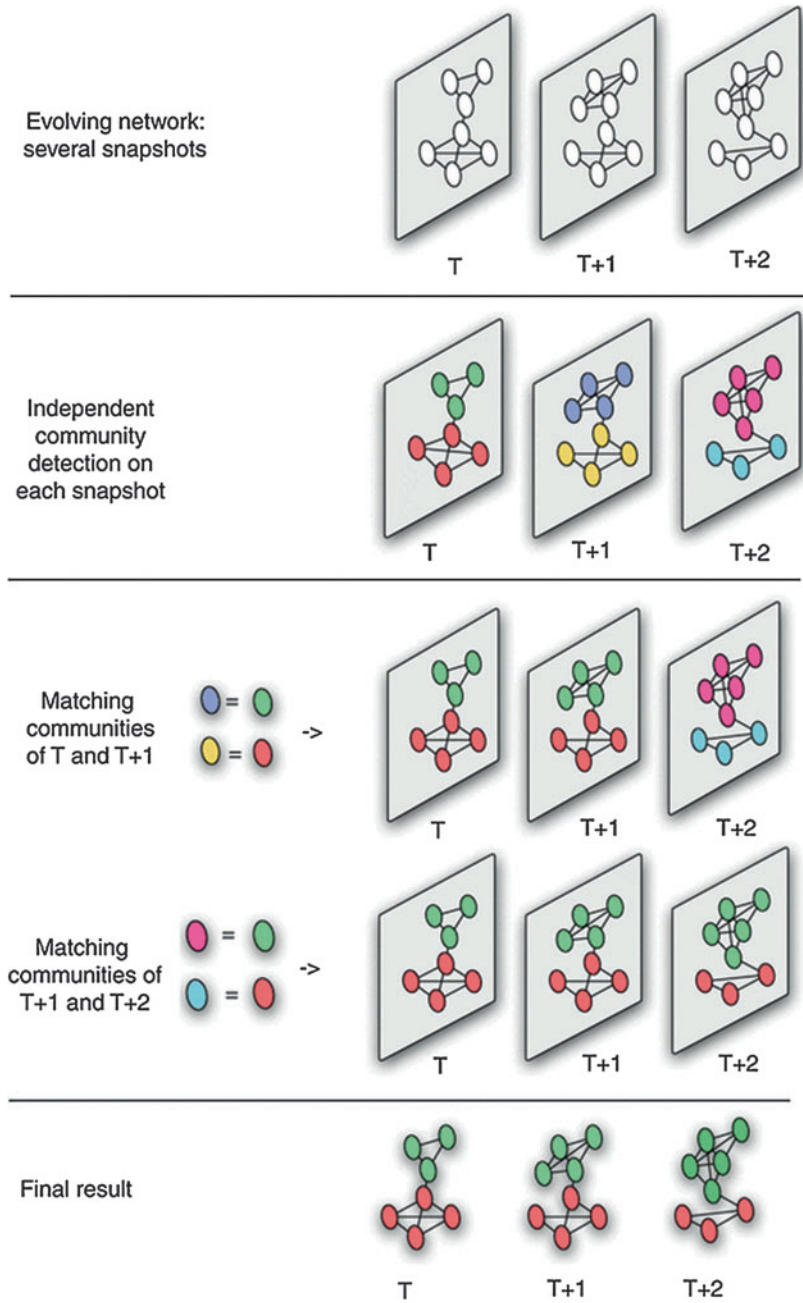
This process is illustrated in Fig. 2.

The earliest 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), Greene et al. (2010), Dhouioui and Akaichi (2014), and Ilhan and Oğuducu (2015).

**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.

### Dynamic Community Detection,

**Fig. 2** Dynamic community detection by *instant optimal* communities



**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 partition.

As a result, when we observe modifications between two snapshots, we cannot know if they correspond to real structural modifications of the 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 other approaches try to fix it.

#### Temporal Trade-Off

In this approach, communities identified at time  $t$  correspond to a trade-off between the best communities relatively to the network at  $t$  and the history of found communities. There are two main categories of such algorithms: those who updates the communities existing at  $t-1$  according to changes in the network between  $t$  and  $t-1$  (implicit trade-off) and those who explicitly define a quality function as a trade-off between a quality function for communities at  $t$  (e.g., modularity) and a similarity between partitions at  $t$  and at  $t-1$  (e.g., normalized mutual information NMI).

The process usually follows three steps:

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

This process is illustrated in Fig. 3.

This approach is notably used by Hopcroft et al. (2004), Palla et al. (2007), Wang et al. (2008), Cazabet et al. (2010), Rosvall and Bergstrom (2010), Chen et al. (2010), Greene et al. (2010), Cazabet and Amblard (2011), Xie and Szymanski (2013), Gorke et al. (2013), Guo et al. (2014), Zakrzewska and Bader (2015), and Rossetti et al. (2016).

