Degeneracy-Based Mining of Social and Information Networks: Dynamics and Applications

Fragkiskos D. Malliaros

Computer Science Laboratory
École Polytechnique, France

Joint work with: M. Vazirgiannis, M.E. Rossi, C. Giatsidis,
D.M. Thilikos, K. Skianis

Complex Networks Team, LIP6
Pierre and Marie Curie University (UPMC)

April 1, 2016
Networks allow to model relationships between entities

General-purpose language for describing real-world systems
Facebook
1 Billion Users (Aug. 2015)

Source: https://www.facebook.com/zuck
Collaboration networks
(Co-authorship)
Term co-occurrence network
(*David Copperfield* novel by Charles Dickens)
Networks are Everywhere

- **Technological networks:**
  - Internet
  - Telephone networks
  - Power grid
  - Road, airline and rail networks

- **Information networks:**
  - World Wide Web
  - Blog networks
  - Citation networks
  - Textual networks

- **Social networks:**
  - Collaboration networks
  - Organizational networks
  - Communication networks

- **Biological networks:**
  - Neural networks
  - Protein-protein interaction networks
  - Gene regulatory networks
  - Food webs
Motivation: Analytics in the Network Science Era

Study large real-world networks

1. Design of effective and efficient graph mining algorithms

2. Apply the algorithms to analyze and understand the structure and dynamics of complex systems

3. Utilize the extracted knowledge to solve real-world applications
Overview

**Models, tools and observations** for problems in the area of mining social and information networks

We build upon *computationally efficient graph mining* methods to:

1. Design models for analyzing the structure and dynamics of real-world networks
   - Unravel properties that can further be used in practical applications

2. Develop algorithmic tools for large-scale analytics on data with:
   - Inherent graph structure (e.g., social networks)
   - Without inherent graph structure (e.g., text)
Overview

Models, tools and observations for problems in the area of mining social and information networks

We build upon computationally efficient graph mining methods to:

1. Design models for analyzing the structure and dynamics of real-world networks
   - Unravel properties that can further be used in practical applications

2. Develop algorithmic tools for large-scale analytics on data with:
   - Inherent graph structure (e.g., social networks)
   - Without inherent graph structure (e.g., text)
Contributions

Overview

Models, tools and observations for problems in the area of mining social and information networks

We build upon computationally efficient graph mining methods to:

1. Design models for analyzing the structure and dynamics of real-world networks
   - Unravel properties that can further be used in practical applications

2. Develop algorithmic tools for large-scale analytics on data with:
   - Inherent graph structure (e.g., social networks)
   - Without inherent graph structure (e.g., text)
Overview

**Models, tools and observations** for problems in the area of mining social and information networks

We build upon **computationally efficient graph mining** methods to:

1. Design models for analyzing the structure and dynamics of real-world networks
   - Unravel properties that can further be used in practical applications

2. Develop algorithmic tools for large-scale analytics on data with:
   - Inherent graph structure (e.g., social networks)
   - Without inherent graph structure (e.g., text)
Overview

**Dynamics of Real Networks**

Modeling Social Engagement  
[ MV, '13 ]

Vulnerability Assessment  
[ MV, '15 ]
Overview

Dynamics of Real Networks

- Modeling Social Engagement
  - [MV, ’13]

- Vulnerability Assessment
  - [MV, ’15]

Identification of Influential Spreaders

- [RMV, ’15], [MRV, ’16]

Graph Clustering Community Detection

- [GMTV, ’14]

Algorithmic Tools for Data Analytics
Overview

Dynamics of Real Networks

- Modeling Social Engagement
  - [MV, '13]

- Vulnerability Assessment
  - [MV, '15]

Degeneracy-based Graph Mining

- Identification of Influential Spreaders
  - [RMV, '15], [MRV, '16]

- Graph Clustering Community Detection
  - [GMTV, '14]

Algorithmic Tools for Data Analytics
$k$-core Decomposition in Networks
Degeneracy and $k$-core Decomposition

