Temporal Core Decomposition

Francesco Gullo

UniCredit, Rome, Italy, gullof@acm.org

joint work with A. Barrat, F. Bonchi, C. Cattuto, M. Ciaperoni, E. Galimberti

Algorithmic Aspects of Temporal Graphs IV
Satellite workshop of ICALP 2021 Conference
Glasgow, Scotland, UK
July 12, 2021

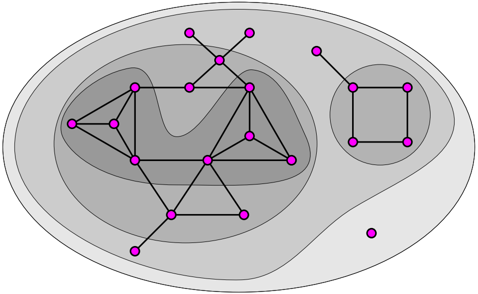
(held online due to the COVID-19 pandemic)

Background

Core decomposition

Definition

*The **k-core** (or core of order k) of a (non-temporal) graph $G = (V, E)$ is a maximal set of vertices $C_k \subseteq V$ such that $\forall u \in C_k : deg(C_k, u) \geq k$.*
*The set of all k-cores $V = C_0 \supseteq C_1 \supseteq \cdots \supseteq C_{k^*}$ is the **core decomposition** of G.*

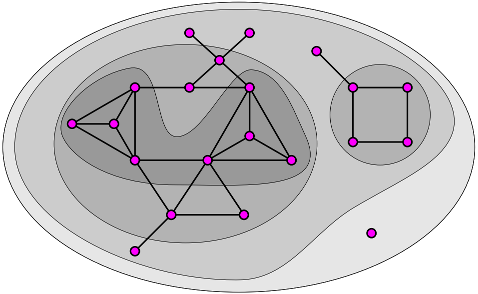


- exact **linear-time** algorithm
- important tool to analyze and **visualize** networks
- **speed-ups** the extraction of dense subgraphs
- at the basis of **approximation algorithms** for, e.g., densest subgraph betweenness centrality

Core decomposition

Definition

The **k-core** *(or core of order k) of a (non-temporal) graph $G = (V, E)$ is a maximal set of vertices $C_k \subseteq V$ such that $\forall u \in C_k : deg(C_k, u) \geq k$.*
The set of all k-cores $V = C_0 \supseteq C_1 \supseteq \cdots \supseteq C_{k^*}$ is the **core decomposition** *of G.*



- exact **linear-time** algorithm
- important tool to analyze and **visualize** networks
- **speed-ups** the extraction of dense subgraphs
- at the basis of **approximation algorithms** for, e.g., densest subgraph betweenness centrality

Temporal graphs

- A **temporal graph** is a representation of
  - **entities** (vertices)
  - their **relations** (links)
  - how these relations are **established/broken along time**

Temporal graphs

Definition

A **temporal graph** *is a triple* $G = (V, T, \tau)$, *where*

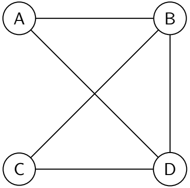
- V *is a set of vertices,*
- $T = [t_0, t_1, \ldots, t_{max}] \subseteq \mathbb{N}$ *is a discrete time domain,*
- $\tau : V \times V \times T \to \{0, 1\}$ *is a function defining for each* $u, v \in V$ *and each* $t \in T$ *whether edge* (u, v) *exists in* t.

Temporal graphs

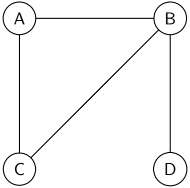
Definition

A **temporal graph** *is a triple* $G = (V, T, \tau)$, *where*

- V *is a set of vertices,*
- $T = [t_0, t_1, \ldots, t_{max}] \subseteq \mathbb{N}$ *is a discrete time domain,*
- $\tau : V \times V \times T \to \{0, 1\}$ *is a function defining for each* $u, v \in V$ *and each* $t \in T$ *whether edge* (u, v) *exists in* t.



