Inferring Tie Strength in Temporal Networks

(ECML PKDD 2022)

Lutz Oettershagen[1], Athanasios L. Konstantinidis[2], **Giuseppe F. Italiano**[2]

September 19, 2023

1) KTH Royal Institute of Technology, Stockholm, Sweden
`lutzo@kth.se`

2) Luiss University, Rome, Italy
`{akonstantinidis, gitaliano}@luiss.it`

Outline

Outline

Outline

Outline

Outline

Social networks are inherently dynamic:[1]

- 4.76 billion social media users,
  ca. 60% of the world's total population
- Digital's share of total global ad spend was 73.3% in 2022
- Social media is now the primary vehicle for digital discovery
- 53.9% of users are concerned about misinformation

Temporal network data analyses:

- Insights can lead to huge societal and monetary impacts
- Requires specialized tools and techniques for handling the dynamic nature of the data

[1] Sources: wearesocial.com, its.ae/social-media-marketing, parse.ly

Introduction and Motivation

Social networks are inherently dynamic:[1]

- 4.76 billion social media users,
  ca. 60% of the world's total population
- Digital's share of total global ad spend was 73.3% in 2022
- Social media is now the primary vehicle for digital discovery
- 53.9% of users are concerned about misinformation

Temporal network data analyses:

- Insights can lead to huge societal and monetary impacts
- Requires specialized tools and techniques for handling the dynamic nature of the data

[1] Sources: wearesocial.com, its.ae/social-media-marketing, parse.ly

Introduction and Motivation

Social networks are inherently dynamic:[1]

- 4.76 billion social media users,
  ca. 60% of the world's total population
- Digital's share of total global ad spend was 73.3% in 2022
- Social media is now the primary vehicle for digital discovery
- 53.9% of users are concerned about misinformation

Temporal network data analyses:

- Insights can lead to huge societal and monetary impacts
- Requires specialized tools and techniques for handling the dynamic nature of the data

[1] Sources: wearesocial.com, its.ae/social-media-marketing, parse.ly

Introduction and Motivation

Social networks are inherently dynamic:[1]

- 4.76 billion social media users,
  ca. 60% of the world's total population
- Digital's share of total global ad spend was 73.3% in 2022
- Social media is now the primary vehicle for digital discovery
- 53.9% of users are concerned about misinformation

Temporal network data analyses:

- Insights can lead to huge societal and monetary impacts
- Requires specialized tools and techniques for handling the dynamic nature of the data

[1] Sources: wearesocial.com, its.ae/social-media-marketing, parse.ly

Introduction and Motivation

Social networks are inherently dynamic:[1]

- 4.76 billion social media users, ca. 60% of the world's total population
- Digital's share of total global ad spend was 73.3% in 2022
- Social media is now the primary vehicle for digital discovery
- 53.9% of users are concerned about misinformation

Temporal network data analyses:

- Insights can lead to huge societal and monetary impacts
- Requires specialized tools and techniques for handling the dynamic nature of the data

[1] Sources: wearesocial.com, its.ae/social-media-marketing, parse.ly

Introduction and Motivation

Social networks are inherently dynamic:[1]

- 4.76 billion social media users,
  ca. 60% of the world's total population
- Digital's share of total global ad spend was 73.3% in 2022
- Social media is now the primary vehicle for digital discovery
- 53.9% of users are concerned about misinformation

Temporal network data analyses:

- Insights can lead to huge societal and monetary impacts
- Requires specialized tools and techniques for handling the dynamic nature of the data

[1]Sources: wearesocial.com, its.ae/social-media-marketing, parse.ly

Introduction and Motivation

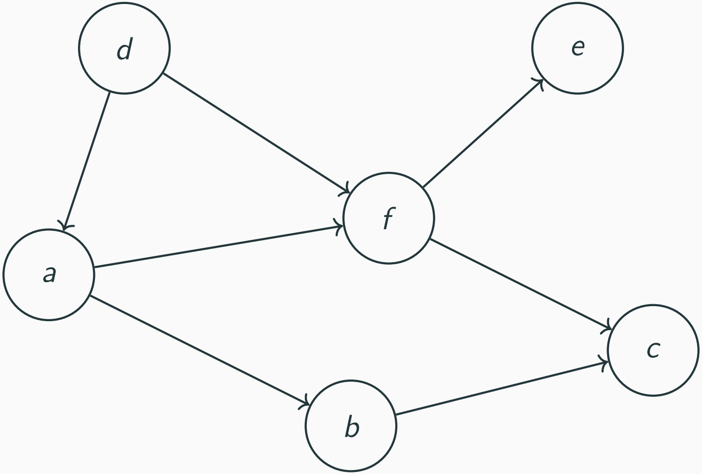
Social networks are inherently dynamic:[1]

- 4.76 billion social media users,
  ca. 60% of the world's total population
- Digital's share of total global ad spend was 73.3% in 2022
- Social media is now the primary vehicle for digital discovery
- 53.9% of users are concerned about misinformation

Temporal network data analyses:

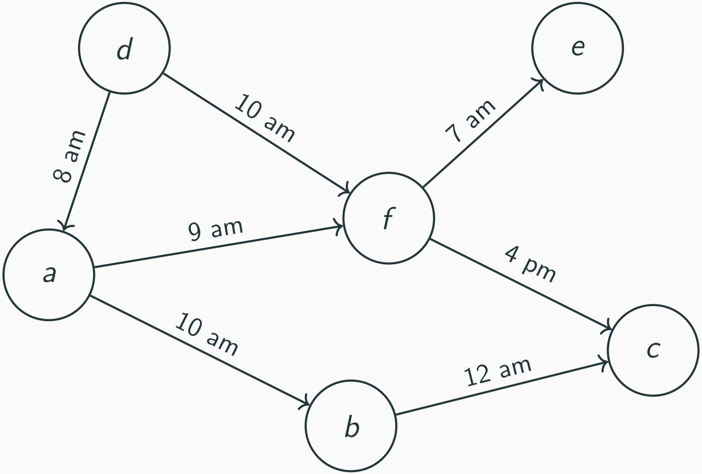
- Insights can lead to huge societal and monetary impacts
- Requires specialized tools and techniques for handling the dynamic nature of the data

[1] Sources: wearesocial.com, its.ae/social-media-marketing, parse.ly



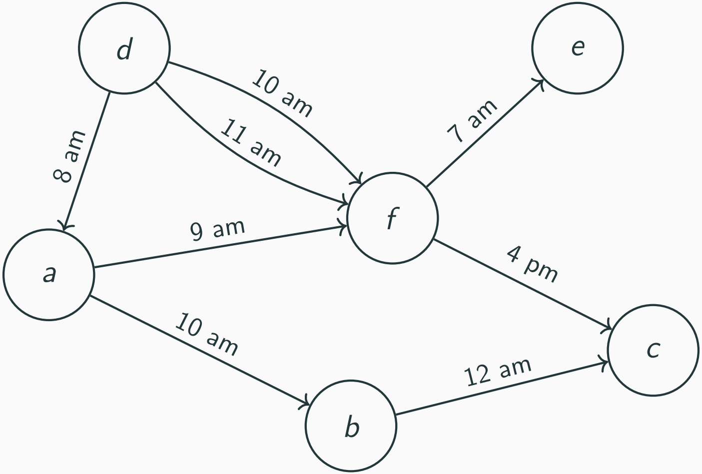
Static directed graph $G = (V, E)$ with edges $(u, v) \in E$

Temporal Networks



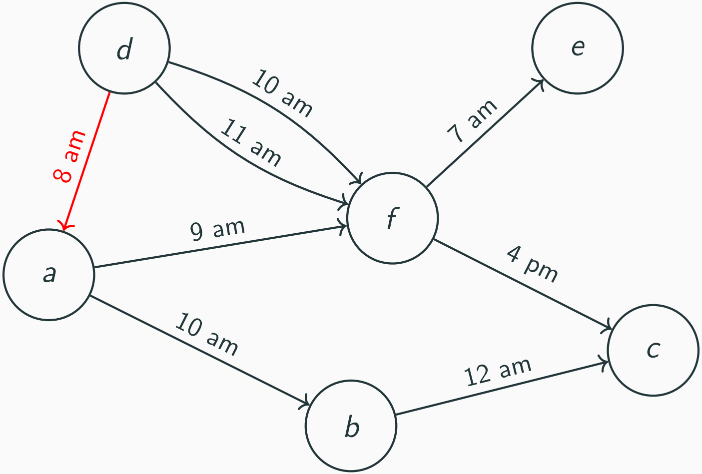
Temporal graph $\mathcal{G} = (V, \mathcal{E})$ with temporal edges $(u, v, t) \in \mathcal{E}$

