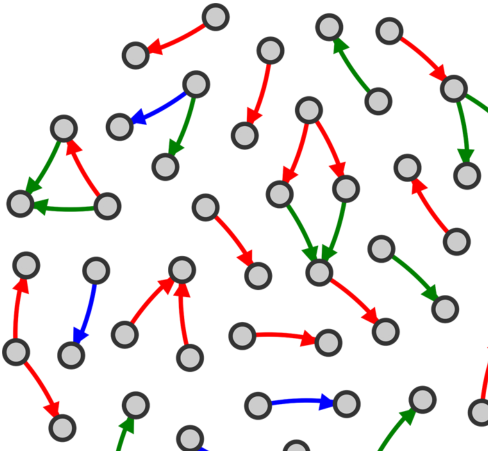
UNIVERSITY OF OXFORD

Mathematical Institute

COLLECTIVE BEHAVIOUR IN TEMPORAL NETWORKS

DOOCN Satellite, CCS18

27th September 2018



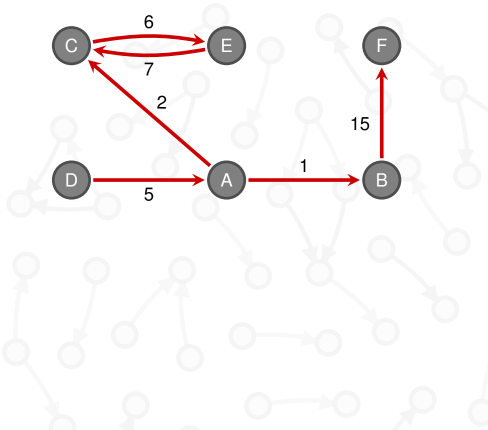
Andrew Mellor

Mathematical Institute

University of Oxford

A sequence of *temporal events*,

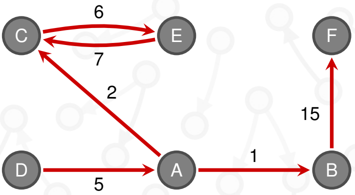
$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$



A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

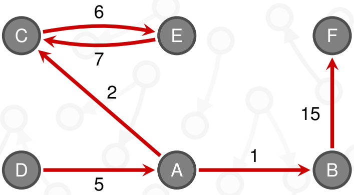


A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

1. Twitter/Social Networks

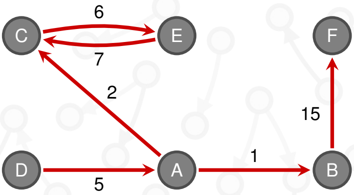


A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

1. Twitter/Social Networks
2. Telephone Calls

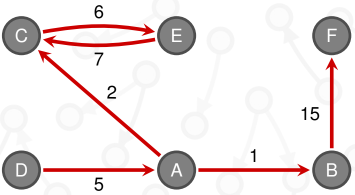


A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

1. Twitter/Social Networks
2. Telephone Calls
3. Proximity Networks

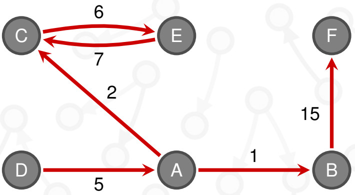


A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

1. Twitter/Social Networks
2. Telephone Calls
3. Proximity Networks
4. Email Correspondence

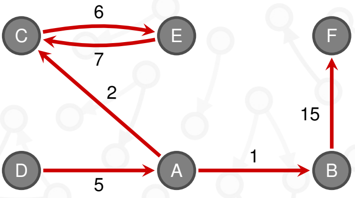


A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

1. Twitter/Social Networks
2. Telephone Calls
3. Proximity Networks
4. Email Correspondence



Other Representations:

A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

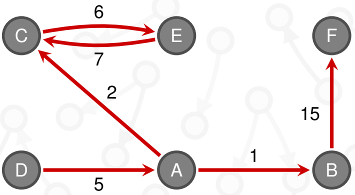
Examples:

1. Twitter/Social Networks
2. Telephone Calls
3. Proximity Networks
4. Email Correspondence

Other Representations:

1. Adjacency matrices

$$(A_k)_{k=1}^{T}$$



A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

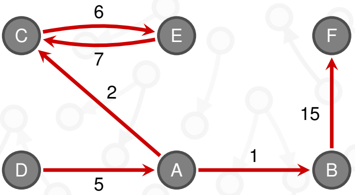
1. Twitter/Social Networks
2. Telephone Calls
3. Proximity Networks
4. Email Correspondence

