Forte Δ-connexité dans les flots de liens

Charles Huyghues-Despointes, Binh-Minh Bui-Xuan, Clémence Magnien

Forte Δ-connexité dans les flots de liens

Charles Huyghues-Despointes[1,2], Binh-Minh Bui-Xuan[3] et Clémence Magnien[3]

[1]ERIC, *Universités Lyon 1 et Lyon 2*
[2]*AMI Software*
[3]*Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris.*

Les flots de liens modélisent les interactions temporelles et sont constitués de paires (t,uv) représentant un lien entre les sommets u et v au temps t.

La question de la connexité est plus difficile dans le cas des flots de liens que dans le cas des graphes, car la relation d'accessibilité n'est pas transitive. Nous étudions la notion de composante fortement Δ-connexe dans les flots de liens (ou les graphes évolutifs). Le calcul exact des composantes fortement Δ-connexes étant NP-difficile, nous proposons un algorithme qui donne une borne supérieure et une borne inférieure de la taille de la plus grande composante.

Mots-clefs : flots de liens, connexité, algorithmique

1 Introduction

The question of connectivity in graphs is well-defined and understood: it is possible to partition a graph's nodes into connected components, such that there is a path from any node to any other node of the same component. Moreover, the computation of this partition can be done in linear time in the size of the graph, with simple algorithms such as a breadth-first search.

In the case where we do not deal with a classical, static graph, but where the links evolve with time, we are faced with two problems. First, though it is straightforward to extend the classical notion of path to that of a temporal path, the corresponding reachability relation is not transitive. This means that the natural extension of the notion of connected components will not lead to a node partition, and instead the components will overlap. Second, computing these connected components is an NP-hard problem [BF03].

The question of connectivity in the dynamic case has been addressed [BF03, NTM$^+$12, TMML10]. In these cases, a strongly connected component is defined as a set of nodes such that there exist a temporal path from any node to any other node in the set.

However, temporal paths may span a small or large fraction of the total duration of the considered link streams. Hence, the fact that a link stream is strongly connected can describe vastly different situations. For instance, there could exist a small time interval such that there are temporal paths from any node to any other node in this interval; in this case no description of the dataset outside of this time interval is given; or the temporal paths may have a time length comparable to that of the whole dataset; or other possibilities.

In order to provide a better description of a link stream with respect to connectivity, we are interested in finding stretches of time during which paths exist recurrently. In an email network, this would for instance amount to: every day, an information can be iteratively forwarded from any user X to any user Y of usergroup `group`. This corresponds to the notion of strongly Δ-connected components, introduced in [GCCLL15] as Δ-*components*.

In this paper, when addressing evolving network data, we will capture timestamped links between nodes under the formalism of a link stream. A *link stream* is a triple $L = (T,V,E)$ where $T = [0,t_{max}]$ represents a time interval, V is a finite set of *nodes*, and $E \subseteq \{(t,uv) : t \in T \wedge u \in V \wedge v \in V \wedge u \neq v\}$ is a set of

timestamped unordered pairs of nodes called *timelinks*. A *timelink* is a pair (t, uv) where t is a time instant and uv is a notation which represents the unordered link $uv = vu$ between two distinct nodes u and v [†].

We show that computing the maximal strongly Δ-connected components is a NP-hard problem, and we propose algorithms for computing upper bounds and lower bounds on their size.

2 Bounding strongly Δ-connected components in link streams

Let $L = (T, V, E)$ be a link stream where $T = [0, t_{max}]$. For any subset $S \subseteq V$ of nodes we call $L[S] = (T, S, E[S])$ the substream of L induced by S, where $E[S] = E \cap \{(t, uv) : t \in T \wedge u \in S \wedge v \in S\}$.

Definition 1 (Strongly Δ-connected component) A *temporal path* from u to v, starting at t_0 and ending at t_k, for $k \geq 0$, is a sequence (t_0, x_0x_1), (t_1, x_1x_2), ..., (t_k, x_kx_{k+1}) of timelinks of E where $t_0 < t_1 < \ldots < t_k$, $x_0 = u$ and $x_{k+1} = v$. Link stream L is *strongly Δ-connected* if for any pair of distinct nodes u and v, for any time instant $t \in [0, t_{max} - \Delta]$, there exists a temporal path from u to v starting after t and ending before $t + \Delta$. A subset $S \subseteq V$ of nodes is strongly Δ-connected if the induced substream $L[S]$ is strongly Δ-connected, and it is called a component when maximal by inclusion.

For instance, consider the link stream $L_s = (T_s, V_s, E_s)$ where $T_s = [0, 3]$, $V_s = \{a, b, c, d, e\}$, and $E_s = \{(0, ab), (1, bc), (1, be), (2, cd), (2, ed), (3, ad)\}$. For $\Delta = 3$, there are only three missing temporal paths: from c to e; from e to c; and from d to b. Hence, there are exactly four strongly Δ-connected components of L_s, which are $\{a, b, c\}$, $\{a, b, e\}$, $\{a, c, d\}$, and $\{a, d, e\}$. They all overlap each other in a non-trivial way.