Temporal Networks



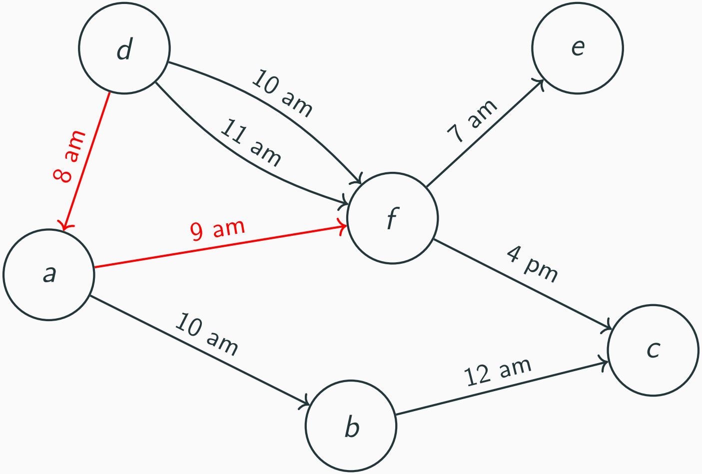
Temporal graph $\mathcal{G} = (V, \mathcal{E})$ with temporal edges $(u, v, t) \in \mathcal{E}$

Temporal Networks



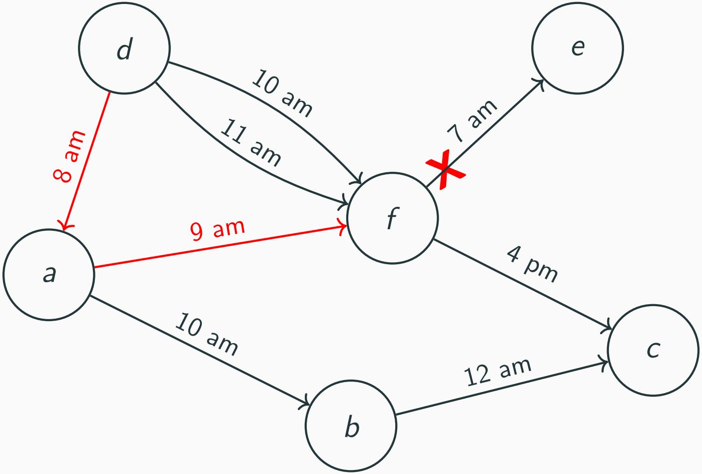
Temporal graph $\mathcal{G} = (V, \mathcal{E})$ with temporal edges $(u, v, t) \in \mathcal{E}$

Temporal Networks



Temporal graph $\mathcal{G} = (V, \mathcal{E})$ with temporal edges $(u, v, t) \in \mathcal{E}$

Temporal Networks



$a \rightarrow f$

$f \rightarrow e$

$a \not\rightarrow e$

Temporal graph $\mathcal{G} = (V, \mathcal{E})$ with temporal edges $(u, v, t) \in \mathcal{E}$

Temporal Networks

Real-world examples:

- Social networks
- Communication networks
- Transportation networks
- Biological networks
- Many more...

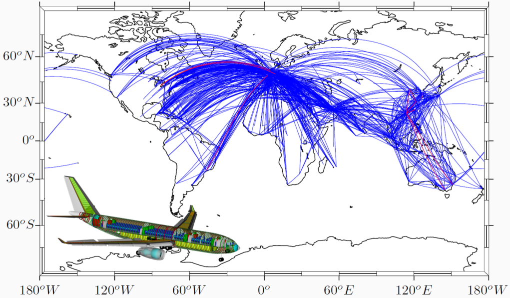
Temporal Networks

Real-world examples:

- Social networks
- Communication networks
- Transportation networks
- Biological networks
- Many more...

Real-world examples:

- Social networks
- Communication networks
- Transportation networks
- Biological networks
- Many more...

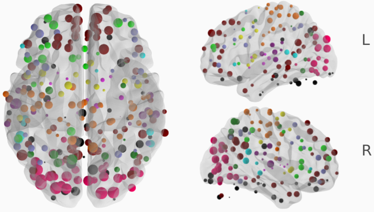


Source: DLR Air Transportation Systems (dlr.de, 22/06/22)

Temporal Networks

Real-world examples:

- Social networks
- Communication networks
- Transportation networks
- Biological networks
- Many more...

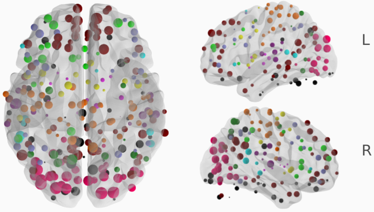


Source: Thompson et al. 2017

Temporal Networks

Real-world examples:

- Social networks
- Communication networks
- Transportation networks
- Biological networks
- Many more...



Source: Thompson et al. 2017

Temporal Networks

Real-world examples:

- Social networks
- Communication networks
- Transportation networks
- Biological networks
- Many more...

Graph Mining Problem:

- **Tie strength inference:** Which ties are strong or weak? Good friend or just acquaintance?

Temporal Networks

Real-world examples:

- Social networks
- Communication networks
- Transportation networks
- Biological networks
- Many more...

Graph Mining Problem:

- **Tie strength inference:** Which ties are strong or weak? Good friend or just acquaintance?

The Strength of Weak Ties

Motivation:

- **Tie strength inference** gained increasing attention since pioneering work of Granovetter[2]
- People with strong ties share similar information and experiences
- Weak ties provide access to new and different information and experiences
- Automated inference of tie strengths is critical for many applications, e.g., advertisement, information dissemination, understanding of complex human behavior, etc.

[2] Granovetter, Mark S. *The strength of weak ties.* American journal of sociology 78.6 (1973): 1360-1380.

Strong Triadic Closure (STC)

People are more likely to get acquainted over time when they have something in common

- We have a bias towards the familiar, thus reducing the pure randomness of connections
- Known as Homophily (*"Birds of a feather flock together"*)

Network connections do not arise independently of each other

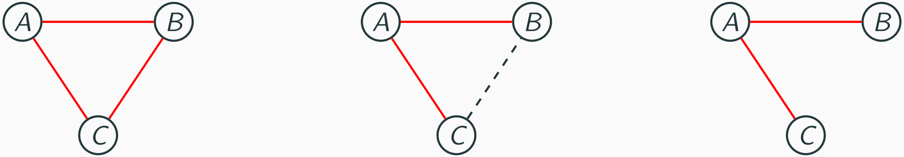
- ... they are influenced by previous connections

If A knows B...

- ... and A knows C
- ... then B is more likely to know C (or at least A has an incentive to let B and C know each other)

Strong Triadic Closure (STC)

- If for three nodes, A, B, and C, there are strong ties between A and B, as well as A and C, there has to be a (weak or strong) tie between B and C



(a) Fulfills STC. **(b)** Fulfills STC. **(c)** Does not fulfill STC.

(An extensive analysis of STC can be found in the book of [Easley and Kleinberg, 2010])

Problem and Contributions

At a very high level

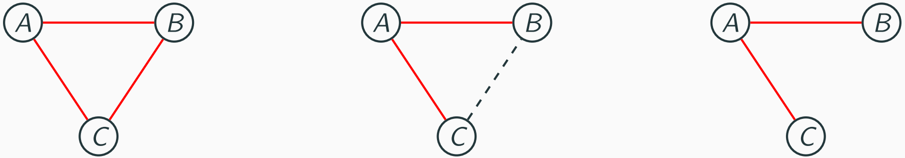
Given a **temporal network** (e.g., a dynamically evolving social network, a communication network, etc...), we wish to **infer the strength of relations between nodes.**

- We introduce a **weighted** version of the **strong triadic closure**
- Provide efficient **streaming algorithm** to approximate the **tie strength over time**

Inferring Tie Strength

We wish to label each tie weak or strong, e.g., good friend vs. acquaintance in social network so as to respect strong triadic closure (STC):

- If for three nodes, A, B, and C, there are strong ties between A and B, as well as between A and C, there has to be a (weak or strong) tie between B and C



(a) Fulfills STC. **(b)** Fulfills STC. **(c)** Does not fulfill STC.