**Advantages** This method allows coping 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.

#### Cross-Time Communities

With this approach, all steps of evolution are studied simultaneously. A single community detection is run, considering in a single pass all the periods of the network and yielding a single decomposition.

This process is illustrated in Fig. 4 for the snapshot case.

There are two main 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, a community detection algorithm can be run.

Another solution is to design a quality function that can directly be optimized on several snapshots; this has been done with modified versions of modularity and stochastic block models (SBM).

This approach is notably used in Tantipathananandh et al. (2007), Jdida et al. (2007), Mucha et al. (2010), Aynaud and Guillaume (2010), Yang et al. (2011), Guo et al. (2014), Viard et al. (2016), and Matias et al. (2015).

**Advantages** The main advantage of this solution is to eradicate the problem of instability. The coherence of detected communities is ensured on the long run.

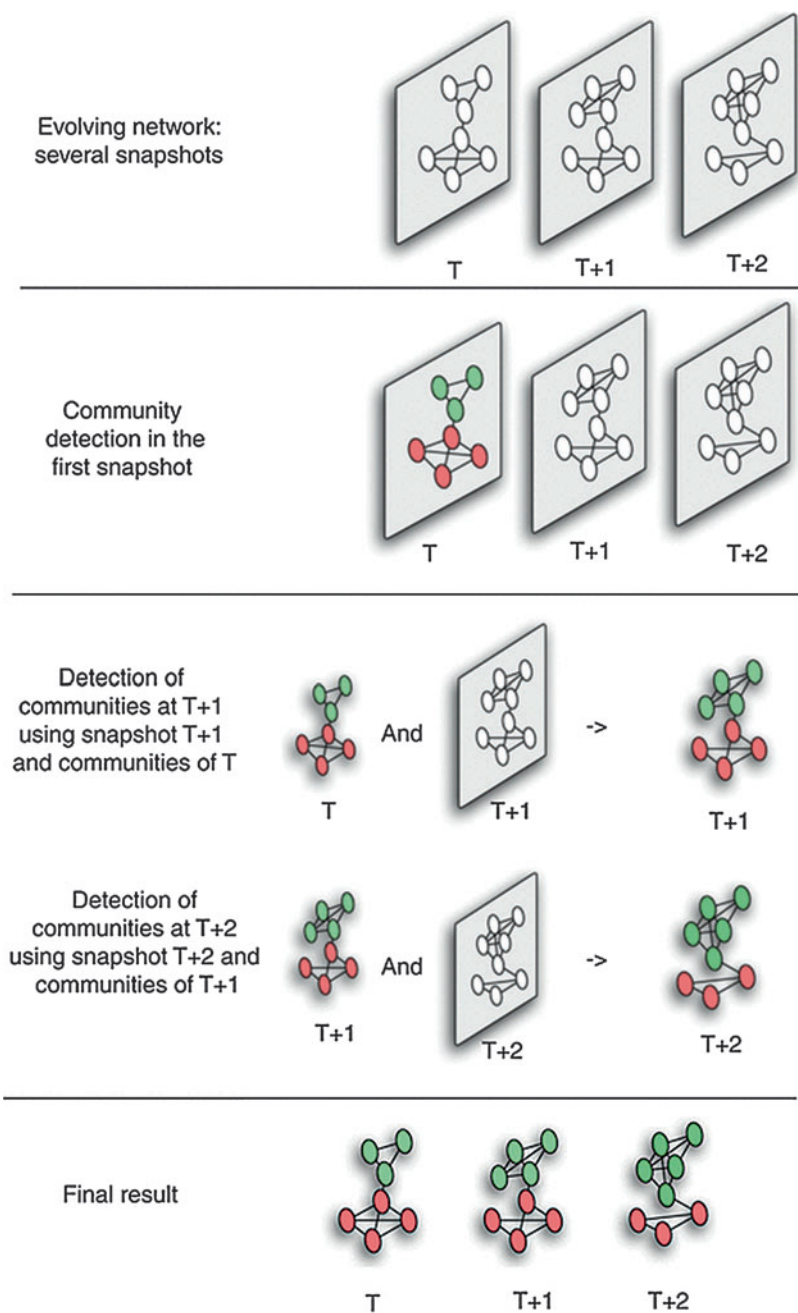
**Drawbacks** With this method, it is not possible to ensure coherent dynamic community detection on a network evolving in real time. As the solution is not iterative, if a new snapshot is added, it is not possible to keep the previous result and just update the evolution of communities with the latest data.

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 is able to find them.



