Time-Varying Graphs Analysis via Δ-duplication

François Queyroi

May 26, 2014

In this paper, we introduce a transformation of time-varying graphs, called Δ-duplication, for the analysis of heterogeneous dynamic network. Instead of building a sequence of snapshots from non-overlapping time windows, we propose a relative approach : consider a vertex as different vertices over multiple *sessions*. A *session* being a time period where the vertex interaction are not separated by more than Δ timestamps. We describe here the general theory of Δ-duplication and provided some directions for applications to TVG analysis. In particular, we introduce a generalization of k-core to temporal graph using this model.

1 Introduction

Complex networks often corresponds to dynamic complex systems. In this context, time-varying graphs [2] (TVG) are used to capture this dynamic : the interactions between actor in the networks appear and disappear according to an unknown process. In practical applications, this process is heterogeneous, in the sense it may be time dependent. Moreover, we can assume the events in a dynamic complex networks are not independent.

One way to deal with this temporal and topological complexity is to reduce the time heterogeneity by cutting the time into different windows. This approach transforms a TVG into a series of static graphs called *snapshots* [5, 7]. An implicit hypothesis in this case is that the way interactions are ordered in those snapshots is not relevant for the analysis, so we can study them as static graph (eventually weighted). So far this reduction is done globally (*e.g.* consider all interactions within an hour, a day, a week) using time windows which seems relevant for the applications' domains.

We want here to reduce the time heterogeneity without considering an absolute time-line. Instead, we propose a vertex-oriented approach *i.e.* where each vertex time-line defined by its interactions is cut according to a parameter Δ. Those cuts define *sessions* which correspond to activity periods of the vertices : time interval in which an individual interacts at least every Δ. This gives birth to a class of TVG we call Δ-successive time-varying graphs (Section 2). We show that a TVG can be turned into a unique Δ-successive Temporal Graphs using a polynomial algorithm (see Section 3). This transformation allows us to define an important network metric, the k-cores, in the context dynamic networks (see Section 4). The (k, Δ)-cores can be used to detect sub temporal graph with low-bounded temporal and topological connectivity.

2 Δ-successive Temporal Graphs

In this section, we introduce a class of TVG called Δ-successive TVG. We call a TVG a tuple $G = (V, E, T, t)$ where V is the set of vertices, $E = V \times V$ the set of edges and $t : E \to [0, T]$ is the time at which an edge is observed. Notice we consider only instantaneous interactions[1], however most of the concepts developed later can be defined for more accurate models, this is further discussed in Section 5.

In this context, we call $t(V')$ the set of different timestamps for which it exists an edge with one endpoint in V'. Furthermore, we call $[\max t(V'), \min t(V')]$ the *session* of V' and define as $\tau(V') = \max t(V') - \min t(V')$ its *duration* of V' *i.e.* the length of the period for which V' has interactions in G. Notice that for any subset V', we have $\tau(V') \leq T$.

Definition 1. *Δ-session. The list of integer $X = (x_1, x_2, \ldots, x_k)$ is called a Δ-session iff for all $i \in [1, k-1]$, $x_{(i+1)} - x_{(i)} \leq \Delta$ where $x_{(i)}$ is the i-th biggest element in X.*

A temporal graph is Δ-successive if all the interactions of a given vertex are separated by at most Δ timestamps.

Definition 2. *Δ-successive Graphs. A temporal graph $G = (V, E, t)$ is said to be* Δ-successive *for $\Delta \in [0, T]$ iff for all $v \in V$, $t(v)$ is a Δ-session.*

Testing whether a temporal graph is Δ-successive is straightforward. We can derive from this definition some useful properties.

Property 1. *Minimum frequency. Let $f_G(v) = \dfrac{d_G(v)}{\tau(v)}$ the frequency of interactions of a vertex over its session. If G is Δ-successive, then $f_G(v) \geq \Delta^{-1}$ for all $v \in V$.*

Property 2. *Inclusion. For $\Delta_1 \leq \Delta_2$, every Δ_1-successive graph is also Δ_2-successive.*

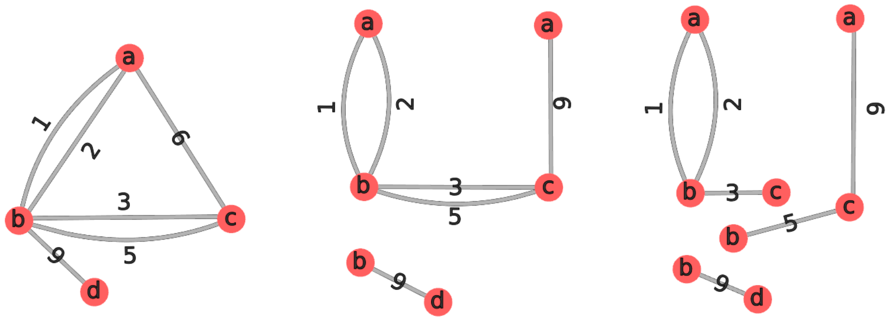
Observe that a Δ-successive G can contain sub-graphs (induced by a set of vertices or formed by a set of edges) that are not Δ-successive themselves.

