DYNAMIC NETWORKS
(Dynamic of networks)
Most real world networks are dynamic

- Facebook friendship
  - People joining/leaving
  - Friend/Unfriend
- Twitter mention network
  - Each mention has a timestamp
  - Aggregated every day/month/year => still dynamic
- World Wide Web
- Urban network
- ...
DYNAMIC NETWORKS

• Most real world networks are dynamic
  ‣ Nodes can appear/disappear
  ‣ Edges can appear/disappear
  ‣ Nature of relations can change

• How to represent those changes?

• How to manipulate dynamic networks?
DYNAMIC NETWORKS

Dynamic Network Properties

Independently of the studied data, dynamic networks can have various properties:

- **Edge** presence can be **punctual** or **with duration**
- **Node** presence can be **unspecified, punctual** or **continuous**
- If **time is continuous**, it can be **bounded** on a period of analysis or **unbounded**
- If **nodes** have attributes, they can be **constant** or **time-dependent**
- If **edges** have weights, they can be **constant** or **time-dependent**
SEVERAL FORMALISMS

[Diagram of network connections and time series graphs]
TEMPORAL NETWORK

Collected dataset, for instance in (t,u,v) format

<table>
<thead>
<tr>
<th>Time</th>
<th>u</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>1353304100</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304100</td>
<td>1613</td>
<td>1672</td>
</tr>
<tr>
<td>1353304100</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304100</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304120</td>
<td>1492</td>
<td>1613</td>
</tr>
<tr>
<td>1353304120</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304120</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304140</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304160</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304160</td>
<td>1108</td>
<td>1601</td>
</tr>
<tr>
<td>1353304160</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304160</td>
<td>626</td>
<td>698</td>
</tr>
</tbody>
</table>

Examples:
- SocioPatterns
- Enron
Snapshots

<table>
<thead>
<tr>
<th>Time</th>
<th>Value1</th>
<th>Value2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1353304100</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304100</td>
<td>1613</td>
<td>1672</td>
</tr>
<tr>
<td>1353304100</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304100</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304120</td>
<td>1492</td>
<td>1613</td>
</tr>
<tr>
<td>1353304120</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304120</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304140</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304140</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304140</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304160</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304160</td>
<td>1108</td>
<td>1601</td>
</tr>
<tr>
<td>1353304160</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304160</td>
<td>626</td>
<td>698</td>
</tr>
</tbody>
</table>

Link Stream

<table>
<thead>
<tr>
<th>Time</th>
<th>Value1</th>
<th>Value2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1353304100</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304100</td>
<td>1613</td>
<td>1672</td>
</tr>
<tr>
<td>1353304100</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304100</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304120</td>
<td>1492</td>
<td>1613</td>
</tr>
<tr>
<td>1353304120</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304120</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304140</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304140</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304140</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304160</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304160</td>
<td>1108</td>
<td>1601</td>
</tr>
<tr>
<td>1353304160</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304160</td>
<td>626</td>
<td>698</td>
</tr>
</tbody>
</table>

Interval Graph

<table>
<thead>
<tr>
<th>Time</th>
<th>Value1</th>
<th>Value2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1353304100</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304100</td>
<td>1613</td>
<td>1672</td>
</tr>
<tr>
<td>1353304100</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304100</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304120</td>
<td>1492</td>
<td>1613</td>
</tr>
<tr>
<td>1353304120</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304120</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304140</td>
<td>1148</td>
<td>1644</td>
</tr>
<tr>
<td>1353304140</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304140</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304160</td>
<td>656</td>
<td>682</td>
</tr>
<tr>
<td>1353304160</td>
<td>1108</td>
<td>1601</td>
</tr>
<tr>
<td>1353304160</td>
<td>1632</td>
<td>1671</td>
</tr>
<tr>
<td>1353304160</td>
<td>626</td>
<td>698</td>
</tr>
</tbody>
</table>
DYNAMIC NETWORKS

Vocabulary

Many different names have been used to for networks changing with time, but there is no broad consensus in the literature on the meaning of those terms, unless they are used with an explicit reference to a paper defining those terms. Here is a list of the most popular:

- **Dynamic Networks** and **Dynamic Graphs**
- **Longitudinal Networks**
- **Evolving Graphs**
- **Link Streams** & **Stream Graphs** (Latapy, Viard, and Magnien 2018)
- **Temporal Networks**, **Contact Sequences** and **Interval Graphs** (Holme and Saramäki 2012)
- **Time Varying Graphs** (Casteigts et al. 2012)
ANALYZING DYNAMIC NETWORKS
ANALYZING DYNAMIC NETWORKS

• Few snapshots
• Slowly Evolving Networks (SEN)
• Degenerate/Unstable temporal networks
FEW SNAPSHOTSHOTS
FEW SNAPSHOTS

• The evolution is represented as a series of *a few* snapshots.