t_0



t_1

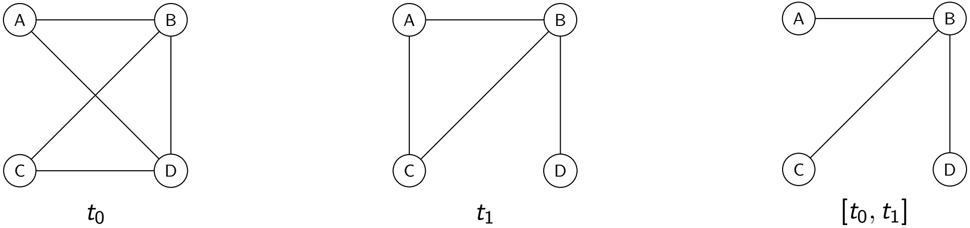
$[t_0, t_1]$

Temporal graphs

Definition

A **temporal graph** *is a triple* $G = (V, T, \tau)$*, where*

- V *is a set of vertices,*
- $T = [t_0, t_1, \ldots, t_{max}] \subseteq \mathbb{N}$ *is a discrete time domain,*
- $\tau : V \times V \times T \to \{0, 1\}$ *is a function defining for each* $u, v \in V$ *and each* $t \in T$ *whether edge* (u, v) *exists in* t*.*



Span-core decomposition

Motivation

Extracting **dense structures** together with their **temporal span** is a key mining primitive

- anomaly detection in proximity networks
- quantify the transmission opportunities of respiratory infections
- identify events and buzzing stories
- understand the dynamics of collaboration in successful professional teams

Span-core decomposition

Definition

The $(\mathbf{k}, \boldsymbol{\Delta})$**-core** *of a temporal graph* $G = (V, T, \tau)$ *is a maximal and non-empty set of vertices* $\emptyset \neq C_{k,\Delta} \subseteq V$, *such that* $\forall u \in C_{k,\Delta} : deg_\Delta(C_{k,\Delta}, u) \geq k$, *where* $\Delta \sqsubseteq T$ *is a temporal interval and* $k \in \mathbb{N}^+$.

- $deg_\Delta(C_{k,\Delta}, u)$ represents the **degree of a vertex** u **in the subgraph induced by** $C_{k,\Delta}$ **within the temporal interval** Δ

Problem

Given a temporal graph G, *find the set of all* (k, Δ)*-cores of* G.

- the number of span-cores is $\mathcal{O}(|T|^2)$

Span-core decomposition

Definition

The $(\mathbf{k}, \boldsymbol{\Delta})$**-core** *of a temporal graph* $G = (V, T, \tau)$ *is a maximal and non-empty set of vertices* $\emptyset \neq C_{k,\Delta} \subseteq V$*, such that* $\forall u \in C_{k,\Delta} : deg_{\Delta}(C_{k,\Delta}, u) \geq k$*, where* $\Delta \sqsubseteq T$ *is a temporal interval and* $k \in \mathbb{N}^{+}$*.*

- $deg_{\Delta}(C_{k,\Delta}, u)$ represents the **degree of a vertex** u **in the subgraph induced by** $C_{k,\Delta}$ **within the temporal interval** Δ

Problem

Given a temporal graph G*, find the set of all* (k, Δ)*-cores of* G*.*

- the number of span-cores is $\mathcal{O}(|T|^2)$

A naïve approach

Algorithm

- generate all temporal intervals $\Delta \sqsubseteq T$
- for each $\Delta \sqsubseteq T$, compute the subgraph $G_\Delta = (V, E_\Delta)$
- run a core-decomposition subroutine on each G_Δ

- $\mathcal{O}(|T|^2 \times |E|)$ time complexity

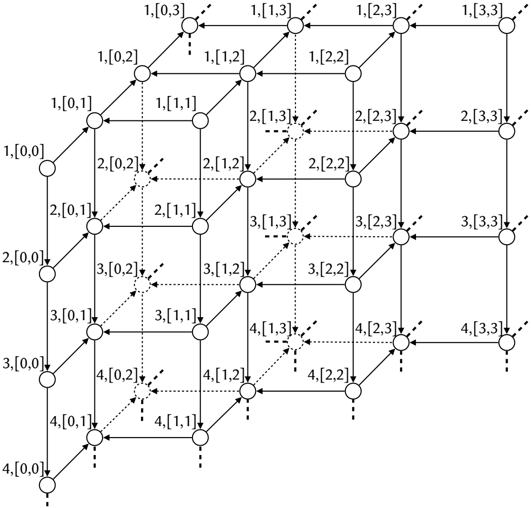
A naïve approach

Algorithm

- generate all temporal intervals $\Delta \sqsubseteq T$
- for each $\Delta \sqsubseteq T$, compute the subgraph $G_\Delta = (V, E_\Delta)$
- run a core-decomposition subroutine on each G_Δ