(Introduced by [Sintos and Tsaparas, 2014])

Strong Triadic Closure (STC)

More formally:

- Given a (static) graph $G = (V, E)$, we can assign one of the labels weak or strong to each edge in $e \in E$.
- We call such a labeling a strong-weak labeling, and we specify the labeling by a subset $S \subseteq E$.
- Each edge $e \in S$ is called strong, and $e \in E \setminus S$ weak.
- The strong triadic closure (STC) of a graph G is a strong-weak labeling $S \subseteq E$ such that for any two strong edges $\{u, v\} \in S$ and $\{v, w\} \in S$, there is a (weak or strong) edge $\{u, w\} \in E$.
- We say that such a labeling fulfills the strong triadic closure.

Strong Triadic Closure (STC)

Decision problem for STC is denoted by $\textsc{MaxSTC}$:

- Given a graph $G = (V, E)$ and a non-negative integer k, does there exist $S \subseteq E$ that fulfills the strong triadic closure and $|S| \geq k$?

Equivalently, can define the problem based on weak edges, $\textsc{MinSTC}$:

- Given a graph $G = (V, E)$ and a non-negative integer ℓ, does there exist $E' \subseteq E$ that $E \setminus E'$ fulfills the strong triadic closure and $|E'| \leq \ell$?

Both $\textsc{MaxSTC}$ and $\textsc{MinSTC}$ are NP-hard [Sintos and Tsaparas, 2014]

Strong Triadic Closure with Edge Additions (STC+)

For this problem, apart from labeling the edges as strong or weak, one can add new (weak) edges between non-adjacent nodes.

Denote problem by MinSTC+:

- Given a graph $G = (V, E)$ and a non-negative integer ℓ. Does there exist a set $F \subseteq \binom{V}{2} \setminus E$ such that there is a $E' \subseteq E$ that $E \setminus E'$ fulfills the strong triadic closure and $|E' \cup F| \leq \ell$?

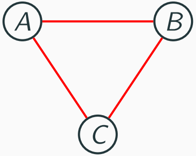
Adding a few edges can improve the labeling hugely
(complete graph with exactly one edge missing, going from $(n-2)$ weak edges to 1)

MinSTC+ is NP-hard [Sintos and Tsaparas, 2014]

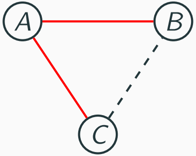
Strong Triadic Closure (Recap)

Strong Triadic Closure (STC)

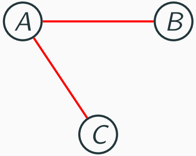
- If for three nodes, A, B, and C, there are strong ties between A and B, as well as A and C, there has to be a (weak or strong) tie between B and C



(a) Fulfills STC.



(b) Fulfills STC.



(c) Does not fulfill STC.

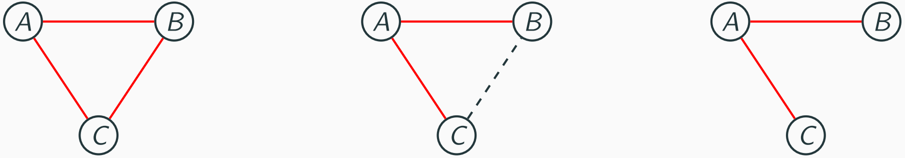
(Optimization problem) Finding edge labelling that fulfils STC with max (resp. min) number of strong (resp. weak) edges is NP-hard

Approximate minimum number of weak edges [Sintos and Tsaparas, 2014]

Strong Triadic Closure (Recap)

Strong Triadic Closure (STC)

- If for three nodes, A, B, and C, there are strong ties between A and B, as well as A and C, there has to be a (weak or strong) tie between B and C



(a) Fulfills STC. **(b)** Fulfills STC. **(c)** Does not fulfill STC.

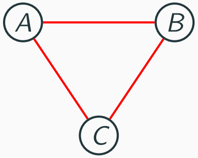
(Optimization problem) Finding edge labelling that fulfils STC with max (resp. min) number of strong (resp. weak) edges is NP-hard

Approximate minimum number of weak edges [Sintos and Tsaparas, 2014]

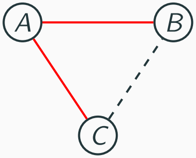
Strong Triadic Closure (Recap)

Strong Triadic Closure (STC)

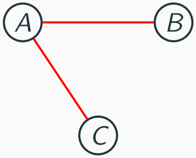
- If for three nodes, A, B, and C, there are strong ties between A and B, as well as A and C, there has to be a (weak or strong) tie between B and C



(a) Fulfills STC.



(b) Fulfills STC.



(c) Does not fulfill STC.

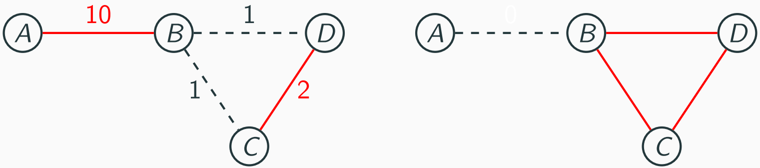
(Optimization problem) Finding edge labelling that fulfils STC with max (resp. min) number of strong (resp. weak) edges is NP-hard

Approximate minimum number of weak edges [Sintos and Tsaparas, 2014]

Weighted Strong Triadic Closure

We consider a **weighted version of the STC:**

- Given an edge-weighted (static) graph G
- Find edge labeling that fulfills the STC with the minimal sum of weak edge weights



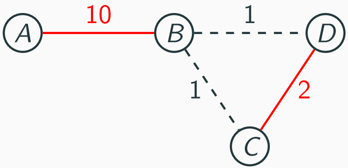
(a) Optimal Weighted STC **(b)** Optimal Non-weighted STC

Motivation: Use empirical knowledge if it is available, e.g., contact frequencies.

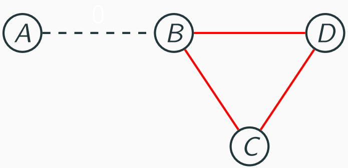
Weights are important: Even though non-weighted solution has more strong edges, weighted version agrees more with empirical knowledge and intuition.

We consider a **weighted version of the STC:**

- Given an edge-weighted (static) graph G
- Find edge labeling that fulfills the STC with the minimal sum of weak edge weights



(a) Optimal Weighted STC



(b) Optimal Non-weighted STC

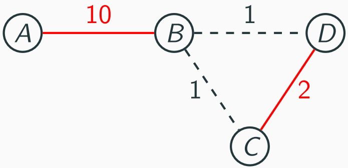
Motivation: Use empirical knowledge if it is available, e.g., contact frequencies.

Weights are important: Even though non-weighted solution has more strong edges, weighted version agrees more with empirical knowledge and intuition.

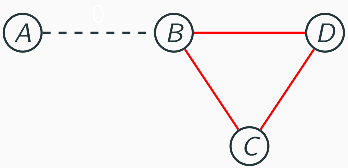
Weighted Strong Triadic Closure

We consider a **weighted version of the STC:**

- Given an edge-weighted (static) graph G
- Find edge labeling that fulfills the STC with the minimal sum of weak edge weights



(a) Optimal Weighted STC



(b) Optimal Non-weighted STC

Motivation: Use empirical knowledge if it is available, e.g., contact frequencies.

Weights are important: Even though non-weighted solution has more strong edges, weighted version agrees more with empirical knowledge and intuition.

