Graph Exploitation Testbed

Peter Jones and Eric Robinson

Graph Exploitation Symposium

April 18, 2012

This work was sponsored by the Office of Naval Research under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.
Graph Exploitation Testbed (GXT)

Streamlines evaluation of graph exploitation techniques for solving intelligence problems

• Goal 1: Dynamic generators and real-world data sets
• Goal 2: Dynamic attribute detection algorithms and models
• Goal 3: Algorithm performance analysis and model selection
• Goal 4: Testbed software engineering

YEAR 1 FOCUS AREA
Static subgraph detection

YEAR 2 FOCUS AREA
Dynamic graphs

Graph Exploitation Testbed
GIVEN: LARGE, COMPLEX DYNAMIC GRAPHS
IDENTIFY: SUBGRAPHS OF INTEREST
SUBJECT TO: EVALUATION CRITERIA

RED NODES: SUBGRAPH

DATA → ALGORITHMS → METRICS

Pd
Pfa
Attribute Detection on Transactional Data

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Static Graph</th>
<th>Transactional Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Representation</strong></td>
<td>Single adjacency matrix</td>
<td>Time-stamped sequence of transactions</td>
</tr>
<tr>
<td><strong>Primary Analytical Tool</strong></td>
<td>Linear algebra</td>
<td>Stochastic processes</td>
</tr>
<tr>
<td><strong>Typical Inference Problem</strong></td>
<td>Community detection</td>
<td>Attribute estimation/prediction</td>
</tr>
</tbody>
</table>
Diffusion on Graphs

- Modeling the flow of information is critical in many areas
  - Disease spreading in social networks
  - Adoption of novel ideas in a populations

- This has direct application to many intelligence problems
  - Spread of extremism in a population
  - Spread of IED tactics and materials throughout a extremist group
Outline

- Motivation
- Data sets
- Algorithms
- Metrics and Results
<table>
<thead>
<tr>
<th>Graph Size</th>
<th>Exploitability</th>
<th>Algorithmic Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>10s-100s of nodes</td>
<td>Enumerative</td>
<td>Combinatorial objects</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(e.g. all paths)</td>
</tr>
<tr>
<td>1000s-10,000s of</td>
<td>Polynomial</td>
<td>Global structure</td>
</tr>
<tr>
<td>nodes</td>
<td></td>
<td>(e.g. eigen decomposition)</td>
</tr>
<tr>
<td>Millions-billions of</td>
<td>Linear or sublinear</td>
<td>Local structure</td>
</tr>
<tr>
<td>nodes</td>
<td></td>
<td>(e.g. degree)</td>
</tr>
</tbody>
</table>

Algorithmic complexity impacts the classes of graphs that can be usefully considered.
### Graph Size and Exploitability

<table>
<thead>
<tr>
<th>Graph Size</th>
<th>Algorithmic Complexity</th>
<th>Exploitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>10s-100s of nodes</td>
<td>Enumerative</td>
<td>Combinatorial objects (e.g. all paths)</td>
</tr>
<tr>
<td>1000s-10,000s of nodes</td>
<td>Polynomial</td>
<td>Global structure (e.g. eigen decomposition)</td>
</tr>
<tr>
<td>Millions-billions of nodes</td>
<td>Linear or sublinear</td>
<td>Local structure (e.g. degree)</td>
</tr>
</tbody>
</table>

Algorithmic complexity impacts the classes of graphs that can be usefully considered.
Datasets (Dynamic Graphs)

- Each dataset contains a sequence of graph-structured transactions and attribute expressions
- Application areas span a broad range of fields
- Simulated and generated datasets supplement real-world examples