- $\mathcal{O}(|T|^2 \times |E|)$ time complexity

Span-core search space



Proposition

For any two span-cores $C_{k,\Delta}$, $C_{k',\Delta'}$ of a temporal graph G it holds that

$$k' \leq k \wedge \Delta' \sqsubseteq \Delta \;\Rightarrow\; C_{k,\Delta} \subseteq C_{k',\Delta'}.$$

Corollary

Given a temporal graph $G = (V, T, \tau)$, and a temporal interval $\Delta = [t_s, t_e] \sqsubseteq T$, let $\Delta_+ = [\min\{t_s + 1, t_e\}, t_e]$ and $\Delta_- = [t_s, \max\{t_e - 1, t_s\}]$. It holds that

$$C_{k,\Delta} \;\subseteq\; (C_{k,\Delta_+} \cap C_{k,\Delta_-}) \;=\; \bigcap_{\Delta' \sqsubseteq \Delta} C_{k,\Delta'}.$$

A more efficient algorithm

Algorithm

- generate temporal intervals $\Delta \sqsubseteq T$ of **increasing** size
- for each $\Delta \sqsubseteq T$ such that $|\Delta| > 1$, run a core-decomposition subroutine from $(C_{1,\Delta_+} \cap C_{1,\Delta_-})$
- if C_{1,Δ_+} or C_{1,Δ_-} does not exist, skip the core decomposition for Δ

- worst-case time complexity still $\mathcal{O}(|T|^2 \times |E|)$, but the algorithm is **much faster in practice** than the naïve one

A more efficient algorithm

Algorithm

- generate temporal intervals $\Delta \sqsubseteq T$ of **increasing** size
- for each $\Delta \sqsubseteq T$ such that $|\Delta| > 1$, run a core-decomposition subroutine from $(C_{1,\Delta_+} \cap C_{1,\Delta_-})$
- if C_{1,Δ_+} or C_{1,Δ_-} does not exist, skip the core decomposition for Δ

- worst-case time complexity still $\mathcal{O}(|T|^2 \times |E|)$, but the algorithm is **much faster in practice** than the naïve one

Maximal span-cores

Maximal span-cores

Definition

A span-core $C_{k,\Delta}$ of a temporal graph G is said **maximal** *if there does not exist any other span-core $C_{k',\Delta'}$ of G such that $k \leq k'$ and $\Delta \sqsubseteq \Delta'$.*

Problem

Given a temporal graph G, find the set of all maximal (k, Δ)-cores of G.

- the number of maximal span-cores is $\mathcal{O}(|T|^2)$
- experimentally, maximal span-cores are **at least one order of magnitude less** than the overall span-cores

Maximal span-cores

Definition

A span-core $C_{k,\Delta}$ of a temporal graph G is said **maximal** *if there does not exist any other span-core $C_{k',\Delta'}$ of G such that $k \leq k'$ and $\Delta \sqsubseteq \Delta'$.*

Problem

Given a temporal graph G, find the set of all maximal (k, Δ)-cores of G.

- the number of maximal span-cores is $\mathcal{O}(|T|^2)$
- experimentally, maximal span-cores are **at least one order of magnitude less** than the overall span-cores

A filtering-based (naïve) approach

Algorithm

- equip the algorithm for span-core decomposition with a data structure $\mathcal{M}$ that
  - stores the span-core of the highest order for every temporal interval $\Delta \sqsubseteq T$
  - at the storage of a span-core $C_{k,\Delta}$, removes the span-cores dominated by $C_{k,\Delta}$
- return the span-cores retained by $\mathcal{M}$

- same running time as the algorithm for finding all the span-cores

A filtering-based (naïve) approach

Algorithm

- equip the algorithm for span-core decomposition with a data structure $\mathcal{M}$ that
  - stores the span-core of the highest order for every temporal interval $\Delta \sqsubseteq T$
  - at the storage of a span-core $C_{k,\Delta}$, removes the span-cores dominated by $C_{k,\Delta}$
- return the span-cores retained by $\mathcal{M}$

- same running time as the algorithm for finding all the span-cores

Properties of maximal span-cores

Lemma

Given a temporal graph $G = (V, T, \tau)$, let $\mathbf{C}_M$ be the set of all maximal span-cores of G, and $\mathbf{C}_{\mathbf{inner}} = \{C_{k^*}[G_\Delta] \mid \Delta \sqsubseteq T\}$ be the set of innermost cores of all graphs G_Δ. It holds that $\mathbf{C}_M \subseteq \mathbf{C}_{\mathbf{inner}}$.