Weighted Strong Triadic Closure

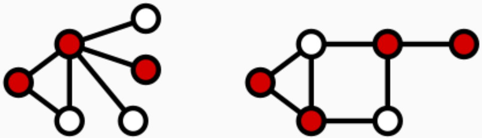
- Given an edge-weighted (static) graph, find a strong-weak labeling that fulfills the STC and minimizes the weight of the weak edges is NP-hard
- Can be solved exactly via Integer Linear Programming
- Impractical, especially for large-scale networks

Our Contributions

- We show how to use temporal information to infer the edge strengths of the underlying static graph. In particular, we generalize STC for weighted graphs and apply weighted STC for determining tie strength in temporal networks.
- We generalize the STC+ variant to weighted graphs that allows addition of new weak edges (to obtain improved solutions).
- We provide a streaming algorithm framework to efficiently approximate the weighted STC and STC+ over time with an approximation factor of 2 and 3, respectively.
- We propose an efficient dynamic k-approximation for the minimum weighted vertex cover problem (MWVC) in k-uniform hypergraphs, a key ingredient of our streaming framework.
- Our experiments with real-world temporal networks show that the weighted STC and STC+ lead to strong edges with higher weights, consistent with the given empirical edge weights. Furthermore, the experiments show that our streaming algorithm is orders of magnitude faster than the baseline, while keeping the same solution quality.

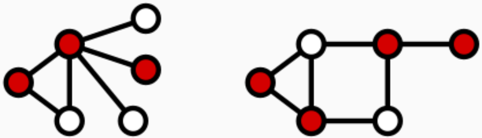
Minimum Vertex Cover

- In graph theory, a vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge of the graph.
- (Trivial vertex cover: take all the vertices.)
- (Optimization problem) Minimum vertex cover: Find a vertex cover of smallest possible size



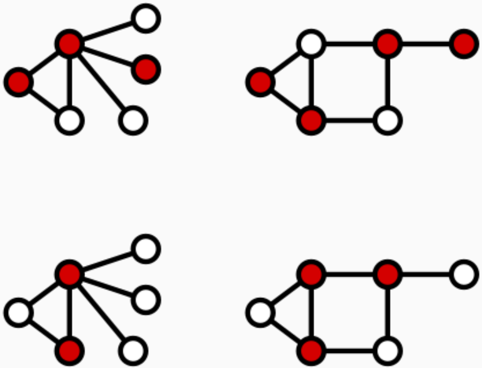
Minimum Vertex Cover

- In graph theory, a vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge of the graph.
- (Trivial vertex cover: take all the vertices.)
- (Optimization problem) Minimum vertex cover: Find a vertex cover of smallest possible size



Minimum Vertex Cover

- In graph theory, a vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge of the graph.
- (Trivial vertex cover: take all the vertices.)
- (Optimization problem) Minimum vertex cover: Find a vertex cover of smallest possible size

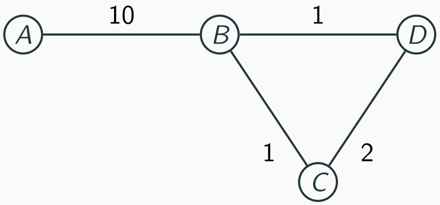


Approximation of the STC

Find a strong-weak labeling that fulfills the STC and minimizes the weight of the weak edges

Approximation:

1. Construct vertex weighted wedge graph $W(G)$
2. Approximate minimum weight vertex cover problem



Wedge: Pair of edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$ (cannot be both strong)

STC: No pair of strong edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$

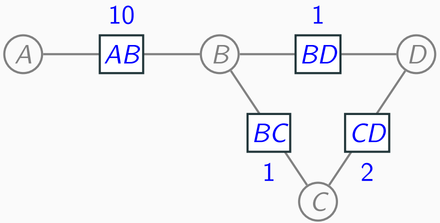
Vertex cover: for each wedge one edge weak $\rightarrow$ STC holds

Approximation of the STC

Find a strong-weak labeling that fulfills the STC and minimizes the weight of the weak edges

Approximation:

1. Construct vertex weighted wedge graph $W(G)$
2. Approximate minimum weight vertex cover problem



Wedge: Pair of edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$ (cannot be both strong)

STC: No pair of strong edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$

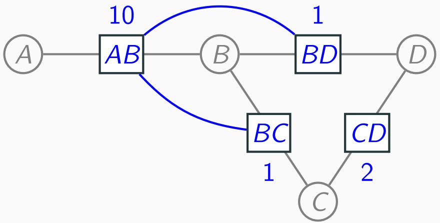
Vertex cover: for each wedge one edge weak $\rightarrow$ STC holds

Approximation of the STC

Find a strong-weak labeling that fulfills the STC and minimizes the weight of the weak edges

Approximation:

1. Construct vertex weighted wedge graph $W(G)$
2. Approximate minimum weight vertex cover problem



Wedge: Pair of edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$ (cannot be both strong)

STC: No pair of strong edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$

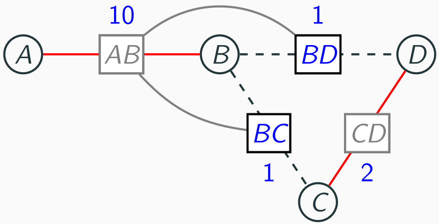
Vertex cover: for each wedge one edge weak $\rightarrow$ STC holds

Approximation of the STC

Find a strong-weak labeling that fulfills the STC and minimizes the weight of the weak edges

Approximation:

1. Construct vertex weighted wedge graph $W(G)$
2. Approximate minimum weight vertex cover problem



Wedge: Pair of edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$ (cannot be both strong)

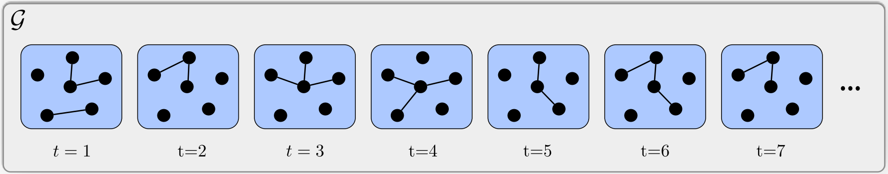
STC: No pair of strong edges $\{u, v\}$ and $\{v, w\}$ such that $\{u, w\} \notin E$

Vertex cover: for each wedge one edge weak $\rightarrow$ STC holds

Temporal Networks and Aggregated Graph

Temporal network:

- Set of (static) vertices V and a set of temporal edges $\mathcal{E}$
- Each temporal edge $(\{u, v\}, t) \in \mathcal{E}$ exists only at a discrete availability time



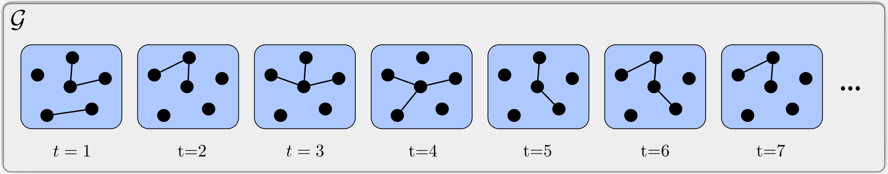
Weighted aggregated graph

- Graph $\mathcal{A}(\mathcal{G}) = (V, E, w)$ with $E = \{\{u, v\} \mid (\{u, v\}, t) \in \mathcal{E}\}$
- Edge weighting function w depends on the temporal edges
  - Contact frequency or duration of contacts

Temporal Networks and Aggregated Graph

Temporal network:

- Set of (static) vertices V and a set of temporal edges $\mathcal{E}$
- Each temporal edge $(\{u, v\}, t) \in \mathcal{E}$ exists only at a discrete availability time



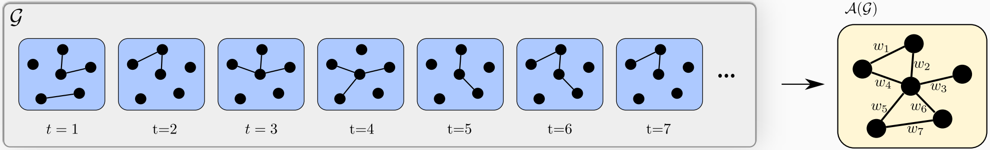
Weighted aggregated graph

- Graph $\mathcal{A}(\mathcal{G}) = (V, E, w)$ with $E = \{\{u, v\} \mid (\{u, v\}, t) \in \mathcal{E}\}$
- Edge weighting function w depends on the temporal edges
  - Contact frequency or duration of contacts