3 Δ-Duplication of Temporal Graphs

For every time-varying graph it exists a Δ such that the graph is Δ-successive. We show now that any temporal graph can be turned into a Δ-successive with lower Δ using vertex duplication.

[1]We consider the time as integers value since timestamped observations can only be done in discrete time. This also justify the existence of instantaneous interactions in this context.

Definition 3. *Vertex-Duplication. The graph $S = (V', E, \theta)$ is a* Vertex-duplication *of $G = (V, E)$ iff $\theta : V' \to V$ is a subjective function such that the graph obtained after the contraction of the vertices $\{v \in V', \theta^{-1}(v) = u\}_{u \in V}$ is G.*



(a) $\Delta = 4$ (b) $\Delta = 2$ (c) $\Delta = 1$

Figure 1: Different (minimum) Δ-duplications of the graph (a). Edges labels are the timestamps $\{t(e)\}_{e \in E}$.

Notice a vertex-duplication is a transformation of G without information loss. We will use here the θ function to encodes vertex *sessions*. More precisely, the couples (vertex,session) will be considered the new vertices of the temporal graph with temporal edges routed accordingly. A vertex will therefore be seen as separate entities if the time spent between two consecutive interactions is higher than Δ. An example of Δ-duplications can be found in Fig. 1.

Definition 4. *Δ-duplication. The temporal graph $S = (V', E, T, t, \theta)$ is a* Δ-duplication *of $G = (V, E, T, t)$ iff S is a vertex-duplication of G and is* Δ-*successive.*

Definition 5. *Minimum Δ-duplication. The temporal graph $S = (V', E, T, t, \theta)$ is a* minimum Δ-duplication *of $G = (V, E, T, t)$ iff $|V'|$ is minimum among all Δ-duplications of G.*

Notice that the minimum T-duplication of G is itself. Equivalently, the minimum 0-duplication of G contains $\sum_{u \in V} |t(u)|$ vertices.

Theorem 1. *Existence and Uniqueness. For every temporal graph G, it exists a unique minimum Δ-duplication of G.*

Proof. A Δ-duplication of G exists since it corresponds to the partitioning of the sets $\{t(v)\}_{v \in V}$ into Δ-sessions. Such partition always exists. Moreover, the minimum Δ-duplications is obtained by partitioning all set $t(v)$ into the minimum number of groups. There is only one way to achieve this : by splitting only the intervals of length greater than Δ between consecutive interactions. This proves the uniqueness. □

Theorem 2. *Complexity. The minimum Δ-duplication of $G = (V, E, T, t)$ can be computed in $\mathcal{O}(|E| \log |E|)$ and can be stored in $\Theta(|E|)$ assuming $|V| \leq |E|$.*

Proof. Time: As said earlier, the minimum Δ-duplication is achieved by splitting $t(v)$ for every consecutive step of length greater than Δ. It requires sorting the edges according to the timestamps. The rerouting of the edges can be done in $\mathcal{O}(|E|)$.

Memory: the maximum number of vertices in a minimum Δ-duplication of G is achieved for $\Delta = 0$. Assuming $|V| \leq |E|$, we have

$$|V'| \leq \sum_{u \in V} |t(u)| \leq \sum_{u \in V} d_G(u) = 2|E|$$

Since the edge set stays the same, the minimum Δ-duplication of G can be stored in $\Theta(|E|)$.

□

Notice that if the edges incident to a vertex are ordered according to the timestamps, the computation of the Δ-duplication can be done in $\mathcal{O}(|E|)$.

4 Generalization of k-cores to temporal graphs

In the section, we describe a possible generalization of the well-know concept of k-core to temporal graph, called the (k, Δ)-core.

Definition 6. *k-core. The* k-core *of a graph $G = (V, E)$ is the maximal subgraph of minimum degree k in G.*

The k-core decomposition is a powerful tool for network analysis [8, 1] : it assigns to each vertex the largest k for which the vertex belongs to a k-core. Previous studies [6] suggested that the latter statistics is correlated to the capacity of spreading information. It is therefore relevant to generalize this property to dynamic networks.