Example

Core number $c_1 = 1$
Core number $c_1 = 2$
Core number $c_1 = 3$

Graph Degeneracy $\delta^*(G) = 3$

$G_0 = G$
$G_1 = 1$-core of $G$
$G_2 = 2$-core of $G$
$G_3 = 3$-core of $G$

$G_0 \supseteq G_1 \supseteq G_2 \supseteq G_3$

[Seidman, '83]
**Algorithm: \( k \)-core decomposition**

**Input:** Undirected graph \( G = (V, E) \)

**Output:** Core numbers \( c(v), \forall v \in V \)

1: \( i \leftarrow 0 \)
2: \textbf{while} \(|V| > 0\) \textbf{do}
3: \hspace{1em} \textbf{while} \( \exists v : d(v) \leq i \) \textbf{do}
4: \hspace{2em} \( c(v) \leftarrow i \)
5: \hspace{2em} \( V \leftarrow V \setminus \{v\} \)
6: \hspace{2em} \( E \leftarrow E \setminus \{(u, v) | u \in V\} \)
7: \hspace{1em} \textbf{end while}
8: \( i \leftarrow i + 1 \)
9: \textbf{end while}

More about the decomposition

- Time complexity \( \mathcal{O}(|E|) \) [Batagelj and Zaversnik, ’03]
- Extensions to various graph types and computation models
**Algorithm: k-core decomposition**

**Input:** Undirected graph $G = (V, E)$

**Output:** Core numbers $c(v)$, $\forall v \in V$

1: $i \leftarrow 0$

2: while $|V| > 0$ do

3: while $\exists v : d(v) \leq i$ do

4: $c(v) \leftarrow i$

5: $V \leftarrow V \setminus \{v\}$

6: $E \leftarrow E \setminus \{(u, v) | u \in V\}$

7: end while

8: $i \leftarrow i + 1$

9: end while

More about the decomposition

- Time complexity $\mathcal{O}(|E|)$ [Batagelj and Zaversnik, ’03]

- Extensions to various graph types and computation models
Modeling Engagement Dynamics in Social Graphs
Modeling Engagement Dynamics

Objectives and contributions

- **Goal**: model and quantify the engagement in social graphs
- **User engagement** refers to the extend that an individual is encouraged to participate in the activities of a community
- Closely related property to the one of **node departure dynamics**

**Main contributions**

- Measures of engagement (node and graph level)
- Experiments: Properties and dynamics of real graphs
- Implications of our study on a new problem of robustness/vulnerability assessment
Modeling Engagement Dynamics

Objectives and contributions

- **Goal:** model and quantify the engagement in social graphs
- **User engagement** refers to the extend that an individual is encouraged to participate in the activities of a community
- Closely related property to the one of **node departure dynamics**

**Main contributions**

- Measures of engagement (node and graph level)
- Experiments: Properties and dynamics of real graphs
- Implications of our study on a new problem of robustness/vulnerability assessment
Goal

Model social engagement from a network-wise point of view

- Consider information only about the underlying graph structure
Goal

Model social engagement from a network-wise point of view

- Consider information only about the underlying graph structure

- Each individual that participates in a social activity, derives a benefit
  - The benefit emanates from his/her neighborhood

- The benefit of each individual is affected by the degree of interaction among its neighbors [Ugander et al., PNAS ’12]
  - Interactions among friends, can increase user’s benefit
Suppose now that a user decides to depart

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an epidemic, forming a cascade of individual departures
Suppose now that a user decides to **depart**

- Direct effects in his neighborhood $\rightarrow$ Some of his friends may also decide to depart
- A departure can become an **epidemic**, forming a **cascade** of individual departures
Suppose now that a user decides to depart

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an epidemic, forming a cascade of individual departures
Suppose now that a user decides to **depart**

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an **epidemic**, forming a **cascade** of individual departures
Suppose now that a user decides to depart

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an epidemic, forming a cascade of individual departures
Suppose now that a user decides to **depart**

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an **epidemic**, forming a **cascade** of individual departures
Suppose now that a user decides to depart