• Many changes between snapshots
  ‣ Cannot be visualized as a “movie”
FEW SNAPSHOTs

• Each snapshot can be studied as a static graph

• The evolution of the properties can be studied “manually”

• “Node X had low centrality in snapshot t and high centrality in snapshot t+n”
SLOWLY EVOLVING NETWORKS (SEN)
SLOWLY EVOLVING NETWORKS

• Edges change (relatively) slowly

• The network is well defined at any t
  ‣ Nodes/edges described by (long lasting) intervals
  ‣ Enough snapshots to track nodes

• A static analysis at every (relevant) t gives a dynamic vision

• No formal distinction with previous case (higher observation frequency)
SLOWLY EVOLVING NETWORKS

- Visualization
  - Problem of stability of node positions
SLOWLY EVOLVING NETWORKS

- Global graph properties

SLOWLY EVOLVING NETWORKS

- Centralities
TIME SERIES ANALYSIS

• TS analysis is a large field of research

• Time series: evolution of a value over time
  ‣ Stock market, temperatures…

• “Killer app”:
  ‣ Detection of periodic patterns
  ‣ Detection of anomalies
  ‣ Identification of global trends
  ‣ Evaluation of auto-correlation
  ‣ Prediction of future values

• e.g. ARIMA (Autoregressive integrated moving average)

UNSTABLE/DEGENERATE TEMPORAL NETWORKS

UNSTABLE TEMPORAL NETWORK

- The network at a given $t$ is not meaningful
- How to analyze such a network?
UNSTABLE TEMPORAL NETWORK
UNSTABLE TEMPORAL NETWORK

- Common solution: transform into SEN using aggregation/sliding windows
  - Information loss
  - How to chose a proper aggregation window size?

- New theoretical tools developed to deal with such networks
  - **Link Streams & Stream Graphs** (Latapy, Viard, and Magnien 2018)
  - **Temporal Networks, Contact Sequences and Interval Graphs** (Holme and Saramäki 2012)
  - **Time Varying Graphs** (Casteigts et al. 2012)
CENTRALITIES & NETWORK PROPERTIES IN STREAM GRAPHS
STREAM GRAPHS

Stream Graph (SG)- Definition

Stream Graphs have been proposed in\(^a\) as a generic formalism – it can represent any type of dynamic networks, continuous, discrete, with or without duration, with the objective or redefining typical notions of graphs on dynamic networks, including degenerate ones.

Let’s define a Stream Graph

\[ S = (T, V, W, E) \]

\( T \) | Set of Possible times (Discrete or Time intervals)
---|---
\( V \) | Set of Nodes
\( W \) | Vertices presence time \( V \times T \)
\( E \) | Edges presence time \( V \times V \times T \)

\(^a\)Latapy, Viard, and Magnien 2018.
Stream Graphs introduce some new notions mixing entities (nodes, edges) and time:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_t$</td>
<td><strong>Nodes At Time</strong>: set of nodes present at time $t$</td>
</tr>
<tr>
<td>$E_t$</td>
<td><strong>Edges At Time</strong>: set of edges present at time $t$</td>
</tr>
<tr>
<td>$G_t$</td>
<td><strong>Snapshot</strong>: Graph at time $t$, $G_t = (V_t, E_t)$</td>
</tr>
<tr>
<td>$v_t$</td>
<td><strong>Node-time</strong>: $v_t$ exist if node $v$ is present at time $t$</td>
</tr>
<tr>
<td>$(u, v)_t$</td>
<td><strong>Edge-time</strong>: $(u, v)_t$ exist if edge $(u, v)$ is present at time $t$, 0 otherwise</td>
</tr>
<tr>
<td>$T_u$</td>
<td><strong>Times Of Node</strong>: the set of times during which $u$ is present</td>
</tr>
<tr>
<td>$T_{uv}$</td>
<td><strong>Times Of Edge</strong>: the set of times during which edge $(u, v)$ is present</td>
</tr>
</tbody>
</table>
STREAM GRAPHS