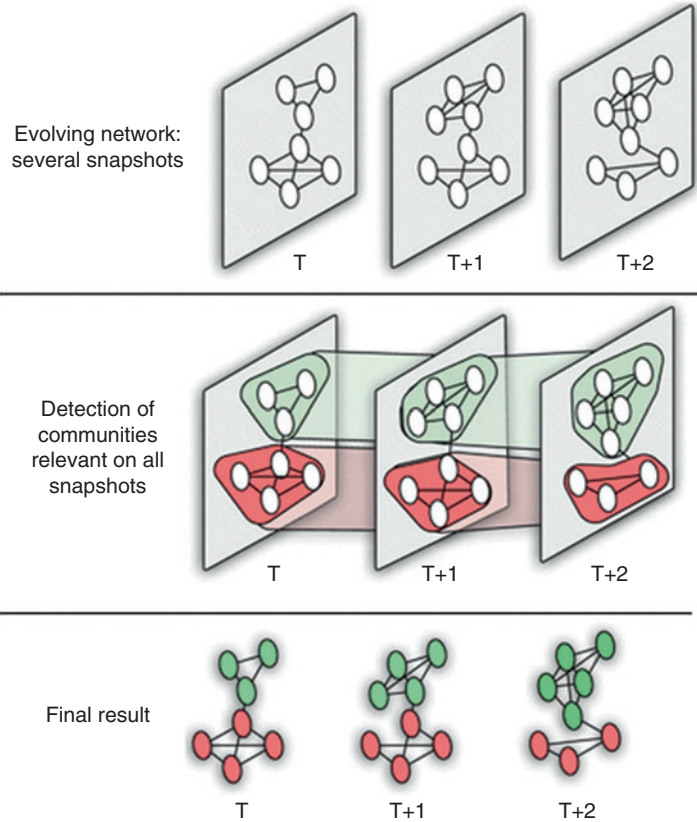
**Dynamic Community Detection,**

**Fig. 3** Dynamic community detection by informed iterative static community detections



### Dynamic Community Detection,

**Fig. 4** Dynamic community detection by global detection on all snapshots



### Conclusion

In conclusion, we have seen that dynamic 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

Dynamic 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 dynamic 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 coauthorship 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 dataset 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 2 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

Dynamic 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 dynamic community analysis 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 dynamic communities.

Another domain, which will probably be explored more deeply, is the processing and visualization of the results. Obtaining dynamic communities is the 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 (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), vol 2. IEEE, Lyon, pp 402–408
- 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, 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. ACM, Philadelphia, pp 554–560
- Chen W, Liu Z, Sun X, Wang Y (2010) A game-theoretic framework to identify overlapping communities in social networks. *Data Min Knowl Disc* 21(2):224–240
- Dhouioui Z, Akaichi J (2014) Tracking dynamic community evolution in social networks. In: IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 10 Asonam (2014)), Beijing, pp 764–770
- 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
- Gorke RG, Maillard P, Schumm A, Staudt C, Wagner D (2013) Dynamic graph clustering combining modularity and smoothness. *J Exp Algorithmics* 18(1):1.1–1.29

- 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). IEEE, Odense, pp 176–183
- Guo C, Wang J, Zhang Z (2014) Evolutionary community structure discovery in dynamic weighted networks. *Phys A Stat Mech Appl* 413:565–576
- 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. *Proc Natl Acad Sci U S A* 101(1):5249–5253
- Ilhan N, Oğuducu SG (2015) Predicting community evolution based on time series modeling. In *DyNo workshop ASONAM*, Paris
- Jdida 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*, vol 2. Lyon, pp 744–749
- Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. *Phys Rev E* 78(4):046110
- Matias C, Rebafka T, and Villers F (2015) Estimation and clustering in a semiparametric Poisson process stochastic block model for longitudinal networks. *arXiv preprint arXiv:1512.07075*
- 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
- Rossetti G, Pappalardo L, Pedreschi D, Giannotti F (2016) Tiles: an online algorithm for community discovery in dynamic social networks. *Mach Learn*:1–29. doi:10.1007/s10994-016-5582-8
- Rosvall M, Bergstrom CT (2010) Mapping change in large networks. *PLoS One* 5(1):e8694
- 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*. ACM, San Jose, pp 717–726
- Viard J, Latapy M, Magnien C (2016) Computing maximal cliques in link streams. *Theor Comput Sci* 609:245–252
- Wang Y, Wu B, Du N (2008) Community evolution of social network: feature, algorithm and model. *Sci Technol arXiv:0804.4356*
- Xie J, Szymanski BK (2013) Labelrank: a stabilized label propagation algorithm for community detection in networks. In: *Network Science Workshop (NSW)*, 2013 I. E. 2nd. IEEE, pp 138–143
- 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
- Zakrzewska Anita and Bader David A (2015) A dynamic algorithm for local community detection in graphs. In *Proceedings of the 2015 IEEE/ACM international conference on advances in social networks analysis and mining 2015*. ACM, Paris, pp 559–564