Other Representations:

1. Adjacency matrices

$$(A_k)_{k=1}^{T}$$

2. Time-node graphs



A sequence of *temporal events*,

$$e_i = (\underbrace{u_i}_{\text{source}}, \underbrace{v_i}_{\text{target}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Examples:

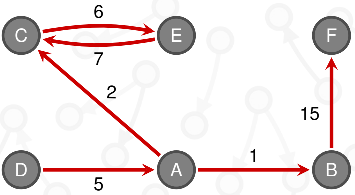
1. Twitter/Social Networks
2. Telephone Calls
3. Proximity Networks
4. Email Correspondence

Other Representations:

1. Adjacency matrices

$$(A_k)_{k=1}^{T}$$

2. Time-node graphs
3. Adjacency tensors

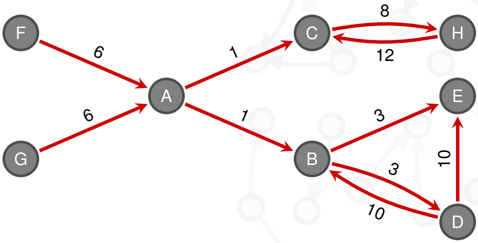


Not all interactions are pairwise, or dyadic.

In these cases we can consider *temporal hyper-events*,

$$e_i = (\underbrace{U_i}_{\text{sources}}, \underbrace{V_i}_{\text{targets}}, \underbrace{t_i}_{\text{time}}, \underbrace{\delta_i}_{\text{duration}})$$

Here a set of sources can interact with a set of targets (undirected hyper-events can also be defined).



EVENT GRAPH

Consider a temporal network $G_T = (V, T, E)$ where $E \subseteq V^2 \times T$ is the set of temporal events.

Consider a temporal network $G_T = (V, T, E)$ where $E \subseteq V^2 \times T$ is the set of temporal events.

Event Graph:

An *event graph* G is a directed static graph given by the tuple $G = (E, f_E)$ where E is a set of temporal events, and $f_E : E \times E \to [0, 1]$ is a binary function which prescribes the edges of the graph.

If f_E has no explicit dependence on the set of events then it is denoted f.

Consider a temporal network $G_T = (V, T, E)$ where $E \subseteq V^2 \times T$ is the set of temporal events.

Event Graph:

An *event graph* G is a directed static graph given by the tuple $G = (E, f_E)$ where E is a set of temporal events, and $f_E : E \times E \to [0, 1]$ is a binary function which prescribes the edges of the graph.

If f_E has no explicit dependence on the set of events then it is denoted f.

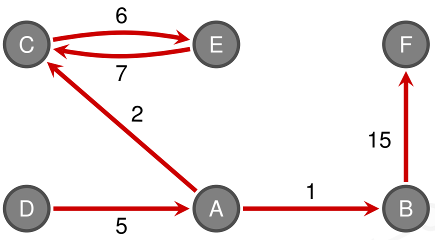
Weighted Event Graph:

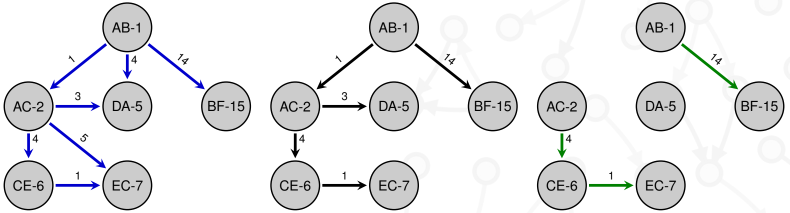
The weighted event graph is topologically equivalent to the event graph only that edges are weighted by the time between the two events.

This amounts to using a weighted joining function

$$f^\tau(e_i, e_j) = \tau_{ij} f(e_i, e_j),$$

where τ_{ij} is the inter-event time.