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an epidemic, forming a cascade of individual departures
Suppose now that a user decides to depart

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an epidemic, forming a cascade of individual departures
Suppose now that a user decides to **depart**

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an **epidemic**, forming a **cascade** of individual departures
Suppose now that a user decides to **depart**

- Direct effects in his neighborhood → Some of his friends may also decide to depart
- A departure can become an **epidemic**, forming a **cascade** of individual departures

**Direct-benefit effects:**
To incur an explicit benefit by remaining engaged

The decision of a node should align with the one of its neighbors

[Easley and Kleinberg, ’10]
Model Description

- $\mathcal{X} = \{0, 1\}$: set of strategies (i.e., leave or stay)
- $x = [x_1, x_2, \ldots, x_n]$: vector that denotes the decision of each node $i \in V$

- **Node payoff function:**
  \[ \Pi_i(x) = \text{benefit}(x_i, \sum_{j \in \mathcal{N}_i} x_j) - \text{cost}(x_i), \mathcal{N}_i = \{j \in V : (i, j) \in E\} \]

- **Benefit function:** depends on node’s own decision and the aggregate decision of the neighbors
- **Cost function:** does not need to be known a priori $\rightarrow$ remain engaged if $\text{cost} \leq \text{benefit}$ (non-negative payoff)

---

Equilibrium Property

The best response of each node $i \in V$ corresponds to the core number $c_i$

[Manshadi and Johari, ’09], [Harkins, ’13], [Bhawalkar et al., ’11]
Model Description

- $X = \{0, 1\}$: set of strategies (i.e., leave or stay)
  - $x = [x_1, x_2, \ldots, x_n]$: vector that denotes the decision of each node $i \in V$

- Node payoff function:
  - $\Pi_i(x) = \text{benefit}(x_i, \sum_{j \in N_i} x_j) - \text{cost}(x_i)$, $N_i = \{j \in V : (i, j) \in E\}$
    - **Benefit function**: depends on node’s own decision and the aggregate decision of the neighbors
    - **Cost function**: does not need to be known a priori → remain engaged if cost $\leq$ benefit (non-negative payoff)

Equilibrium Property

The best response of each node $i \in V$ corresponds to the core number $c_i$

[Manshadi and Johari, ’09], [Harkins, ’13], [Bhawalkar et al., ’11]
Model Description

- $\mathcal{X} = \{0, 1\}$: set of strategies (i.e., leave or stay)
  - $x = [x_1, x_2, \ldots, x_n]$: vector that denotes the decision of each node $i \in V$

- **Node payoff function**: 
  - $\Pi_i(x) = \text{benefit}(x_i, \sum_{j \in \mathcal{N}_i} x_j) - \text{cost}(x_i), \mathcal{N}_i = \{j \in V : (i, j) \in E\}$

  - **Benefit function**: depends on node’s own decision and the aggregate decision of the neighbors
  - **Cost function**: does not need to be known a priori → remain engaged if cost $\leq$ benefit (non-negative payoff)

**Equilibrium Property**

The best response of each node $i \in V$ corresponds to the core number $c_i$ 
[Manshadi and Johari, ’09], [Harkins, ’13], [Bhawalkar et al., ’11]
Proposed Engagement Measures

Node engagement

Proposition (Node Engagement)

The engagement level $e_i$ of each node $i \in V$ is defined as its core number $c_i$

- Nodes with higher core number $\rightarrow$ better engagement

Prob. of departure vs. core number

More refined modeling explanation of the departure dynamics in social graphs [Wu et al., ’13]

- Active users: core of the graph
- Inactive users: periphery of the graph
- The departure of nodes is proportional to their position in the graph
Proposed Engagement Measures

Engagement subgraphs

**Definition (**)\(k\)-Engagement Subgraph \(G_k\)**

The graph which is induced by the nodes \(i \in V\) with engagement level \(e_i \geq k\)