- $\Delta = [t_s, t_e]$ yields a maximal span-core it suffices to start from a subgraph, which is composed of all the vertices whose temporal degree is larger than the maximum between the orders of the innermost cores of intervals $\Delta' = [t_s - 1, t_e]$ and $\Delta'' = [t_s, t_e + 1]$
- **Top-down** strategy: start from larger temporal intervals
- This also allows us to skip the computation of complete core decompositions of the whole "singleton-interval" graphs $\{G_{[t,t]}\}_{t \in T}$

Properties of maximal span-cores

Lemma

Given a temporal graph $G = (V, T, \tau)$, let $\mathbf{C}_M$ be the set of all maximal span-cores of G, and $\mathbf{C}_{\mathbf{inner}} = \{C_{k^*}[G_\Delta] \mid \Delta \sqsubseteq T\}$ be the set of innermost cores of all graphs G_Δ. It holds that $\mathbf{C}_M \subseteq \mathbf{C}_{\mathbf{inner}}$.

- $\Delta = [t_s, t_e]$ yields a maximal span-core it suffices to start from a subgraph, which is composed of all the vertices whose temporal degree is larger than the maximum between the orders of the innermost cores of intervals $\Delta' = [t_s - 1, t_e]$ and $\Delta'' = [t_s, t_e + 1]$
- **Top-down** strategy: start from larger temporal intervals
- This also allows us to skip the computation of complete core decompositions of the whole "singleton-interval" graphs $\{G_{[t,t]}\}_{t \in T}$

Properties of maximal span-cores

Lemma

Given a temporal graph $G = (V, T, \tau)$*, and three temporal intervals* $\Delta = [t_s, t_e] \sqsubseteq T$*,* $\Delta' = [t_s - 1, t_e] \sqsubseteq T$*, and* $\Delta'' = [t_s, t_e + 1] \sqsubseteq T$*. The innermost core* $C_{k^*}[G_\Delta]$ *is a maximal span-core of* G *if and only if* $k^* > \max\{k', k''\}$ *where* k' *and* k'' *are the orders of the innermost cores of* $G_{\Delta'}$ *and* $G_{\Delta''}$*, respectively.*

Lemma

Given G*,* Δ*,* Δ'*,* Δ''*,* k'*, and* k'' *as in previous Lemma, let* $\widetilde{V} = \{u \in V \mid deg_\Delta(V, u) > \max\{k', k''\}\}$*, and let* $C_{k^*}[G_\Delta[\widetilde{V}]]$ *be the innermost core of* $G_\Delta[\widetilde{V}]$*. If* $k^* > \max\{k', k''\}$*, then* $C_{k^*}[G_\Delta[\widetilde{V}]]$ *is a maximal span-core; otherwise, no maximal span-core exists for* Δ*.*

Efficient maximal-span-core finding

Algorithm

- consider intervals $\Delta = [t_s, t_e] \sqsubseteq T$, for increasing values of t_s and decreasing values of t_e
  - e.g., with $t_{max} = 10$, $\{[0,10],[0,9],\ldots,[0,0],[1,10],[1,9],\ldots,[1,1],[2,10],[2,9],\ldots\}$
  - this guarantees that once we consider Δ, no $\Delta' \sqsupseteq \Delta$ will be considered at later stage
- compute the **lower bound lb on the order** of a span-core in Δ to be recognized as maximal
- build the sets of vertices V_{lb} that have degree in Δ larger than lb
- extract the **innermost** core of the subgraph $(V_{lb}, E_\Delta[V_{lb}])$
- identify such a core as maximal if its order is actually larger than lb