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Nodes</th>
<th># Nodes</th>
<th>Edges</th>
<th># Transactions</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enron</td>
<td>Email Accounts</td>
<td>154</td>
<td>Emails</td>
<td>38388</td>
<td>Email Topics</td>
</tr>
<tr>
<td>Bluegrass</td>
<td>Locations</td>
<td>275</td>
<td>Vehicle Tracks</td>
<td>474</td>
<td>Scripted Activity</td>
</tr>
<tr>
<td>Co-authors</td>
<td>Authors</td>
<td>6337</td>
<td>Co-authorship</td>
<td>14667</td>
<td>Sub-specialty</td>
</tr>
<tr>
<td>Meme Tracker</td>
<td>URLs</td>
<td>6574</td>
<td>Hyperlinks</td>
<td>88314</td>
<td>Meme expression</td>
</tr>
<tr>
<td>IDA</td>
<td>Locations</td>
<td>4483</td>
<td>Tracks</td>
<td>116720</td>
<td>Bad Activity</td>
</tr>
<tr>
<td>SEIRS</td>
<td>Agents</td>
<td>-</td>
<td>Interactions</td>
<td>-</td>
<td>Infection State</td>
</tr>
<tr>
<td>Point Process</td>
<td>Agents</td>
<td>-</td>
<td>Interactions</td>
<td>-</td>
<td>Rumor Knowledge</td>
</tr>
</tbody>
</table>

Real World

Simulated

Generators
Datasets (Dynamic Graphs)

- Each dataset contains a sequence of graph-structured transactions and attribute expressions
- Application areas span a broad range of fields
- Simulated and generated datasets supplement real-world examples

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Nodes</th>
<th># Nodes</th>
<th>Edges</th>
<th># Transactions</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enron Email</td>
<td>Email</td>
<td>154</td>
<td>Emails</td>
<td>38388</td>
<td>Email Topics</td>
</tr>
<tr>
<td>Bluegrass</td>
<td>Locations</td>
<td>275</td>
<td>Vehicle Tracks</td>
<td>474</td>
<td>Scripted Activity</td>
</tr>
<tr>
<td>Co-authors</td>
<td>Authors</td>
<td>6337</td>
<td>Co-authorship</td>
<td>14667</td>
<td>Sub-specialty</td>
</tr>
<tr>
<td>Meme Tracker</td>
<td>URLs</td>
<td>6574</td>
<td>Hyperlinks</td>
<td>88314</td>
<td>Meme expression</td>
</tr>
<tr>
<td>Simulated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDA</td>
<td>Locations</td>
<td>4483</td>
<td>Tracks</td>
<td>116720</td>
<td>Bad Activity</td>
</tr>
<tr>
<td>SEIRS</td>
<td>Agents</td>
<td>-</td>
<td>Interactions</td>
<td>-</td>
<td>Infection State</td>
</tr>
<tr>
<td>Point Process</td>
<td>Agents</td>
<td>-</td>
<td>Interactions</td>
<td>-</td>
<td>Rumor Knowledge</td>
</tr>
</tbody>
</table>
Each node is in one of four possible states:

- **Susceptible**: Not infected but capable of being infected by neighbors.
- **Exposed**: Infected but not capable of infecting others.
- **Infectious**: Infected and capable of infecting its neighbors.
- **Recovered**: Not infected nor capable of becoming infected.

The epidemic spreads through dynamic transactions occurring probabilistically according to some (static) baseline expectation.
IDA Simulated Vehicle Traffic

**Data Source**
- Data source consisted of over 100,000 vehicle movement tracks simulated by the Institute for Defense Analysis (IDA)
- IDA developed an IED threat scenario consisting of 24 red actors which was embedded into background traffic consisting of over 6000 gray actors

**Graph Description**
- Graph contains 4800 vertices and 100,000 edges
  - A vertex represents a location on the ground
  - An edge represents a vehicle track that moves from one location to another
- Each edge has two timestamps of when the vehicle left the source location and arrived at its destination
Memetracker Data Set

Data Source

- Data set contains appr. 100M records scraped from internet sources (blogs, news sites, etc.) from August 2008 through April 2009
- Each record includes a time stamp, all hyperlinks from the source document, and any identified “memes” (appr. 200M across all records)
- Subselected all records exhibiting eight major memes from Sept. 2008 through Oct. 2009

Graph Description

- Graph contains 6574 vertices and 16,269 edges
  - A vertex represents a root webpage
  - An edge represents a hyperlink between webpages