Temporal Networks and Aggregated Graph

Weighted aggregated graph

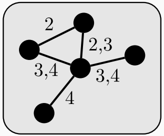
- Graph $\mathcal{A}(\mathcal{G}) = (V, E, w)$ with $E = \{\{u, v\} \mid (\{u, v\}, t) \in \mathcal{E}\}$
- Edge weighting function w depends on the temporal edges
  - Contact frequency or duration of contacts



Approximation of tie strength in temporal networks

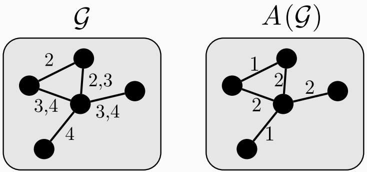
1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem

$\mathcal{G}$



Approximation of tie strength in temporal networks

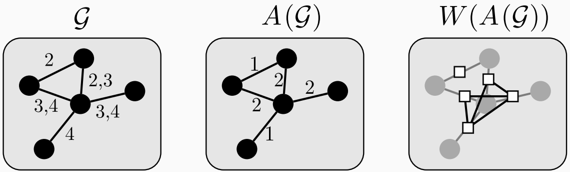
1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem



Approach for Temporal Networks

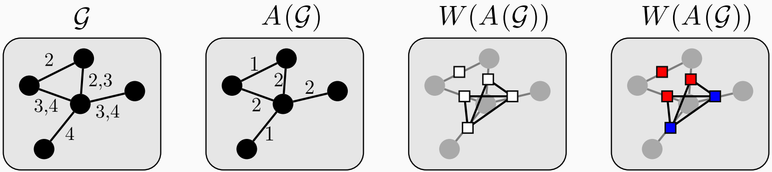
Approximation of tie strength in temporal networks

1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem



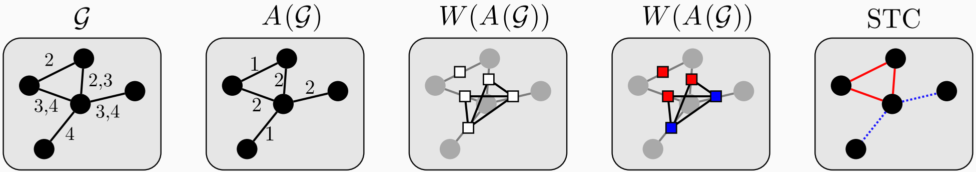
Approximation of tie strength in temporal networks

1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem



Approximation of tie strength in temporal networks

1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem



Approach for Temporal Networks

Approximation of tie strength in temporal networks

1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem

Problems:

- Size of wedge graph (number of wedges) is $\mathcal{O}(|V|^3)$ [Pyatkin et al., 2019]
- Temporal graph possibly infinite long stream of incoming edges
- Interested only in last day, week, month etc., or the change over time

Approach for Temporal Networks

Approximation of tie strength in temporal networks

1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem

Problems:

- Size of wedge graph (number of wedges) is $\mathcal{O}(|V|^3)$ [Pyatkin et al., 2019]
- Temporal graph possibly infinite long stream of incoming edges
- Interested only in last day, week, month etc., or the change over time

Approach for Temporal Networks

Approximation of tie strength in temporal networks

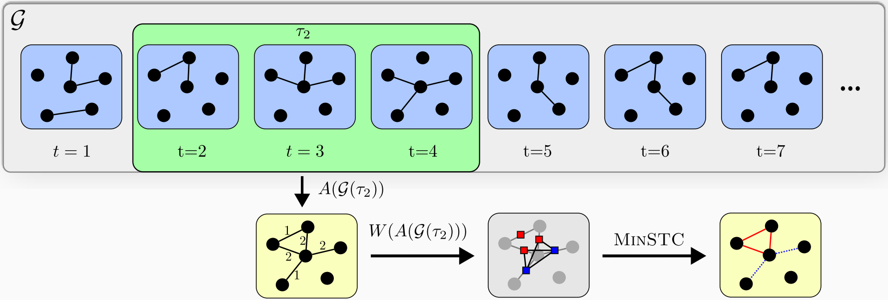
1. Compute edge-weighted aggregated graph $A(\mathcal{G})$
2. Construct vertex-weighted wedge graph $W(A(\mathcal{G}))$
3. Approximate minimum weighted vertex cover problem

Problems:

- Size of wedge graph (number of wedges) is $\mathcal{O}(|V|^3)$ [Pyatkin et al., 2019]
- Temporal graph possibly infinite long stream of incoming edges
- Interested only in last day, week, month etc., or the change over time

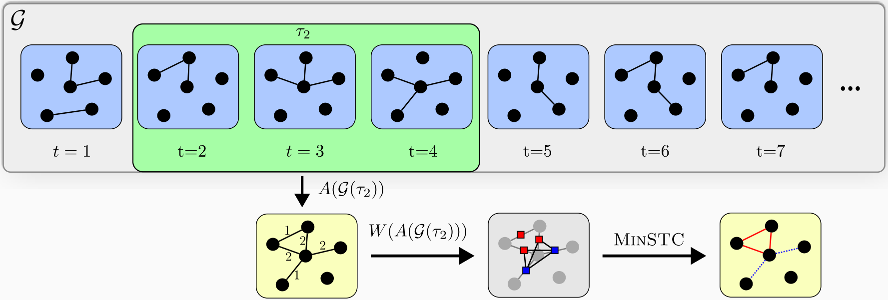
Streaming Algorithm

- **Sliding time window** to approximate changing STC in each time windows
  1. Smaller graphs: usually not all nodes have contact in the same time window.
  2. Can capture tie strength changes over time – also in possibly infinite edge streams



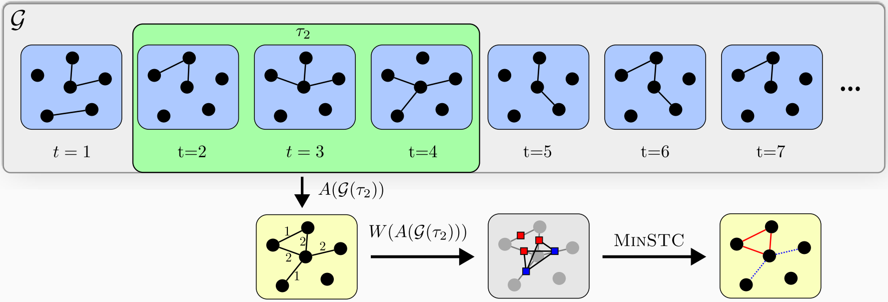
Streaming Algorithm

- **Sliding time window** to approximate changing STC in each time windows
  1. Smaller graphs: usually not all nodes have contact in the same time window.
  2. Can capture tie strength changes over time – also in possibly infinite edge streams

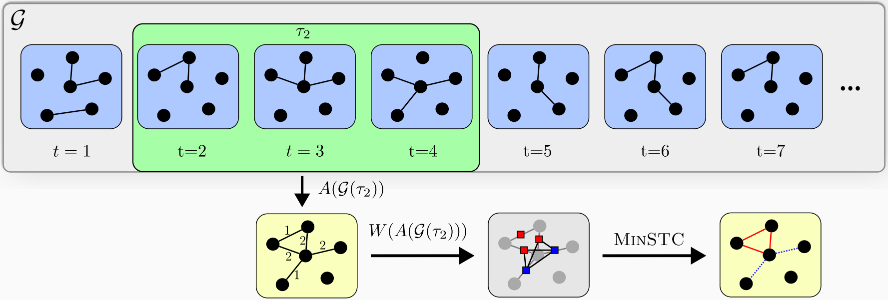


Streaming Algorithm

- **Sliding time window** to approximate changing STC in each time windows
  1. Smaller graphs: usually not all nodes have contact in the same time window.
  2. Can capture tie strength changes over time – also in possibly infinite edge streams