- running time **much faster in practice** than the filtering-based algorithm

Efficient maximal-span-core finding

Algorithm

- consider intervals $\Delta = [t_s, t_e] \sqsubseteq T$, for increasing values of t_s and decreasing values of t_e
  - e.g., with $t_{max} = 10$, $\{[0, 10], [0, 9], \ldots, [0, 0], [1, 10], [1, 9], \ldots, [1, 1], [2, 10], [2, 9], \ldots\}$
  - this guarantees that once we consider Δ, no $\Delta' \sqsupseteq \Delta$ will be considered at later stage
- compute the **lower bound lb on the order** of a span-core in Δ to be recognized as maximal
- build the sets of vertices V_{lb} that have degree in Δ larger than lb
- extract the **innermost** core of the subgraph $(V_{lb}, E_\Delta[V_{lb}])$
- identify such a core as maximal if its order is actually larger than lb

- running time **much faster in practice** than the filtering-based algorithm

Experiments

Datasets

dataset	$\|V\|$	$\|E\|$	$\|T\|$	window size (days)	domain
ProsperLoans	89k	3M	307	7	economic
Last.fm	992	4M	77	21	co-listening
WikiTalk	2M	10M	192	28	communication
DBLP	1M	11M	80	366	co-authorship
StackOverflow	2M	16M	51	56	question-and-answer
Wikipedia	343k	18M	101	56	co-editing
Amazon	2M	22M	115	28	co-rating
Epinions	120k	33M	25	21	co-rating

Evaluation

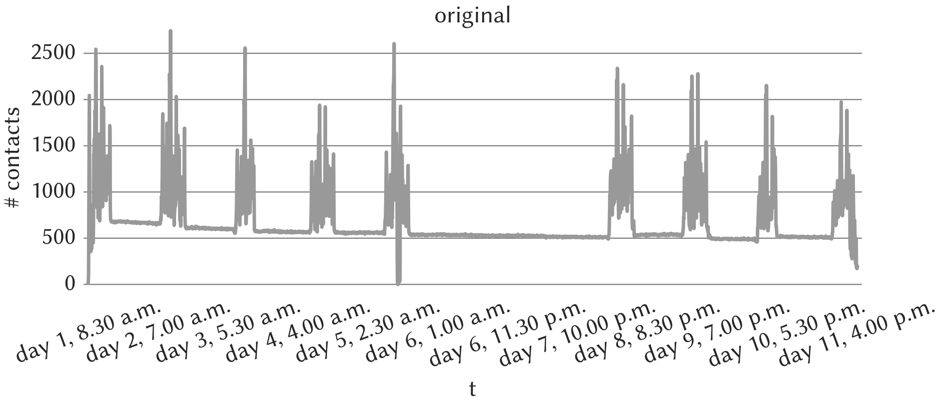
dataset	method	# output span-cores	time (s)	memory (GB)	# processed vertices
WikiTalk	Naïve-span-cores	19 693	322 302	36	25B
	Span-cores		1 084	36	555M
	Naïve-maximal-span-cores	632	1 194	36	555M
	Maximal-span-cores		126	35	2M
Wikipedia	Naïve-span-cores	125 191	17 155	4	1B
	Span-cores		522	4	35M
	Naïve-maximal-span-cores	2 147	537	4	35M
	Maximal-span-cores		201	4	320k
Amazon	Naïve-span-cores	29 318	10 415	18	2B
	Span-cores		409	18	247M
	Naïve-maximal-span-cores	303	580	18	247M
	Maximal-span-cores		123	18	688k

Applications

Datasets

- **face-to-face interaction networks** gathered by a proximity-sensing infrastructure in schools
  - PrimarySchool (242 individuals, 2 days)
  - HighSchool (327 individuals, 5 days)
  - HongKong (774 individuals, 11 days)
- window size of 5 minutes
- discarded span-cores of $|\Delta| = 1$