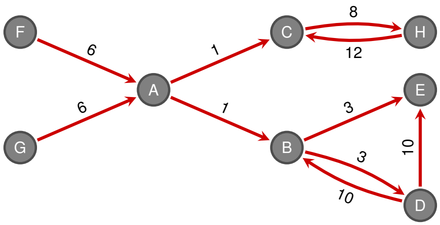




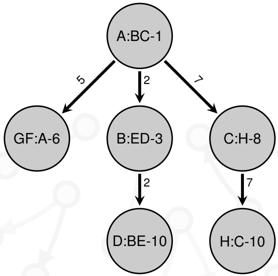
(a) Δt-adjacency

(b) Node-subsequent Δt-adjacency

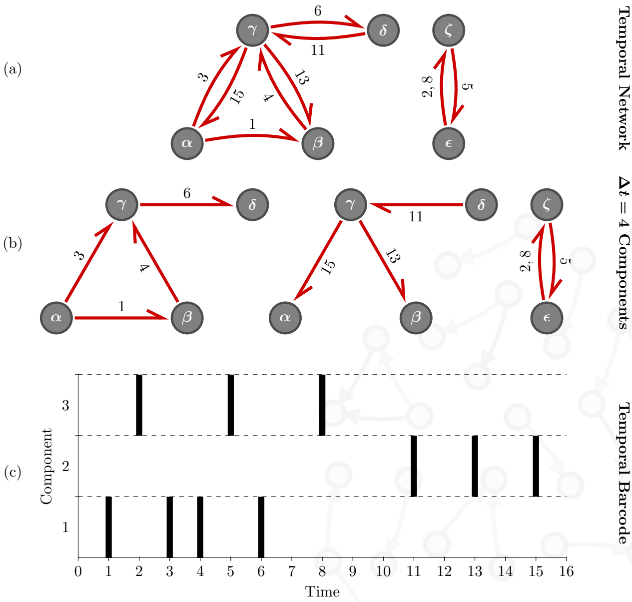
(c) Walk-forming Δt-adjacency



(a) Temporal Hypergraph



(b) Node-subsequent Δt-adjacent event graph

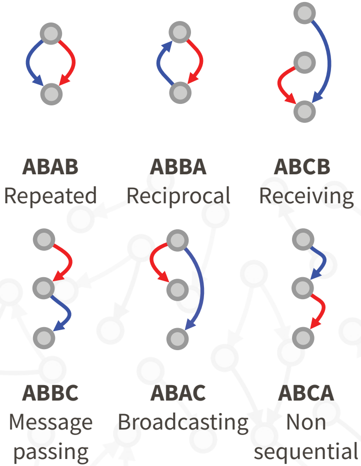


(a)
Temporal Network
6
11
3
15
4
13
1
2, 8
5
(b)
Δt = 4 Components
6
3
4
1
11
15
13
2, 8
5
(c)
Temporal Barcode
Component
Time

Temporal motifs are repeatable patterns observed in the network.

Two adjacent events can have one of 6 node patterns, or motifs.

Each motif is associated with a particular behaviour.



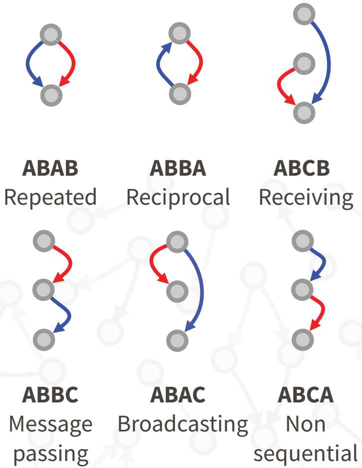
Red events occur before blue events.

Temporal motifs are repeatable patterns observed in the network.

Two adjacent events can have one of 6 node patterns, or motifs.

Each motif is associated with a particular behaviour.

- We can incorporate coloured motifs (e.g. distinguish between tweets/retweets or phonecalls/SMS).



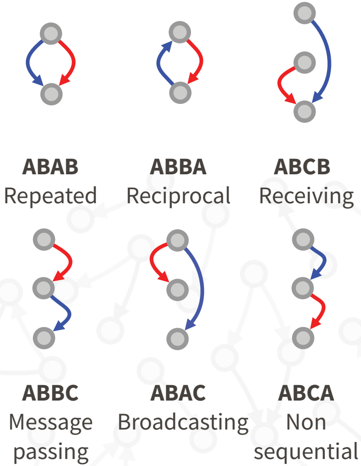
Red events occur before blue events.

Temporal motifs are repeatable patterns observed in the network.

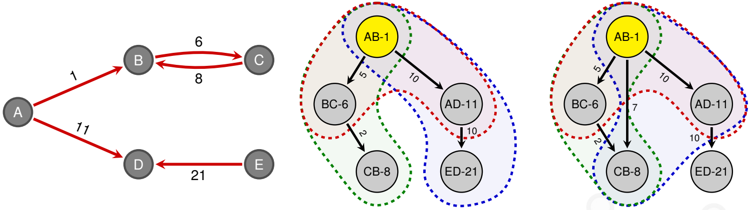
Two adjacent events can have one of 6 node patterns, or motifs.