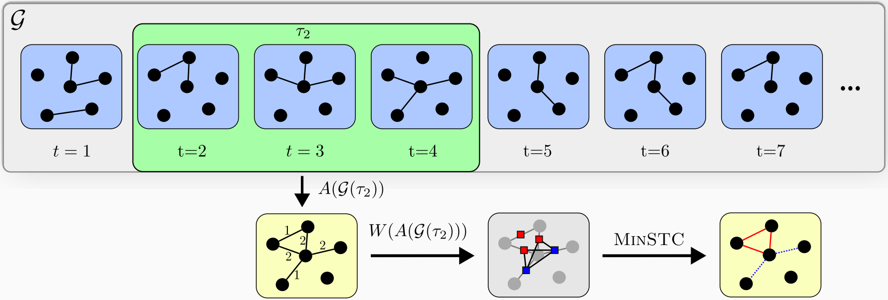


Streaming Algorithm



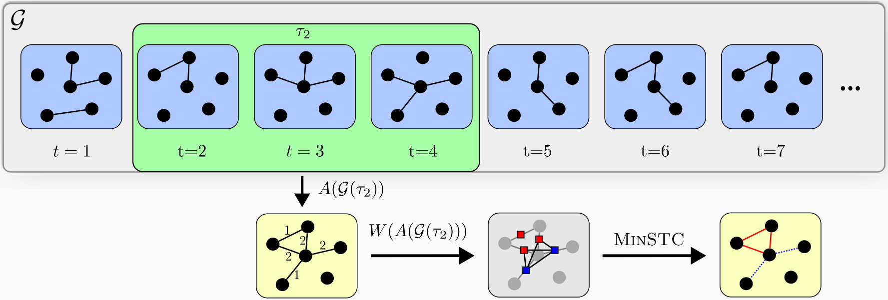
- **Move time window forward:**
  - **Step 1:** Update A_t to A_{t+1}
  - **Step 2:** Update W_t to W_{t+1} according to the changes in A
  - **Step 3:** Dynamically update vertex cover C in W_{t+1} s.t. $w(C) \leq 2w(OPT)$

Streaming Algorithm



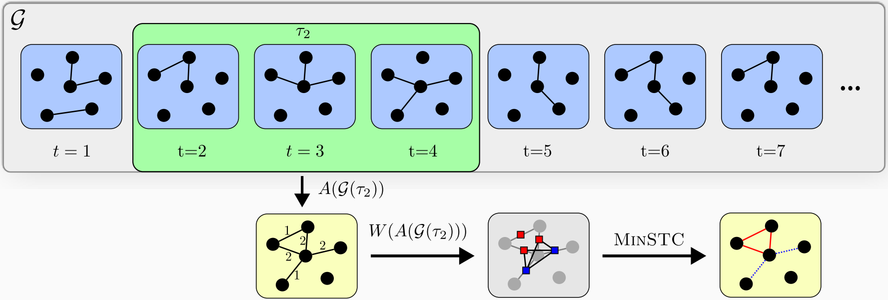
- **Move time window forward:**
  - **Step 1:** Update A_t to A_{t+1}
  - **Step 2:** Update W_t to W_{t+1} according to the changes in A
  - **Step 3:** Dynamically update vertex cover C in W_{t+1} s.t. $w(C) \leq 2w(OPT)$

Streaming Algorithm



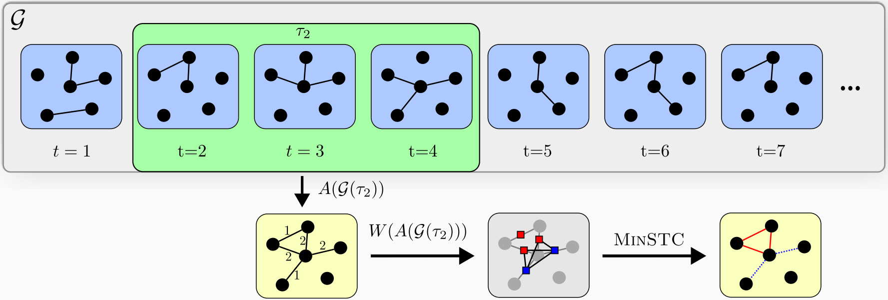
- **Move time window forward:**
  - **Step 1:** Update A_t to A_{t+1}
  - **Step 2:** Update W_t to W_{t+1} according to the changes in A
  - **Step 3:** Dynamically update vertex cover C in W_{t+1} s.t. $w(C) \leq 2w(OPT)$

Streaming Algorithm



Move time window forward:

- Algorithm maintains a vertex cover with $w(C) \leq 2w(OPT)$
- Time complexity $\mathcal{O}(\xi \cdot d_A \cdot d_W^2)$
  - ξ is the maximum of number of effective edge insertions or deletions in A in iteration t
  - d_W (d_A) is the maximal degree in W (A, resp.) after iteration t of the streaming algorithm



Move time window forward:

- Algorithm maintains a vertex cover with $w(C) \leq 2w(OPT)$
- Time complexity $\mathcal{O}(\xi \cdot d_A \cdot d_W^2)$
  - ξ is the maximum of number of effective edge insertions or deletions in A in iteration t
  - d_W (d_A) is the maximal degree in W (A, resp.) after iteration t of the streaming algorithm

Experiments

Compared the weighted and unweighted STC and STC+ on real-world temporal networks and evaluated the efficiency of our streaming algorithm.

More specifically, we discuss the following questions:

(Q1) How do the weighted and non-weighted versions of the STC and STC+ compare to each other?

(Q2) How fast is our streaming algorithm?

Experiments

Compared the weighted and unweighted STC and STC+ on real-world temporal networks and evaluated the efficiency of our streaming algorithm.

More specifically, we discuss the following questions:

(Q1) How do the weighted and non-weighted versions of the STC and STC+ compare to each other?

(Q2) How fast is our streaming algorithm?

Algorithms (weighted STC)

Implemented the following algorithms for computing the weighted STC:

1. `ExactW` and `ExactW+` are the weighted exact computation using the ILPs for the weighted STC and STC+.
2. `Pricing` and `Pricing+` use the non-dynamic pricing approximation in the wedge graph for the weighted STC and STC+.
3. `DynAppr` is our dynamic streaming algorithm (dynamization of pricing method).
4. `STCtime` is a baseline streaming algorithm that recomputes the minimum weight vertex cover (MWVC) with the pricing method for each time window.

1. and 2. are for quality.
3. and 4. are for performance

Algorithms (non-weighted)

Implemented the following algorithms for computing the non-weighted STC:

- `ExactNw` and `ExactNw+` are the exact computations using ILP (see [Adriaens et al., 2020]).
- `Matching` is the matching-based approximation of the unweighted vertex cover in the (non-weighted) wedge graph (see [Sintos and Tsaparas, 2014]).
- `Matching+` is the adapted matching-based approximation of the unweighted vertex cover for the non-weighted wedge hypergraph.
- `HighDeg` is a $\mathcal{O}(\log n)$ approximation by iteratively adding the highest degree vertex to the vertex cover, and removing all incident edges (see [Sintos and Tsaparas, 2014]).

Data Sets

Several real-world temporal networks from different domains.

1. Human contact networks from the SocioPatterns project at `www.sociopatterns.org/`. For these networks, the edges represent human contacts that are recorded using proximity sensors in 20-second intervals.
2. Online communication and social networks

Human Contact Networks

- Malawi is a contact network of individuals living in a village in rural Malawi [Ozella et al., 2021]. The network spans around 13 days.
- Copresence is a contact network representing spatial copresence in a workplace over 11 days [Génois and Barrat, 2018].
- Primary is a contact network among primary school students over two days [Stehlé et al., 2011].

Online Communication and Social Networks

- Enron is an email network between employees of a company spanning over 3.6 years [Klimt and Yang, 2004]. Data set available at `www.networkrepository.com/`.
- Yahoo is a communication network available at the Network Repository [Rossi and Ahmed, 2015] (`www.networkrepository.com/`). The network spans around 28 days.
- StackOverflow is based on the stack exchange website StackOverflow [Paranjape et al., 2017]. Edges represent answers to comments and questions. The network spans around 7.6 years. Available at `snap.stanford.edu/data/index.html`.
- Reddit is based on the Reddit social network [Hessel et al., 2016, Liu et al., 2019]. A temporal edge $(\{u, v\}, t)$ means that a user u commented on a post or comment of user v at time t. The network spans over 10.05 years. We used a subgraph from the data set provided at `www.cs.cornell.edu/~arb/data/temporal-reddit-reply/`.