We claim that finding the largest strongly Δ-connected component of link stream L is NP-hard. The best way to see this is to consider the Δ-reachability digraph of L.

Definition 2 (Δ-reachability digraph) The *Δ-reachability digraph* of link stream $L = (T, V, E)$ is the digraph $D_L = (V, A)$ where an arc (u, v) belongs to the arc set A if and only if for any time instant $t \in [0, t_{max} - \Delta]$ there exists a temporal path from u to v starting after t and ending before $t + \Delta$.

By extension, if $S \subseteq V$, we denote by D_S the Δ-reachability subgraph of $L[S]$.

Strongly Δ-connected components of link stream L are exactly maximal cliques of digraph D_L. We can then prove the NP-hardness by making a reduction from the problem of cliques in classical graphs to the problem of cliques in Δ-reachability digraphs, in a similar way as what has been done in [BF03].

2.1 Lower bounding strongly Δ-connected components

We are interested in the size of the largest strongly Δ-connected component. We present here a partition refinement approach for finding large strongly Δ-connected sets, which provides a lower bound for the size of the largest component.

Algorithm 1 Input: link stream $L = (T, V, E)$. First partition V into strongly connected components of its Δ-reachability digraph D_L. For each partition part S that is not strongly Δ-connected in L, successively perform both following splitting operations. First split S into two parts $S \setminus \{u\}$ and $\{u\}$ following some RULE. Second, split $S \setminus \{u\}$ into strongly connected components of the Δ-reachability digraph $D_{S \setminus \{u\}}$. Keep on refining as long as there are partition parts that are not strongly Δ-connected. After refinement stability, scan through the partition parts and output their maximum size as a lower bound of the maximum strongly Δ-connected component of link stream L.

Claim 1 *Algorithm 1 first computes a partition of V into strongly Δ-connected sets of L before outputting the maximum size of these sets, hence, correctly outputs a lower bound of the maximum strongly Δ-connected component of link stream L.*

Proof: The main idea here is that cliques of digraphs are subsets of their strongly connected component. □

This process is an inversely incremental approach whose efficiency depends heavily on the step when we split S into $S \setminus \{u\}$ and $\{u\}$ according to RULE. The exact choice of RULE will be discussed below. Concerning complexity issues, it would be time consuming to compute the many Δ-reachability digraphs

[†] As in most timestamped log data, we suppose that the input of timelink set E is already sorted by time instants.

involved in the process. However, we are not really interested in these digraphs, except for their strongly connected components. We define the (undirected) *underlying graph* of link stream $L = (T, V, E)$ as $\overline{G} = (V, \overline{E})$ where $\overline{E} = \{uv : \exists t \in T, (t, uv) \in E\}$.

Claim 2 *For any link stream, the strongly connected components of its Δ-reachability digraphs are exactly the connected components of its underlying graph.*

Proving this claim is a straightforward exercise. Note that computing connected components of an undirected graph can be done in linear time, and the graph itself can be obtained implicitly from the corresponding link stream.

We now address the test condition whether a partition part S is strongly Δ-connected in link stream L. By lack of space, we skip the details and claim that it is possible to modify the algorithm computing temporal distance vector in [KKW08] and devise an algorithm deciding strong Δ-connectedness.

Eventually, we are left with defining the RULE for choosing the node $u \in S$ in the splitting of S into $S \setminus \{u\}$ and $\{u\}$. We pick u as the node of the Δ-reachability digraph of $L[S]$ having the smallest sum of in and out degrees. We experimented with other criteria and this one provided the best results. We leave a precise evaluation of the impact of this RULE for future work. It is possible to find this node by modifying the above-mentioned variant of Algorithm [KKW08]. The overall complexity is therefore at most that of the routine for testing whether S is strongly Δ-connected in L.

2.2 Refining the timeline and upper bounding strongly Δ-connected components

Link streams may or may not be strongly Δ-connected, and this may evolve with time (and the sizes of components may also evolve). For instance, professional email exchanges are not to be expected during weekends and holidays. It is therefore important to be able to compute time intervals where some special phenomenon occurs, e.g. when the network records heavy activities or, on the contrary, a very quiet period. Formally, for any time interval $I \subseteq T$ we call $L[I] = (I, V, E[I])$ the substream of L induced by I, where $E[I] = E \cap \{(t, uv) : t \in I\}$. Here we cover the timeline with overlapping time intervals using the formalism of Δ-existence.

Definition 3 (Δ-existence) A node $u \in V$ of link stream $L = (T, V, E)$ is said to Δ-*exist* at time instant $t \in T$ if there exists a node $v \in V$ and a time instant $t' \in [t, t + \Delta]$ such that $(t', uv) \in E$. We say that an interval $I = [a, b]$ of time $I \subseteq T$ is a *consistent interval* if every node of the link stream satisfies one of the following: either the node Δ-exists at all time instant $t \in [a, b]$; or, the node does not Δ-exist at any time instant $t \in [a, b]$.