The parameter Δ provides a bound to the temporal connectivity of a TVG (the frequency of interaction). The parameter k in the k-core decomposition provides a bound to the topological connectivity (the degree). We now bring those two constraints together. To do that, observe the definition 6 can be formulated in term of edge subsets, which is useful here since the vertices set changes with Δ while the edge set of a TVG stays the same.

Definition 7. *(k, Δ)-core. Let G be a TVG, the* (k, Δ)-core *of G denoted $C_{k,\Delta}(G)$ is the maximal subset of edges such that the subgraph formed by $C_{k,\Delta}(G)$ has a Δ-duplication where the minimum degree is at least k.*

Notice that $C_{k,\Delta}(G)$ can be an empty subset.

Theorem 3. *Uniqueness. For every temporal graph $G = (V, E, T, t)$, it exists a unique (k, Δ)-core.*

Proof. Suppose it exists two different (k, Δ)-cores of G denoted $C_1 \subset E$ and $C_2 \subset E$ with $C_1 \neq C_2$. Since both C_1 and C_2 are maximal, the graph formed by the union $C_1 \cup C_2$ should not have a Δ-duplication with a minimum degree of k. For a vertex $v \in V$, its incident edges are therefore split into two groups : $C_1(v)$ and $C_2(v)$. Both of these sets can be partitioned into Δ-sessions having at least k elements. Observe merging a couple of Δ-sessions in $C_1(v)$ and $C_2(v)$ that overlaps over their time periods produced a larger Δ-session. Therefore, doing the union of the two sets $C_1(v)$ and $C_2(v)$ and merging the pairs that overlap produce a set of Δ-sessions each having at least k elements. Since it is true for every vertices, it invalidates our hypothesis. We conclude it exists no maximal (k,Δ)-cores C_1 and C_2 such that $C_1 \neq C_2$. □

Theorem 4. *Inclusion. For $\Delta_1 \leq \Delta_2$ and $k_1 \leq k_2$, we have*

$$C_{k_2,\Delta_1}(G) \subseteq C_{k_1,\Delta_2}(G) \tag{1}$$

Proof. According to Property 2, a Δ_1-successive graphs is also Δ_2-successive. For all $\Delta \in [\Delta_1, \Delta_2]$, the edge set $C_{k_2,\Delta_1}(G)$ has a Δ-duplication duplication with minimum degree at least k_2, therefore it has one with minimum degree at least k_1. □

The theorem 4 indicates that a partial order exists between the different cores. It means that one can find the largest k for a non-empty (k,Δ)-core when Δ is fixed. Equivalently, one can find the smallest Δ such that a (k,Δ)-core is non-empty for a given k.

Notice that the concept of k-core is here defined with respect to vertices degrees. It can also be defined for any vertex statistic. For example, the in or out-degree [4], the weighted degree [3] or the number of triangles [10] can also be used. In our case, since temporal graph are generally multiple (*i.e.* multiple interactions appears between two vertices) a relevant statistic could be the number of distinct neighbours.

Algorithm 1: Computation of (k,Δ)-cores

Input: $G = (V, E, T, t)$, k, Δ
Output: $C_{k,\Delta}(G)$
1 $G' \leftarrow G$;
2 **while** $\min_{u \in V(G')} d_{G'}(u) < k$ *and* G' *not* Δ*-successive* **do**
3     $G' \leftarrow$ min-Δ-duplication(G') ;
4     $G' \leftarrow k$-core(G') ;
5 **end**
6 **return** $E(G')$;

Algorithm 1 can be used to compute the (k,Δ)-core of a temporal graph. The procedure alternates between the computation of minimum Δ-duplications (line 3) and the peeling of vertices degree lower than k (line 4) until the conditions given in Definition 7 are met.

Theorem 5. *Correctness of Algorithm 1. For a TVG $G = (V, E, T, t)$, the Algorithm 1 returns $C_{k,\Delta}(G)$.*

Proof. We call degree (resp. core value) of an edge in G the minimum degree (resp. core value) of its endpoints. First, at each iteration of the while loop, we have $C_{k,\Delta}(G) \subseteq E(G')$. The transformation into minimum Δ-duplication can only decrease the core values. Those values are not modified for the edges that

remain after the extraction of the k-core of G'. Moreover, the Δ-duplication for which the core of values of edges is maximum is the minimum Δ-duplication of G'. For every edge $e \in C_{k,\Delta}(G)$, the core value of e is therefore at least k at each iteration and $C_{k,\Delta}(G) \subseteq E(G')$ at the end of the algorithm.
Next, $E(G') \subseteq C_{k,\Delta}(G)$ since G' is a Δ-successive subgraph of G with a minimum degree of k. By definition the edges of G' belong to $C_{k,\Delta}(G)$. Therefore, we have $E(G') = C_{k,\Delta}(G)$.