Each motif is associated with a particular behaviour.

- We can incorporate coloured motifs (e.g. distinguish between tweets/retweets or phonecalls/SMS).
- Higher-order motifs are possible (3, 4 events).



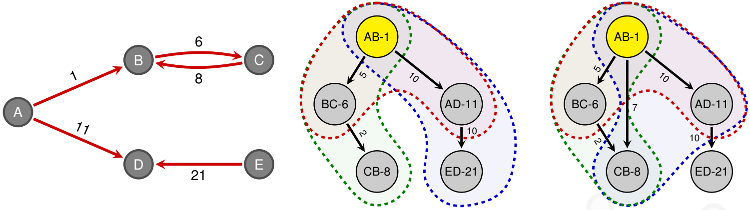
Red events occur before blue events.



(a) **(b)** Sequential **(c)** Windowed

Figure: Schematics for the calculation of temporal motifs using an event graph.

Sequential	Temporal motifs in time-dependent networks. [*Kovanen et. al.* (2011)]
Windowed	Motifs in temporal networks. [*Paranjape et. al.* (2017)]

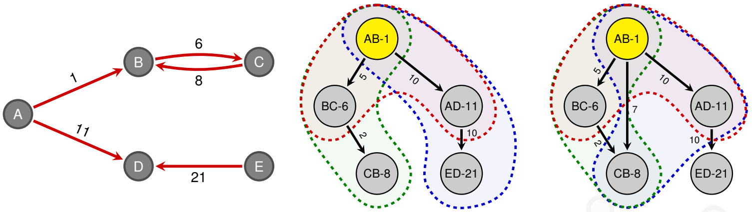


(a) **(b)** Sequential **(c)** Windowed

Figure: Schematics for the calculation of temporal motifs using an event graph.

Sequential	Temporal motifs in time-dependent networks. [*Kovanen et. al.* (2011)]
Windowed	Motifs in temporal networks. [*Paranjape et. al.* (2017)]

Temporal motifs are closely related to the event graph formalism.



(a) (b) Sequential (c) Windowed

Figure: Schematics for the calculation of temporal motifs using an event graph.

Sequential	Temporal motifs in time-dependent networks. [*Kovanen et. al.* (2011)]
Windowed	Motifs in temporal networks. [*Paranjape et. al.* (2017)]

Temporal motifs are closely related to the event graph formalism.

There are at least two definitions of temporal motifs, both can be expressed as subgraphs of the event graph.

Encode temporal components based on event graph features and aggregate graph features (to capture higher-order effects).

Event Graph Features:

1. Temporal motif distribution
2. Temporal motif entropy
3. Inter-event time entropy
4. Activity

Aggregate Graph Features:

1. Clustering
2. Reciprocity
3. Degree assortativity*

Encode temporal components based on event graph features and aggregate graph features (to capture higher-order effects).

Event Graph Features:

1. Temporal motif distribution
2. Temporal motif entropy
3. Inter-event time entropy
4. Activity

Aggregate Graph Features:

1. Clustering
2. Reciprocity
3. Degree assortativity*

Normalise feature vectors and perform a hierarchical clustering.

Data collected from Twitter (sampled by keyword using the Twitter Streaming API).

Tweets which contain the word *Emirates* (and common variants).

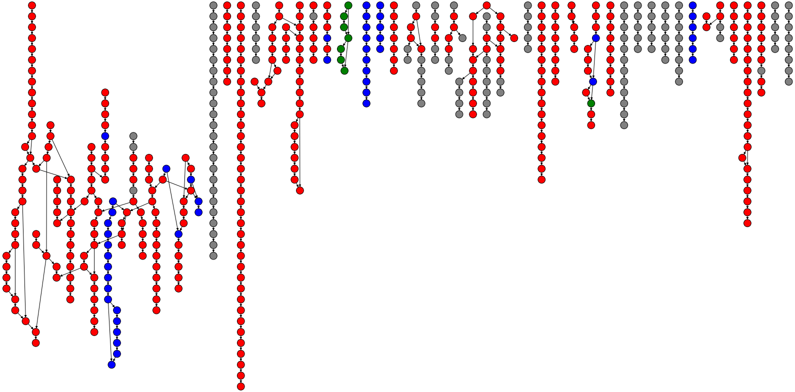
Subsequently collected all the accounts present in the sample and collect all tweets they have participated in during the course of the entire day using the REST API.