Data Sets Statistics

Data set	Properties					
	$\|V\|$	$\|\mathcal{E}\|$	$\|\mathcal{T}(\mathcal{G})\|$	$\|V(W)\|$	$\|E(W)\|$	#Triangles
Malawi	86	102 293	43 438	347	2 254	441
Copresence	219	1 283 194	21 536	16 725	549 449	713 002
Primary	242	125 773	3 100	8 317	337 504	103 760
Enron	87 101	1 147 126	220 312	298 607	45 595 540	1 234 257
Yahoo	100 001	3 179 718	1 498 868	594 989	18 136 435	590 396
StackOverflow	2 601 977	63 497 050	41 484 769	28 183 518	*33 898 217 240	*110 670 755
Reddit	5 279 069	116 029 037	43 067 563	96 659 109	*86 758 743 921	*901 446 625

(*Estimated)

($|\mathcal{T}(\mathcal{G})|$ is the number of different timestamps in the networks)

Implementations and Hw/Sw Platform

- All algorithms implemented in C++, using GNU CC Compiler 9.3.0 with the flag `--O2` and Gurobi 9.5.0 with Python 3 for solving ILPs.
- All experiments ran on a workstation with an AMD EPYC 7402P 24-Core Processor with 3.35 GHz and 256 GB of RAM running Ubuntu 18.04.3 LTS, and with a time limit of twelve hours.
- Source code available at `gitlab.com/tgpublic/tgstc`.

(Q1) Weighted vs. Unweighted STC

Data set	Weighted		Non-weighted		
	`ExactW`	`Pricing`	`ExactNw`	`Matching`	`HighDeg`
Malawi	30.83	29.97	37.75	27.38	36.31
Copresence	31.12	21.37	37.95	29.20	35.31
Primary	27.17	21.94	27.83	18.99	27.35
Enron	OOT	2.75	OOT	3.28	4.61
Yahoo	OOT	9.86	OOT	9.98	14.29

Percentage of strong edges in aggregated graph (OOT—out of time)

- *StackOverflow* and *Reddit* too large
- Contact frequency as weighting function
- For exact algs: # strong edges higher for non-weighted (expected, weights > 1)

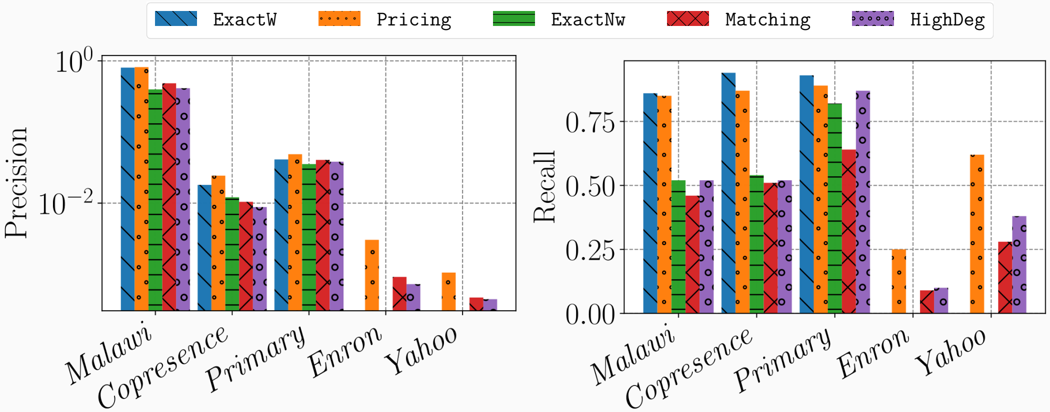
(Q1) Weighted vs. Unweighted STC

Data set	Weighted				Non-weighted					
	ExactW		Pricing		ExactNw		Matching		HighDeg	
	Weak	Strong	Weak	Strong	Weak	Strong	Weak	Strong	Weak	Strong
Malawi	23.87	902.46	24.40	926.58	218.08	421.27	255.33	399.48	242.84	385.92
Copresence	20.30	78.32	46.13	189.31	27.22	56.56	58.85	120.07	57.13	112.63
Primary	2.73	20.48	6.58	45.50	3.34	18.49	9.32	39.88	6.19	38.84
Enron	OOT	OOT	3.69	9.33	OOT	OOT	3.77	6.01	3.76	5.50
Yahoo	OOT	OOT	4.37	14.23	OOT	OOT	4.78	10.42	4.60	9.84

Mean edge weights (OOT—out of time)

- Better STC labeling: strong edges with high weights / weak edges with low weights
- Exact methods: Mean weight of strong edges higher for `ExactW` than `ExactNw`
- `Pricing` highest mean edge weight for strong edges
- Suggests effectiveness of our approach: Empirical *a priori* knowledge given by edge weights seems successfully captured by weighted STC

(Q1) Weighted vs. Unweighted STC



H: set of top-100 highest weight edges, S: set of strong edges

- **Precision**: $p = |H \cap S| / |S|$
- **Recall**: $r = |H \cap S| / |H|$

(Q1) Weighted vs. Unweighted STC+

Data set	Weighted		Non-weighted	
	`ExactW+`	`Pricing+`	`ExactNw+`	`Matching+`
Malawi	31.70	31.12	50.72	34.29
Copresence	83.04	38.00	90.73	57.27
Primary	37.39	26.46	OOT	32.25
Enron	OOT	3.66	OOT	5.57
Yahoo	OOT	12.35	OOT	14.03

Percentage of strong edges in aggregated graph (newly inserted edges excluded, OOT—out of time)

- Weighting parameter $\alpha = 0.5$ for newly inserted edges in the weighted version
- More strong edges compared to standard STC, increase strongest for *Copresence* (by inserting additional weak edges, number of strong edges can be increased)
- Unweighted `Matching+` approximation more strong edges than weighted `Pricing+` (`Pricing+` tries to minimize weight of weak edges, `Matching+` number of weak edges)

(Q1) Weighted vs. Unweighted STC+

Data set	Weighted				Non-weighted			
	`ExactW+`		`Pricing+`		`ExactNw+`		`Matching+`	
	Weak	Strong	Weak	Strong	Weak	Strong	Weak	Strong
Malawi	21.33	883.97	18.61	905.97	242.02	343.19	198.56	479.16
Copresence	27.93	86.69	31.17	151.02	38.75	80.60	46.68	99.13
Primary	4.83	32.36	5.35	42.24	OOT	OOT	8.52	28.98
Enron	OOT	OOT	3.63	9.31	OOT	OOT	3.77	5.11
Yahoo	OOT	OOT	4.15	13.68	OOT	OOT	4.53	10.21

Mean edge weights (OOT—out of time)

- Compared to STC, mean weights of strong edges lower (more strong edges in solution)
- Except for *Copresence*: mean weights of strong edges higher for exact solutions
- Similarly to STC, weighted STC+ (`ExactW+`, `Pricing+`) leads to higher quality solutions (higher mean weights for strong edges and lower mean weights for weak edges)

(Q2) Efficiency of Streaming Algorithm

Data set	$\Delta = 1$ hour		$\Delta = 1$ day		$\Delta = 1$ week	
	DynAppr	STCtime	DynAppr	STCtime	DynAppr	STCtime
Enron	264.74	**89.18**	**306.13**	1 606.09	**352.01**	20 870.77
Yahoo	**15.99**	767.40	**91.46**	OOT	**144.52**	OOT
StackOverflow	**170.38**	2 298.58	**971.22**	OOT	**16 461.53**	OOT
Reddit	**1 254.66**	13 244.84	**37 627.79**	OOT	OOT	OOT

Running times (secs) with time window Δ (OOT—out of time after 12h)

- Our streaming algorithm `DynAppr` often orders of magnitudes faster than baseline `STCtime`
- Only exception: *Enron* and $\Delta = 1$ hour (on average, computed wedge graphs very small)
- However, even for *Enron* `DynAppr` scales well with Δ (`STCtime` does not, as size of wedge graph increases with Δ – more contacts happen in longer time windows)