Anomaly detection

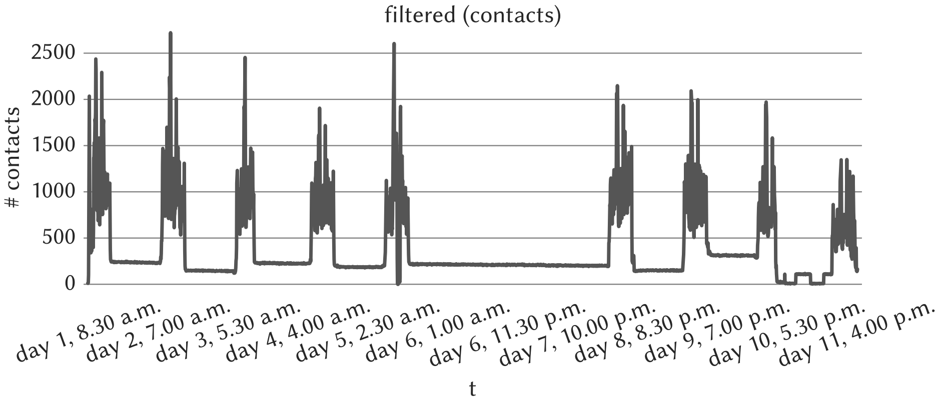


HongKong

Anomaly detection

1. find a set of anomalously long temporal intervals supporting maximal span-cores
   - find the set of temporal intervals $\mathcal{I} = \{\Delta \sqsubseteq T \mid C_{k,\Delta} \in \mathbf{C}_M \wedge |\Delta| > tr\}$ that are the span of a maximal span-core $C_{k,\Delta}$ with size longer than a certain threshold tr
   - we use $tr = 22$ (110 minutes)
2. identify anomalous vertices
   - for each timestamp $t \in T$, select as anomalous all those vertices that appear in the span-cores $\{C_{1,\Delta} \mid \Delta \in \mathcal{I} \wedge t \in \Delta\}$, i.e., the span-cores of $k = 1$ whose span is in $\mathcal{I}$ and contains t
3. filter out anomalous contacts
   - at each timestamp $t \in T$, filter out the contacts having at least an anomalous endpoint at time t.

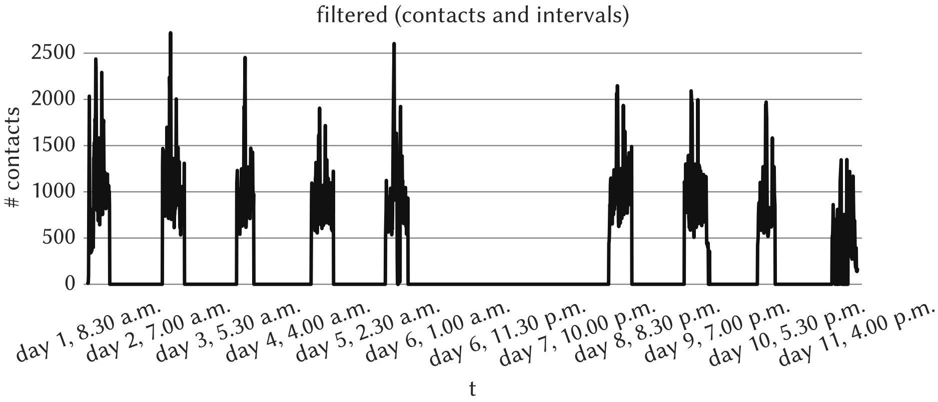
Anomaly detection



HongKong

- 0.91 precision, 0.64 recall

Anomaly detection



HongKong

- 0.93 precision, 0.99 recall

Conclusions

- introduced a notion of dense pattern in temporal networks that
  - takes into account the **sequentiality** of connections
  - is assigned with a clear **temporal collocation**
- developed efficient algorithms for computing all the span-cores, and only the maximal ones
- future work:
  - spreading processes analysis
  - temporal community search and temporal densest subgraph
  - network finger-printing

References

- E. Galimberti, M. Ciaperoni, A. Barrat, F. Bonchi, C. Cattuto, F. Gullo. Span-core Decomposition for Temporal Networks: Algorithms and Applications. ACM Transactions on Knowledge Discovery from Data (TKDD), 2020, ONLINE
- M. Ciaperoni, E. Galimberti, F. Bonchi, C. Cattuto, F. Gullo, A. Barrat. Relevance of temporal cores for epidemic spread in temporal networks. Scientific Reports (SciRep), 2020, ONLINE
- E. Galimberti, A. Barrat, F. Bonchi, C. Cattuto, F. Gullo. Mining (maximal) Span-cores from Temporal Networks. In Proceedings of the ACM International Conference on Knowledge and Information Management (CIKM '18), pp. 107-116. Turin, Italy, October 22-26, 2018

Thanks!