Definition 4 (Candidate interval) For any consistent interval $I = [a, b]$ as described above, the *candidate interval* associated to I is the time interval $[a, b + \Delta]$. Notice that, as consistent intervals form a partition of the timeline, candidate intervals may overlap.

The computation of Δ-existence of a node can be done by scanning the link stream. From that computation, outputting *consistent intervals* (and hence candidate intervals) is a straightforward exercise.

Finally, for every candidate interval I, the number of nodes which Δ-exist throughout I is an upper bound of the size of the maximum strongly Δ-connected component of $L[I]$.

3 Application to the Rollernet dataset

We considered the Rollernet dataset [TLD+09], collected during a rollerblade tour in Paris. 62 participants were equipped with wireless sensors recording when they are at a communication distance from one another. The total dataset duration is approximately 2 hours and 45 minutes (with a break of approximately 30 minutes). The candidate intervals, as well as the corresponding upper and lower bounds for the size of the largest strongly Δ-connected component are presented in Figure 1. The computation for each value of Δ took just under 3s on a laptop.

We observe a very long candidate interval (minutes 20 to 65) where the bounds show that the maximum strongly Δ-connected component includes over 90% of the nodes, for both values of Δ. The following candidate interval also lasts for a long period of time (> 25 minutes). For $\Delta = 6$ minutes the bounds again indicate that the largest component spans over 90%of the nodes. For $\Delta = 4$ minutes, the lower bound shows

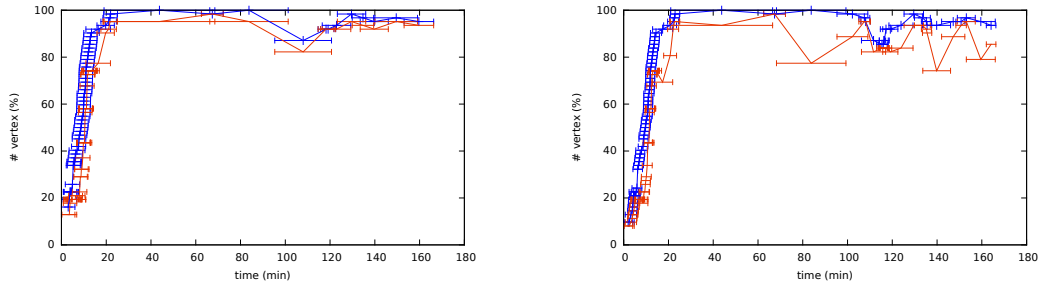

Fig. 1: Candidate time interval covering of Rollernet, with upper and lower bounds of the size of the maximum strongly Δ-connected components for $\Delta = 6$ minutes (left) and $\Delta = 4$ minutes (right). Red (resp. blue) segments correspond to lower (resp. upper) bounds during the corresponding candidate interval. The middle points of the segments are joined for a better interpretation.

that the largest component includes over 75% of the total number of nodes for the same interval. We believe that this bound can be improved to be closer to the upper bound, and leave this for future work.

Our results strongly suggest that, after a short initial phase (20 first minutes), the Rollernet network records dense and regular (every 5 minutes) exchanges for a long lasting period, until a drop of activity (between 90th and 120th minute, probably corresponding to the break) after which point a regular communication density can resume again.

4 Conclusion

We presented an algorithm for computing upper and lower bounds for the size of the largest strongly Δ-connected components. We also introduced a method for splitting a linkstream's timeline into relevant subintervals. Illustrated on a real-world dataset, our method highlights some interesting features and allows a better understanding of the dataset.

Future work includes the computation of the exact time complexity of the algorithm, as well as formal results on the bounds' tightness. We think it is possible to improve the tightness of the lower bound. Finally we aim at making more detailed analysis of this and other datasets.

References

[BF03] S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *Proceedings of ADHOC-NOW*, volume 2865, pages 259–270, 2003.

[GCCLL15] Carlos Gómez-Calzado, Arnaud Casteigts, Alberto Lafuente, and Mikel Larrea. A connectivity model for agreement in dynamic systems. In *EuroPar*, 2015.

[KKW08] Gueorgi Kossinets, Jon Kleinberg, and Duncan Watts. The Structure of Information Pathways in a Social Communication Network. In *KDD'08*, 2008.

[NTM+12] Vincenzo Nicosia, John Tang, Mirco Musolesi, Giovanni Russo, Cecilia Mascolo, and Vito Latora. Components in time-varying graphs. *Chaos*, 22, 2012.

[TLD+09] Pierre Ugo Tournoux, Jérémie Leguay, Marcelo Dias de Amorim, Farid Benbadis, Vania Conan, and John Whitbeck. The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing. In *Proceedings of IEEE Infocom*, pages 1116–1124. IEEE, 2009.

[TMML10] John Tang, Mirco Musolesi, Cecilia Mascolo, and Vito Latora. Characterising temporal distance and reachability in mobile and online social networks. *ACM SIGCOMM Computer Communication Review*, 40(1):118–124, 2010.