Graph Attributes

<table>
<thead>
<tr>
<th>Vertex (Website)</th>
<th>Edge (Hyperlinks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Time</td>
</tr>
</tbody>
</table>
| Type: blog, news site, aggregator, etc. | Direction
| Meme expression  |                   |

Outline

• Motivation

• Data sets

• Algorithms

• Metrics and Results
Algorithms

- **Static Algorithms (single adjacency matrix)**
  - Breadth-first Search
  - Spectral Modularity (Miller/Bliss)
  - Threat Propagation (Philips)

- **Dynamic Algorithms (sequence of adjacency matrices)**
  - Shortest Temporal Paths (Tang et al)
  - Dynamic Threat Propagation (Philips et al)
  - Dynamic Centrality (Lerman/Ghosh/Kang)

- **Related Dynamic Algorithms**
  - Dynamic Spectral Modularity (Miller/Bliss)
  - GraphScope (Sun et al)
  - EventRank (O’Madadhain/Smyth)
Shortest Temporal Paths

**Problem**

- Diffusion across a network must follow temporally directed paths
- Standard dynamic metrics only indirectly account for “time’s arrow”

**Shortest Temporal Path Approach**

- A journey from source node $s$ to destination node $d$ is a sequence of transactions $(u, v, t_{\text{DEP}}, t_{\text{ARR}})_k$ such that $u_0 = s$, $v_k = d$, $v_k = u_{k+1}$ and $t_{\text{ARR}_k} < t_{\text{DEP}_{k+1}}$
- The shortest temporal path distance measure between node $s$ and node $d$ is the minimum over all journeys between $s$ and $d$ of $[t_{\text{ARR}_k} - t_{\text{DEP}_0}]$
- Additional metrics (e.g. graph efficiency, temporal betweenness, etc.) can be defined using shortest temporal path as the fundamental metric

Shortest temporal path measures how quickly information could have travelled from $s$ to $d$
Dynamic Centrality

Problem

• One node’s influence on another may be either direct (1-hop connection) or indirect (multi-hop connection).

• Each path is a potential route for infection/ideology to spread, with shorter paths with lower temporal spreads being more likely contagion vectors.

Dynamic Centrality Approach

• Form a sequence of instantaneous adjacency matrices, $A(t)$.

• Aggregate $A(t)$ over time, using discounted weights to form the retained adjacency matrices $R(t)$.

• Create a sequence of dynamic centrality matrices $R^d(t)$ based on weighted path-wise combinations of $R(t)$.

• Finally, aggregate over all time to obtain a measure of each node’s influence on every other node, $RC(t)$.

Dynamic Centrality is a path-based measure of each node’s influence on all other nodes.

Dynamic Centrality

Problem

• One node’s influence on another may be either direct (1-hop connection) or indirect (multi-hop connection).
• Each path is a potential route for infection/ideology to spread, with shorter paths with lower temporal spreads being more likely contagion vectors.

Dynamic Centrality Approach

• Form a sequence of instantaneous adjacency matrices, $A(t)$
• Aggregate $A(t)$ over time, using discounted weights to form the retained adjacency matrices $R(t)$
• Create a sequence of dynamic centrality matrices $R^d(t)$ based on weighted path-wise combinations of $R(t)$
• Finally, aggregate over all time to obtain a measure of each node’s influence on every other node, $RC(t)$

Dynamic Centrality is a path-based measure of each node’s influence on all other nodes.

MLG'10 Washington, DC USA
Dynamic Centrality

Problem

• One node’s influence on another may be either direct (1-hop connection) or indirect (multi-hop connection)
• Each path is a potential route for infection/ideology to spread, with shorter paths with lower temporal spreads being more likely contagion vectors

Dynamic Centrality Approach

• Form a sequence of instantaneous adjacency matrices, $A(t)$
• Aggregate $A(t)$ over time, using discounted weights to form the retained adjacency matrices $R(t)$
• Create a sequence of dynamic centrality matrices $R^d(t)$ based on weighted path-wise combinations of $R(t)$
• Finally, aggregate over all time to obtain a measure of each node’s influence on every other node, $RC(t)$

Dynamic Centrality is a path-based measure of each node’s influence on all other nodes

Dynamic Centrality

Problem

• One node’s influence on another may be either direct (1-hop connection) or indirect (multi-hop connection).