We sampled our keyword using the above procedure over one day:

# Tweets	# Events	# Users	% Retweets
161,730	130,360	48,249	52.95

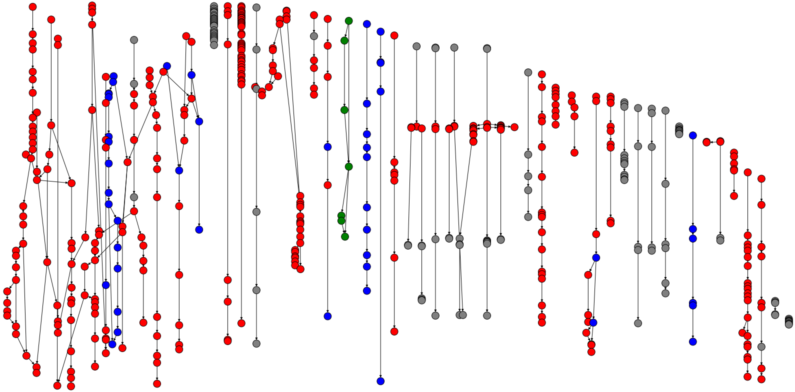
We set the link threshold $\Delta t = 240$s (or four minutes). The 4 minute event graph has 2300 components with at least ten events.

EXAMPLE - EVENT GRAPH

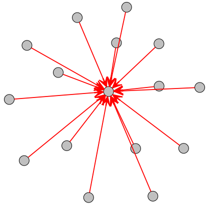


Green (message) / Red (retweet) / Blue (reply) / Grey (status)

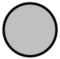
EXAMPLE - EVENT GRAPH



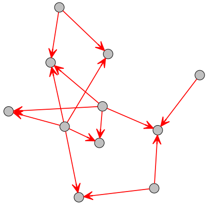
Green (message) / Red (retweet) / Blue (reply) / Grey (status)



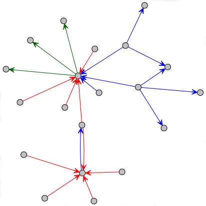
(a) Cluster 1



(b) Cluster 2



(c) Cluster 3



(d) Cluster 4

Aggregate (static) graphs of representative examples from each cluster.

Now take the opposite viewpoint, imagine we have an underlying network (unknown), but we observe a process evolving on the network.

Now take the opposite viewpoint, imagine we have an underlying network (unknown), but we observe a process evolving on the network.

How can we...

Now take the opposite viewpoint, imagine we have an underlying network (unknown), but we observe a process evolving on the network.

How can we...

- Understand the process dynamics?

Now take the opposite viewpoint, imagine we have an underlying network (unknown), but we observe a process evolving on the network.

How can we...

- Understand the process dynamics?
- Uncover the underlying network?

Now take the opposite viewpoint, imagine we have an underlying network (unknown), but we observe a process evolving on the network.

How can we...

- Understand the process dynamics?
- Uncover the underlying network?

One way is through correlations of node state timeseries, but can we instead use edge information?

Consider two processes on an Erdos-Renyi network:

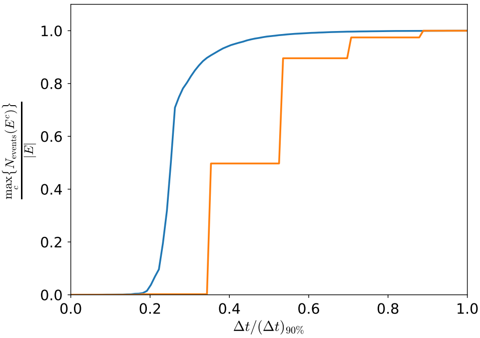
Consider two processes on an Erdos-Renyi network:

1. Random Edge Sampling: Edges are picked randomly at each timestep. Time incremented by a value draw from an exponential r.v. with mean 1.

Consider two processes on an Erdos-Renyi network:

1. Random Edge Sampling: Edges are picked randomly at each timestep. Time incremented by a value draw from an exponential r.v. with mean 1.