□

5 Related Works

The temporal node representation described in [9] actually corresponds to a 0-duplication of the temporal graph[2]. However, it is not minimum since a vertex is created for each timestamps in T *i.e.* the function θ has $V \times T$ for input space even if a vertex is not the extremity of an edge at some instant.

Notice our formalism can be extended to more general definitions of time-varying graphs [2]. For example, non-instantaneous interactions can be used *i.e.* when temporal edges corresponds to quadruple $(u, t_1, v, t_2) \in V \times E \times V \times E$ or continuous time intervals $(u, v, [t_1, t_2])$. In the later, the edge (u, v) is active during the period $[t_1, t_2] \subseteq T$. The Δ-duplication depends on the notion of vertex activity and we consider a vertex inactive if it is not the extremity of an edge for a duration of at least Δ. This concept is still valid for the TVG definitions given here although different types of computation may be needed.

In TVG analysis, the transformation into sequences of *snapshots* is often justified if the "time cuts" between the snapshots correspond to time periods of inactivity in the networks (for example during the night in a communications network within the same time zone). In this case, the transformation is not likely to break relevant patterns. If a TVG exhibits such property *i.e.* regular periods with no interactions then those cuts can be recovered by the Δ-duplication looking at the connected components of the static graph obtained by ignoring the timestamps of the edges.

Property 3. *Connected components of Δ-duplication. Let $G = (V, E, T, t)$ be a TVG and $t_1 < t_2$ such that there is no edge $e \in E$, with $t(e) \in [t_1, t_2]$. The $(t_2 - t_1)$-duplication of G contains at least two connected components (in the static and temporal sense).*

According to Property 3, having $\Delta \leq (t_2 - t_1)$ will disconnect the TVG over the time period. Therefore, if it exists multiple cuts of length at least $(t_2 - t_1)$, the connected components of the $(t_2 - t_1)$-duplication will correspond to the transformation into snapshots. The Δ-duplication is however more flexible since

- the period of inactivity in the network can have different length
- the transformation into snapshot requires another (hidden) parameter t_0 which is the time of the first snapshot (most of the time $t_0 = 0$).

[2]Assuming there is no directed edge between successive temporal node corresponding to the same node.

Notice the two items given above can be achieved by creating a new snapshot when no interaction is observed for a duration of at least Δ timestamps. This leads to a transformation of a TVG into a sequence of static graphs with (non-overlapping) time windows of different length. The Δ-duplication is however vertex-centred, meaning a vertex may be duplicated even if there are simultaneous activities in the networks that does not affect the vertex directly. Therefore, the Δ-duplications allows finding "non-vertical" time cuts.

6 Conclusion and Future Directions

We introduced in this paper a transformation of a time-varying graph that can be used to handle time heterogeneity. The idea is to consider couples of vertices and sessions. The latter correspond to periods of quasi-continuous activity (defined using a parameter Δ) of an individual in the network.
We described properties of this transformation and discuss computational aspects. A generalisation of k-core to TVG analysis was also proposed. Although different concepts could also be extended in the context of Δ-duplication, we focus here of the new notion of (k, Δ)-core as its static counterpart is widely used in network analysis for a variety of applications.

Our objective is to use Δ-duplication and (k, Δ)-cores in the practical context of dynamic networks analysis. In particular, we want to compare the information obtained or the pattern detected by comparing them to the conclusion of studies based on a transformation into snapshots of fixed or varying length.

References

[1] James Abello and François Queyroi. Fixed points of graph peeling. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 256–263. ACM, 2013.

[2] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. In *Ad-hoc, Mobile, and Wireless Networks*, pages 346–359. Springer, 2011.

[3] C. Giatsidis, D.M. Thilikos, and M. Vazirgiannis. Evaluating cooperation in communities with the k-core structure. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 87–93. IEEE, 2011.

[4] Christos Giatsidis, Dimitrios M Thilikos, and Michalis Vazirgiannis. D-cores: measuring collaboration of directed graphs based on degeneracy. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 201–210. IEEE, 2011.

[5] John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. Tracking evolving communities in large linked networks. *Proceedings of the national academy of sciences of the United States of America*, 101(Suppl 1):5249–5253, 2004.

[6] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893, 2010.

[7] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187. ACM, 2005.

[8] S.B. Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.

[9] Klaus Wehmuth, Artur Ziviani, and Eric Fleury. A Unifying Model for Representing Time-Varying Graphs. Research Report RR-8466, INRIA, February 2014.

[10] Yang Zhang and Srinivasan Parthasarathy. Extracting analyzing and visualizing triangle k-core motifs within networks. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1049–1060. IEEE, 2012.