| $N_u$  | **Node presence**: The fraction of the total time during which $u$ is present in the network $\frac{|T_u|}{|T|}$ |
|------|--------------------------------------------------|
| $L_{uv}$ | **Edge presence**: The fraction of the total time during which $(u, v)$ is present in the network $\frac{|T_{uv}|}{|T|}$ |
STREAM GRAPHS

SG - Redefining Graph notions

The general idea of redefining static network properties on Stream Graphs is that if the network stays unchanged along time, then properties computed on the stream graph should yield the same values as the same property computed on the aggregated graph.
The number/quantity of nodes in a stream graph is defined as the total presence time of nodes divided by the dataset duration. In general, it isn't an integer. More formally:

\[ N = \sum_{v \in V} N_v = \frac{|W|}{|T|} \]

For instance, \( N = 2 \) if there are 4 nodes present half the time, or two nodes present all the time.
STREAM GRAPHS

\[
\begin{align*}
N &= 2 \\
S_G - \text{Time-Entity designation} \\
T &\in \mathbb{R}^+ \\
G &\in \mathbb{G}^+ \\
V &\in \mathbb{V}^+ \\
E &\in \mathbb{E}^+ \\
\end{align*}
\]
STREAM GRAPHS

**SG - \( L \)**

The number of edges is defined as the total presence of edges divided by the total dataset duration. More formally:

\[
L = \sum_{(u,v), u, v \in V} L_{uv} = \frac{|E|}{|T|}
\]

For instance, \( L = 2 \) if there are 4 edges present half the time, or two edges present all the time.
STREAM GRAPHS

\[ L = 1 \]
STREAM GRAPHS

SG - Edge domain - $L_{\text{max}}$

In Stream Graphs, several possible definitions of $L_{\text{max}}$ could exist:

- Ignoring nodes duration: $L_{\text{max}}^1 = |V|^2$
- Ignoring nodes co-presence $L_{\text{max}}^2 = N^2$
- Taking nodes co-presence into account: $L_{\text{max}}^3 = \sum_{(u,v), u,v \in V} |T_u \cap T_v|$
STREAM GRAPHS

The density in static networks can be understood as the fraction of existing edges among all possible edges,

\[ d = \frac{L}{L_{\text{max}}} . \]

In the following, we will use \( L_{\text{max}}^3 \), as in Latapy et al.

Examples of graphs with \( N = 2 \) nodes, \( L = 1 \) link, but with different densities, respectively 0.75 (left) and 1 (right).
Examples of graphs with \( N = 2 \) nodes, \( L = 1 \) link, but with different densities, respectively 0.75 (left) and 1 (right).
STREAM GRAPHS

SG - Clusters & Substreams

In static networks, a cluster is a set of nodes, and we have defined an (induced) subgraph of this cluster as a graph composed of the nodes of the cluster and the edges existing between those nodes. In Stream Graphs, a clusters $C$ is as subset of $W$, and the corresponding (induced) substream $S(C) = (T, V, C, E(C))$, with $E(C) = \{(t, (u, v)) \in E, (t, u), (t, v) \in C\}$.

Example of subgraph and induced substream.
Having defined substreams and density, we can now naturally define a clique by analogy with static networks as a substream of density 1. A clique is said to be a maximal clique if it is not included in any other clique.

Red and Blue are the two maximal cliques of size three in this Stream Graph.
STREAM GRAPHS

SG - Neighborhood $N(u)$

The neighborhood $N(u)$ of node $u$ is defined as the cluster composed of node-times such as an edge-time exists between it and a node-time of $u$, i.e.,

$$N(u) = \{v_t, (u, v)_t \in E\}$$

SG - Degree $k(u)$

The degree $k(u)$ of node $u$ is defined as the quantity of node in the Neighborhood of node $u$, i.e.

$$k(u) = |N(u)|$$

Example, the neighborhood of node $c$ is highlighted in blue.

$$k(c) = 1.3$$

($|1.8| + |2.5| + |6.9|$).
STREAM GRAPHS

**SG - Ego-network**

The Ego network $G_u$ of node $u$ is defined as the substream induced by its neighborhood, i.e., $G_u = (T, V, N(u), E(N(u)))$.

