Towards Dynamic Graph Analysis: a Position Note

Erwan Le Merrer
Technicolor, Rennes

Gilles Trédan
LAAS-CNRS, Toulouse

In this note, we elaborate on directions for efficient graph analysis, taking into account the dynamic nature of complex networks.

1 Introduction and problematic

Tailored network analysis is at the core of some of the most successful Internet stories, as the Google search engine for instance [2]. Understanding complex networks is a challenge that receives a growing attention. Interestingly, this challenge is tackled not only by computer scientists, but also by biologists and physicists. Together, these efforts led to the definition of several important methodological tools to extract information from a complex network, such as centralities, sampling, or scale-free property analysis.

However, most of these tools have been defined in a static way, by only providing insights on crawled networks, at some fixed point in time [3]. This is notably counter intuitive, as the complex networks observed are by nature evolving constantly [4]; predictions are doomed to be inaccurate.

At the same time, researchers started to develop models that capture graph dynamics over time. For instance the concept of *time varying graphs* (TVGs) [1] has been recently recognized as a suitable model to describe the evolution of a network in time, pushed by the development of delay-tolerant and opportunistic networks. The TVG model may then constitutes the next step of understanding complex networks, observing them on a dynamic time dimension.

Adapting the aforementioned static tools to TVGs may seem easy. A basic solution is to simply take snapshots of the network at fixed intervals in time, and to run the analytical computation based on those snapshots [5]. This is of course cumbersome if the time interval is small, and even simply not computationally feasible if the observed network is large; this prohibits online use by resource-limited devices or applications for instance.

Since the scale of each snapshot already forbids the use of expensive algorithms, it is clear to us that analyzing TVGs calls for developing methods to reduce the size of the input dataset[1]. The question we ask in this note is then the following one: **"How to reduce the computational/memory costs when analyzing TVGs?"**.

2 Possible directions

We foresee 4 directions to reduce the global cost in computing characteristics of TVGs: *i)* observe a subset of *relevant* nodes over time, *ii)* concentrate on a subset of *relevant* snapshots, *iii)* observe the evolution over time of a set of graph measures and *iv)* develop/adapt graph measures to TVGs. We will hereafter briefly detail each approach.

Sample nodes Sampling has been extensively studied and provides accurate results in static contexts. This is probably a very efficient approach to gather the evolution of aggregated values, such as the evolution of the average age of social network users. However unbiased sampling is difficult, and adding the time dimension complicates the problem.

The question of newcomers (nodes joining the system after the beginning of the computation) is pending: should we only sample nodes from the first snapshot ? Moreover, such an approach cannot answer questions regarding the evolution of the graph structure itself.

[1] `Google+` graph has now more than 40e6 users. At the time of this writing, its most famous user receives more than 3000 followers each day.

Sample snapshots Here, the objective is to reduce the analysis complexity by discarding snapshots that bring little information to concentrate on those conveying more meaning. However, selecting which snapshot is relevant to keep is a non trivial task.

One difficulty is tied to the arbitrarily defined interval at which a snapshot should be kept for analysis. One could create a snapshot for each topological change, *i.e.* appearance/disappearance of a node or edge (called *graph centric evolution* in [1]). It is also possible to spare resources and use a sampling rate that is lower than the graph update rate, but the danger is then to loose key information, such as causality relations in the link/node creation mechanism. It might as well be of a real interest to sample more snapshots when topological changes occur in the core of the network (or on nodes or edges that have been previously marked as important), than when events occur at the fringe.

Evolution of graph measures The approach here is to run standard static graph analysis tools on each snapshot independently, and to interpret the obtained time series. Reusing the static graph analysis toolbox is probably a very good starting strategy. Similarly to node sampling, this strategy has the drawback of only studying the evolution of static graph properties.

Keeping only an history of past analysis concerning a network may drastically reduce memory needed to track evolution in time, when compared to the snapshot-based approach. Starting from an initial network analysis, algorithms are providing outputs that are compared to previously observed results (one can for instance keep track of network diameter [4]). Only one network snapshot is then required in memory, for each evolution analysis, as compared to an array of snapshots as in the previous solution. The negative side being of course the loss of information when only conserving past measures instead of full network snapshots.

Adapted analytical tools The approach here is to develop new algorithms suited to TVGs and their natural information redundancy. For instance, in many social networks, such as the co-authorship graph in DBLP, links are never removed. Each snapshot is thus a subset of the following one. Recycling the results of previous snapshots to empower the current snapshot analysis could then lead to significant optimizations.

There is probably no generic solution for transforming static algorithms into their dynamicity-aware counterparts. For most of known tools for graph analysis, even the slightest update of the initial graph might have unpredictable consequences on the result of the algorithm.

Most of all, we believe that some very important TVG metrics are only definable in a dynamic context. Even very simple but interesting questions such as "how long does a link takes to be reciprocal, for a triangle to be closed ?" can not be addressed by simply adapting our static toolbox to a dynamic setup.

When collecting data from evolving structures, researchers are often bound to the limitations of the API or available datasets. Thus, it is also important to think about the robustness of these complexity reduction techniques against incomplete datasets. For instance, the sampling interval might not be a parameter under control because of crawling time and delays. It is often hard to ensure that the full graph is collected. Moreover, crawling times might lead us to question the association of one snapshot to one point in time.

We believe that each of these directions could be valuably explored. It is still unclear whether some methods are above the others with respect to costs, but each of them probably leads to the observation of different interesting phenomena; we have to understand their respective relevance to specific contexts.

References

[1] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. In *ADHOC-NOW*, pages 346–359, 2011.

[2] M. Franceschet. Pagerank: standing on the shoulders of giants. *Commun. ACM*, 54:92–101, 2011.

[3] A.-M. Kermarrec, E. Le Merrer, B. Sericola, and G. Trédan. Second order centrality: Distributed assessment of nodes criticity in complex networks. *Comput. Commun.*, 34:619–628, 2011.

[4] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of ACM SIGKDD*, KDD '05, pages 177–187. ACM, 2005.

[5] G. Palla, A.-L. Barabi, and T. Vicsek. Community dynamics in social networks. *Fluctuation and Noise Letters*, 7:273–287, 2007.