Conclusions

- Generalized STC and STC+ to weighted versions to include *a priori* knowledge in the form of edge weights representing empirical tie strength
- Applied our new STC variants to temporal networks and showed that can yield meaningful results
- Our main contribution is a 2-approximation (resp. 3-approximation) streaming algorithm that can efficiently compute weighted STC (resp. STC+) in temporal networks over time
- Empirically validated its efficiency in with experimental evaluations
- Introduced a fully dynamic k-approximation of the MWVC problem in hypergraphs with k-uniform hyperedges that allows efficient updates in our streaming algorithm

Conclusions

- Generalized STC and STC+ to weighted versions to include *a priori* knowledge in the form of edge weights representing empirical tie strength
- Applied our new STC variants to temporal networks and showed that can yield meaningful results
- Our main contribution is a 2-approximation (resp. 3-approximation) streaming algorithm that can efficiently compute weighted STC (resp. STC+) in temporal networks over time
- Empirically validated its efficiency in with experimental evaluations
- Introduced a fully dynamic k-approximation of the MWVC problem in hypergraphs with k-uniform hyperedges that allows efficient updates in our streaming algorithm

Conclusions

- Generalized STC and STC+ to weighted versions to include *a priori* knowledge in the form of edge weights representing empirical tie strength
- Applied our new STC variants to temporal networks and showed that can yield meaningful results
- Our main contribution is a 2-approximation (resp. 3-approximation) streaming algorithm that can efficiently compute weighted STC (resp. STC+) in temporal networks over time
- Empirically validated its efficiency in with experimental evaluations
- Introduced a fully dynamic k-approximation of the MWVC problem in hypergraphs with k-uniform hyperedges that allows efficient updates in our streaming algorithm

Conclusions

- Generalized STC and STC+ to weighted versions to include *a priori* knowledge in the form of edge weights representing empirical tie strength
- Applied our new STC variants to temporal networks and showed that can yield meaningful results
- Our main contribution is a 2-approximation (resp. 3-approximation) streaming algorithm that can efficiently compute weighted STC (resp. STC+) in temporal networks over time
- Empirically validated its efficiency in with experimental evaluations
- Introduced a fully dynamic k-approximation of the MWVC problem in hypergraphs with k-uniform hyperedges that allows efficient updates in our streaming algorithm

Conclusions

- Generalized STC and STC+ to weighted versions to include *a priori* knowledge in the form of edge weights representing empirical tie strength
- Applied our new STC variants to temporal networks and showed that can yield meaningful results
- Our main contribution is a 2-approximation (resp. 3-approximation) streaming algorithm that can efficiently compute weighted STC (resp. STC+) in temporal networks over time
- Empirically validated its efficiency in with experimental evaluations
- Introduced a fully dynamic k-approximation of the MWVC problem in hypergraphs with k-uniform hyperedges that allows efficient updates in our streaming algorithm

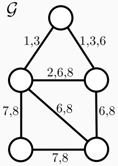
Future Work

- Extend streaming algorithm to further variants of STC and STC+
- E.g., variants with multiple relationship types, or allowing violations [Sintos and Tsaparas, 2014]

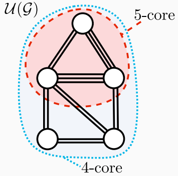
(Ongoing) Temporal Network Core Decomposition and Community Search

- We introduce new generalization of k-core decomposition for temporal networks.
- We try to respect more temporal dynamics.

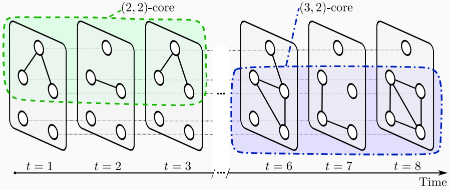
(Ongoing) Temporal Network Core Decomposition and Community Search



(a) Temporal graph $\mathcal{G}$.



(b) Underlying multigraph.



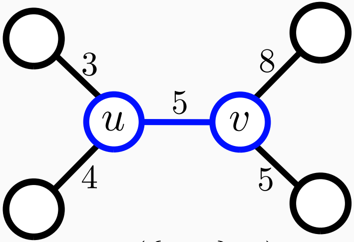
(c) $\mathcal{G}$ in time-slice representation with (k, Δ)-cores.

For $\Delta = 2$ two (k, Δ)-cores in the temporal graph. In contrast to the static k-cores shown in (b), our (k, Δ)-cores can identify core structures in time.

(Ongoing) Temporal Network Core Decomposition and Community Search

- In contrast to the standard definition and previous core-like decompositions for temporal graphs, our (k, Δ)-core decomposition is an edge-based decomposition founded on the new notion of Δ-degree.
- The Δ-degree of an edge is defined as the minimum number of edges incident to one of its endpoints that have a temporal distance of at most Δ.
- Moreover, we define a new notion of Δ-connectedness leading to an efficiently computable equivalence relation between connected components of the temporal network.

(Ongoing) Temporal Network Core Decomposition and Community Search



$$e = (\{u, v\}, 5)$$

Example for Δ-degree d_Δ: $d_1(e) = 2$, $d_2(e) = 2$, and $d_3(e) = 3$.

(Ongoing) Temporal Network Core Decomposition and Community Search

- We provide efficient algorithms for the (k, Δ)-core decomposition and Δ-connectedness,
- Apply them to solve community search problems, where we are given a query node and want to find a densely connected community containing the query node.
- Such a community is an edge-induced temporal subgraph representing cohesive and persistent groups of nodes that interact frequently over time taking the temporal locality and dynamics of interactions into consideration.

(Ongoing) Temporal Network Core Decomposition and Community Search

- In our experimental evaluations (Twitter dataset), we found that in a real-world social network, the inner (k, Δ)-cores contain only the spreading of misinformation and that the Δ-connected components of the cores are highly edge-homophilic, i.e., the majorities of the edges in the Δ-connected components represent either misinformation or fact-checking.
- Moreover, we demonstrate how our algorithms for Δ-community search successfully and efficiently identify informative structures in collaboration networks (such as DBLP).

Adriaens, F., De Bie, T., Gionis, A., Lijffijt, J., Matakos, A., and Rozenshtein, P. (2020).
Relaxing the strong triadic closure problem for edge strength inference.
Data Mining and Knowledge Discovery, pages 1–41.

Easley, D. A. and Kleinberg, J. M. (2010).
Networks, Crowds, and Markets - Reasoning About a Highly Connected World.
Cambridge University Press.

Génois, M. and Barrat, A. (2018).
Can co-location be used as a proxy for face-to-face contacts?
EPJ Data Science, 7(1):11.

Hessel, J., Tan, C., and Lee, L. (2016).
Science, askscience, and badscience: On the coexistence of highly related communities.
In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 10, pages 171–180.

Klimt, B. and Yang, Y. (2004).
The Enron corpus: A new dataset for email classification research.
In *European Conf. on Machine Learning*, pages 217–226. Springer.

Liu, P., Benson, A. R., and Charikar, M. (2019).
Sampling methods for counting temporal motifs.
In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 294–302.

Ozella, L., Paolotti, D., Lichand, G., Rodríguez, J. P., Haenni, S., Phuka, J., Leal-Neto, O. B., and Cattuto, C. (2021).
Using wearable proximity sensors to characterize social contact patterns in a village of rural malawi.
EPJ Data Science, 10(1):46.

Paranjape, A., Benson, A. R., and Leskovec, J. (2017).
Motifs in temporal networks.
In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 601–610.

Pyatkin, A., Lykhovyd, E., and Butenko, S. (2019).
The maximum number of induced open triangles in graphs of a given order.
Optimization Letters, 13(8):1927–1935.

Rossi, R. A. and Ahmed, N. K. (2015).
The network data repository with interactive graph analytics and visualization.
In *AAAI*.

Sintos, S. and Tsaparas, P. (2014).
Using strong triadic closure to characterize ties in social networks.
In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1466–1475.

Stehlé, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.-F., Quaggiotto, M., Van den Broeck, W., Régis, C., Lina, B., et al. (2011).
High-resolution measurements of face-to-face contact patterns in a primary school.
PloS one, 6(8):e23176.