**SG - Clustering coefficient**

The clustering coefficient $C(u)$ of node $u$ is defined as the density of the ego-network of $u$, i.e.,

$$ C(u) = d(N(u)) $$
PATHS AND DISTANCES IN STREAM GRAPHS
In a Stream Graph $S=(T,V,W,E)$, a **path** $P$ from node-time $x_\alpha$ to node-time $y_\omega$ is a sequence $(t_0, x, v_0), (t_1, v_0, v_1), \ldots, (t_k, v_k, y)$ of elements of $T \times V \times V$ such that $t_0 \geq \alpha, t_k \leq \omega, ((t_i, u_i, v_i)) \in E$. We say that $P$ **starts at** $t_0$, **arrives at** $t_k$, has **length** $k + 1$ and **duration** $t_k - t_0$.

Examples of two paths. The left one starts at 2, arrives at 5, has length 3 and duration 3. The right one starts at 2, arrives at 8, has length 4 and duration 6.
PATHS

**SG - Shortest - Fastest - Foremost**

- **Shortest Paths**, as in static networks, are paths of **minimal length**.
- **Fastest Paths** are paths of **minimal duration**.
- **Foremost Paths** are paths **arriving first**.

Furthermore, one can combine those properties, defining for instance:

- **Fastest shortest paths** (paths of minimum duration among those of minimal length)
- **Shortest fastest paths** (paths of minimal length among those of minimal duration)
PATHS

SG - Connected Components

Various definitions for connected components have been proposed for temporal networks, see (Latapy, Viard, and Magnien 2018) for details. One of the simplest one is the **weakly connected component**, defined such as two node-times belong to the same connected component if and only if there is a path from one to the other, *ignoring time*.

Example of a Stream Graph decomposed in 4 weakly connected components.
RANDOM MODELS

• In many cases, in network analysis, useful to compare a network to a randomized version of it
  ‣ Clustering coefficient, assortativity, modularity, …

• In a static graph, 2 main choices:
  ‣ Keep only the number of edges (ER model)
  ‣ Keep the number of edges and the degree of nodes (Configuration model)

• In dynamic networks, it is more complex…
RANDOM MODELS

**Snapshot Shuffling**

**Snapshot Shuffling** keeps the order of snapshots, randomize edges inside snapshots. Any random model for static network can be used, such as ER random graphs or the Configuration Model.

**Sequence Shuffling**

**Sequence Shuffling** keeps each snapshot identical, switch randomly their order.
RANDOM MODELS

**Link Shuffling**

*Link Shuffling* keeps activation time per node pairs, randomize the aggregated graph. For instance, a simple way to achieve this is to pick two node pairs at random (connected or not) of the aggregated graph, and to exchange activation time of these node pairs, e.g.:

![Link Shuffling Diagram](image1)

**Timeline Shuffling**

*Timeline Shuffling* keeps the aggregated graph, randomize edges activation time. For instance, a simple way to achieve this is to redistribute randomly activation period among all edges, e.g.:

![Timeline Shuffling Diagram](image2)
More constrained Shuffling

Variants of these shufflings with additional constraints have been proposed, for instance the **Local timeline shuffling**, randomizing events time edge by edge, or the **Weight constrained timeline shuffling**, randomizing events while conserving the number of observations for each edge. See (Gauvin et al. 2018) for details.

DYNAMIC COMMUNITY DETECTION

COMMUNITY DETECTION

Static networks

Sets of nodes

Dynamic Networks

Sets of periods of nodes

[Viard 2016]
COMMUNITY DETECTION

Community events (or operations)
IDENTITY PRESERVATION

Ship of Theseus [Plutarch., 75]
TO SUM UP ON DYNAMIC GRAPHS
TO SUM UP

• Currently, most practitioners still use the snapshot approaches
  ‣ No widespread framework
  ‣ No widespread coding libraries (pathpy, tnetwork, tacoma=>limited usage)
  ‣ Datasets still relatively limited

• But considered an important topic to work on
  ‣ Dynamic is everywhere
  ‣ Dynamic change everything in many cases