**Proposition (Max-Engagement Subgraph \(G_{e_{\text{max}}}\)**

- Maximum engagement level of the graph: \(e_{\text{max}} = \delta^* (G)\) (degeneracy of the graph)
- **Max-Engagement subgraph**: induced by the nodes with engagement \(e = e_{\text{max}}\)
Proposed Engagement Measures

Graph engagement

**Definition (Graph Engagement $\mathcal{E}_G$)**

The total engagement level of a graph $G$ is defined as the area under curve of the normalized CCDF $\mathcal{F}(e) = \Pr(X \geq e)$

- Values in the range $[0, 1]$
- Higher $\mathcal{E}_G$ values $\rightarrow$ higher total engagement

Schematic representation of $\mathcal{E}_G$

Complete graph $K_n$
## Datasets

### Basic characteristics of real-world networks

<table>
<thead>
<tr>
<th>Graph</th>
<th># Nodes</th>
<th># Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACEBOOK</td>
<td>63,392</td>
<td>816,886</td>
</tr>
<tr>
<td>YOUTUBE</td>
<td>1,134,890</td>
<td>2,987,624</td>
</tr>
<tr>
<td>SLASHDOT</td>
<td>77,360</td>
<td>546,487</td>
</tr>
<tr>
<td>EPINIONS</td>
<td>75,877</td>
<td>405,739</td>
</tr>
<tr>
<td>EMAIL-EUALL</td>
<td>224,832</td>
<td>340,795</td>
</tr>
<tr>
<td>EMAIL-ENRON</td>
<td>33,696</td>
<td>180,811</td>
</tr>
<tr>
<td>CA-GR-QC</td>
<td>4,158</td>
<td>13,428</td>
</tr>
<tr>
<td>CA-ASTRO-PH</td>
<td>17,903</td>
<td>197,031</td>
</tr>
<tr>
<td>CA-HEP-PH</td>
<td>11,204</td>
<td>117,649</td>
</tr>
<tr>
<td>CA-HEP-TH</td>
<td>8,638</td>
<td>24,827</td>
</tr>
<tr>
<td>CA-COND-MAT</td>
<td>21,363</td>
<td>91,342</td>
</tr>
<tr>
<td>DBLP</td>
<td>404,892</td>
<td>1,422,263</td>
</tr>
</tbody>
</table>
Properties of $k$-Engagement Subgraphs

Size distribution

- Almost skewed distribution
- Small subgraphs with high engagement $k$
- Different behavior between real and randomly rewired graphs
Node Engagement vs. Node Degree

High degree nodes are possible to have low engagement
Graphs’ Engagement Properties

Engagement index $\mathcal{E}_G$

- **FACEBOOK** has the maximum engagement index $\mathcal{E}_G$
  - A relatively high fraction of nodes has high (normalized) engagement $e$
- **DBLP** shows the lower engagement index $\mathcal{E}_G$ in the collaboration graphs
  - Possible explanation: significant number of “relatively” new authors with low engagement
Discussion

- Engagement metrics and dynamics in social graphs
  - Local and global metrics
  - $k$-core decomposition-based
  - Limitation: Ground truth data to further evaluate the proposed metrics

- Engagement-based vulnerability assessment
Vulnerability Assessment in Social Networks under Cascade-Based Node Departures
Vulnerability Assessment in Real Networks

- Well-studied problem in the broad area of network science
  - [Albert and Barabási, Rev. Mod. Phys. ’02]

- Networks with skewed degree distribution
  - Robust against random failures
  - Vulnerable under targeted attacks to hubs
  - [Albert et al., Nature ’00]
Engagement-based Vulnerability Assessment

Motivation of this work

- In social networks
  - Instead of **degree-based** failures and attacks ...
  - Departures based on the **engagement** level

- The engagement level of nodes is not accurately captured by the node degree

- Well-known **degree-based** notions of robustness assessment may not accurately capture this feature of social networks
  - [Albert et al., Nature ’00]
  - [Cohen et al., Phys. Rev. Lett. ’01]
  - [Motter and Lai, Phys. Rev. E ’02]
  - ...