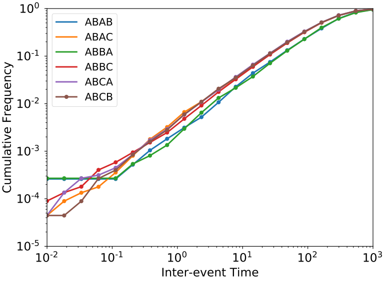
2. An SIS model: Two node states: infected and susceptible. Infected nodes pass on the infection to neighbouring susceptible nodes with rate λ. Infected nodes recover with rate μ.



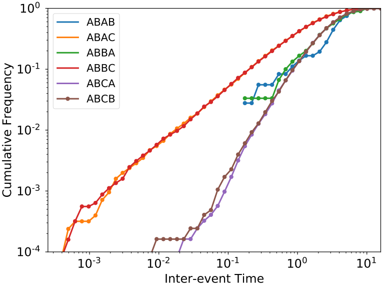
Blue = Random / Orange = SIS

Growth of the largest temporal component as a function of Δt.

One characteristic timescale in the random model, however there are multiple for the SIS model.



(a) Random Process



(b) SIS Process

Inter-event times conditioned on the motif type.

ABBA, ABAB, ABCB and ABCA motifs are far less prominent in the SIS model than the random process.

1. Investigate new embeddings of temporal networks and event graphs.

1. Investigate new embeddings of temporal networks and event graphs.

2. Use the event graph as a generative model for synthetic temporal networks.

1. Investigate new embeddings of temporal networks and event graphs.

2. Use the event graph as a generative model for synthetic temporal networks.

3. Compare node-based and event graph based dynamical process inference. Can the two be combined?

Thank you

References:

1. Analysing Collective Behaviour in Temporal Networks Using Event Graphs and Temporal Motifs
   arXiv:1801.10527
2. Event Graphs: Advances and Applications of Second-order Time-unfolded Temporal Network Models
   arXiv:1809.03457

Software: `https://github.com/empiricalstateofmind/eventgraphs`

Website: `andrewmellor.co.uk`

Email: `mellor@maths.ox.ac.uk`

Background

The **inter-event time** between two events e_i, e_j is given by

$$\tau_{ij} = \begin{cases} t_j - t_i & \text{if } t_j > t_i \\ 0 & \text{otherwise.} \end{cases}$$

If events have a duration then

$$\tau_{ij} = \begin{cases} t_j - (t_i + \delta_i) & \text{if } t_j > t_i + \delta_i \\ 0 & \text{otherwise.} \end{cases}$$

IETs are crucial to understanding dynamics and spreading processes on temporal networks.

Two events are Δt-adjacent if they share at least one node, and the time between the two events (inter-event time) is less than or equal to Δt.

As a function, Δt-adjacency can be written as

$$f(e_i, e_j) = \underbrace{(0 < \tau_{ij} \leq \Delta t)}_{\text{Temporal proximity}} \wedge \underbrace{(\{u_i, v_i\} \cap \{u_j, v_j\} \neq \emptyset)}_{\text{Topological proximity}}.$$

In this case, an event can be connected to any number of subsequent events provided the criteria is met.

We can also consider only the subsequent event for each node in the event previous event.

We define the set of all events that node u participates in after time t, given by

$$A_u^t = \{(u', v', t') \in E \text{ s.t. } u \in \{u', v'\} \text{ and } t' > t\}.$$

The joining rule is then given by

$$f_E(e_i, e_j) = (0 < \tau_{ij} \leq \Delta t) \wedge \left(\bigvee_{s \in \{u_i, v_i\}} (j = \min\{k | e_k \in A_s^{t_i}\}) \right).$$

Similarly, we can create a rule for simply the next event that *either* node participates in.

Temporal paths are are an important feature of a temporal network for dynamics.

We can adapt our connection rule to consider only pairs of events where a temporal path is formed (for directed events).

$$f(e_i, e_j) = \underbrace{(0 < \tau_{ij} \leq \Delta t)}_{\text{Temporal proximity}} \quad \wedge \quad \underbrace{(u_j = v_i)}_{\text{Target of event } i = \text{Source of event } j}$$

For certain temporal network processes it might be suitable to introduce a minimum time between events occurring.

For example, on transportation networks (rail, aviation) it is an unrealistic assumption to that a change can be made at a station/airport without first taking time to traverse from one vehicle to another.

$$f(e_i, e_j) = \underbrace{(\Delta t_1 < \tau_{ij} \leq \Delta t_2)}_{\text{Temporal proximity}} \quad \wedge \quad \underbrace{(u_j = v_i) \wedge (v_j \neq u_i)}_{\text{Path forming AND non-backtracking}}$$