• Each path is a potential route for infection/ideology to spread, with shorter paths with lower temporal spreads being more likely contagion vectors.

Dynamic Centrality Approach

• Form a sequence of instantaneous adjacency matrices, $A(t)$.

• Aggregate $A(t)$ over time, using discounted weights to form the retained adjacency matrices $R(t)$.

• Create a sequence of dynamic centrality matrices $R^d(t)$ based on weighted path-wise combinations of $R(t)$.

• Finally, aggregate over all time to obtain a measure of each node’s influence on every other node, $RC(t)$.

$RC(t)$: Cumulative Dynamic Centrality Matrix

Dynamic Centrality is a path-based measure of each node’s influence on all other nodes.
Dynamic Threat Propagation

Dynamic threat propagation estimates a vertex’s time-varying community membership based on dynamic interactions.

**Problem**
- Vertices in a community are defined by a coordinated set interactions that unfold over time.
- Membership in a community is a time-varying property defining when a vertex is acting as a member of the community.

**Approach**
- Propagate a kernel (e.g., a Gaussian) for every interaction.
- Center kernels at time of each interaction.
- Combine all kernel to form smoothly vary function of membership over time.

\[
P_{s_i}(t) = \alpha \left( \frac{\lambda}{E(s_i)} \sum_{e_n \in E(s_i)} g(t | e_y) + (1 - \lambda) \max_{e_n \in E(s_i)} g(t | e_y) \right)
\]
Outline

• Motivation

• Data sets

• Algorithms

• Metrics and Results
Metrics

• Two fundamental inference tasks in ‘tipped’ or ‘cued’ diffusion networks
  – Detect if information ever reaches a node
  – Estimate when information reaches a node

• Measuring algorithm performance on detection task
  – Receiver Operator Characteristic
  – Figure of merit: Area under Curve (AUC)

• Measuring algorithm performance on estimation task
  – Scatter plot of score vs. time of info arrival
  – Figure of merit: Ratio of Covariant Eigenvalues

AUC: 0.81 (DC), 0.79 (STP)

$\lambda_1/\lambda_2 = 3.25$
Results (SEIRS)

- Experiment with SEIRS simulated data set
  - Initially, one node infected, all others susceptible
  - Infection diffuses over time
  - Simulation length = 1000 steps

- Goal: detect which nodes become infected over the course of the simulation

- Result: Dynamic Centrality significantly outperforms all other detection algorithms
Results (IDA)

• Experiment with IDA data set
  – Simulated vehicle tracks
  – Fully developed threat scenario embedded in background traffic

• Goal: detect which nodes are participating in the threat scenario

• Result: Dynamic Centrality slightly outperforms Dynamic Threat Propagation
  – Dynamic algorithms generally outperform static algorithms
Results (Memetracker)

• Experiment with Memetracker data set
  – Collect time-stamped hyper-links between websites
  – Record instances of popular quotations (memes)

• Goal: detect nodes expressing a specified meme* during the two-month experiment period

• Result: Extremely difficult detection task
  – Most algorithms perform slightly better than chance
  – Simplest algorithm performs the best

* “do not have to be scared of as President”
Results (Memetracker ‘Patient Zero’)

- Experiment with Memetracker data set
  - Collect time-stamped hyperlinks between websites
  - Record instances of popular quotations (memes)
- Goal: detect nodes expressing a specified meme* during the two-month experiment period
- Result: Tipping off ‘patient zero’ improves some algorithms’ performance

* “do not have to be scared of as President”
Conclusions

• Graph Exploitation Testbed provides researchers a powerful tool for developing and testing inference algorithms on graph-structured data

• Estimating diffusion processes on networks presents a challenging and important inference problem
  – Open literature provides a modest but growing set of algorithms for determining node-to-node similarity based on transactional data
  – Additional algorithms developed for the GXT program perform comparably to current state of the art

• Future research directions
  – Learn dataset characterizations to predict algorithm performance
  – Consider other prediction/estimation tasks (e.g. temporal link prediction)