Engagement-based Vulnerability Assessment

Motivation of this work

- In social networks
  - Instead of **degree-based** failures and attacks ...  
  - Departures based on the **engagement** level

- The engagement level of nodes is not accurately captured by the node degree

- Well-known **degree-based** notions of robustness assessment may not accurately capture this feature of social networks
  - [Albert et al., Nature ’00]
  - [Cohen et al., Phys. Rev. Lett. ’01]
  - [Motter and Lai, Phys. Rev. E ’02]
  - ...
Engagement-based Vulnerability Assessment

Motivation of this work

- In social networks
  - Instead of **degree-based** failures and attacks ...
  - Departures based on the **engagement** level

- The engagement level of nodes is not accurately captured by the node degree

- Well-known **degree-based** notions of robustness assessment may not accurately capture this feature of social networks
  - [Albert et al., Nature ’00]
  - [Cohen et al., Phys. Rev. Lett. ’01]
  - [Motter and Lai, Phys. Rev. E ’02]
  - ...
Contributions of this Work

**Goal:** novel concept of **engagement-based vulnerability assessment** in social graphs

- Cascading Departure (CasD) model
  - Captures the cascading disengagement effect due to the departure of a node
- Vulnerability assessment under node departures
  - Experimental results on real networks
Contributions of this Work

**Goal:** novel concept of *engagement-based vulnerability assessment* in social graphs

- Cascading Departure (CasD) model
  - Captures the cascading disengagement effect due to the departure of a node

- Vulnerability assessment under node departures
  - Experimental results on real networks
Contributions of this Work

**Goal:** novel concept of *engagement-based vulnerability assessment* in social graphs

- Cascading Departure (CasD) model
  - Captures the cascading disengagement effect due to the departure of a node
- Vulnerability assessment under node departures
  - Experimental results on real networks
Disengagement Epidemic Model
How to model user departures – the idea

- We introduce a model of cascading departures in social graphs
- Main idea:
  - The departure of a node can cause direct effects in its neighborhood
  - Some of the neighbors may also decide to depart
  - It can lead to an epidemic of disengagement (or churn effect)
  - Model the process using the $k$-core decomposition and its connection to the engagement dynamics
Disengagement Epidemic Model
How to model user departures – the idea

- We introduce a model of **cascading departures** in social graphs.
- Main idea:
  - The departure of a node can cause direct effects in its neighborhood.
  - Some of the neighbors may also decide to depart.
  - It can lead to an **epidemic of disengagement** (or churn effect).
  - Model the process using the $k$-core decomposition and its connection to the engagement dynamics.
Cascading Departure (CasD) Model

**Algorithm**: CasD model

**Input**: Undirected graph \( G = (V, E) \) and node \( v \in V \)

**Output**: Set of removed nodes \( R \)

1: \( c = [c_1, c_2, \ldots, c_{|V|}] = k\text{-core}\text{-decomposition}(G) \)
2: \( \tilde{V} = V \setminus \{v\} \) \{Remove node \( v \) and the incident edges\}
3: repeat
4: \{Recompute the core number of each node \( i \in \tilde{V} \}\}
5: \( \tilde{c} = k\text{-core}\text{-decomposition}(G) \)
6: \{Normalize the core numbers \( \tilde{c}_i \) into the interval \([0, 1]\)\}
7: \( \tilde{c}_i^{\text{norm}} = \frac{\tilde{c}_i - \min(\tilde{c})}{\max(\tilde{c}) - \min(\tilde{c})}, \forall i \in \tilde{V} \)
8: for all \( i \in \tilde{V} \) do
9: \( \text{if } \tilde{c}_i < c_i \text{ then} \)
10: \( \text{Remove node } i \text{ from } G \text{ with probability:} \)
11: \( \Pr (\tilde{V} = \tilde{V} \setminus \{i\}) = 1 - \tilde{c}_i^{\text{norm}} \) (also \( R = R \cup \{i\} \))
12: end if
13: end for
14: until No more nodes are removed
15: return Set of affected (removed) nodes \( R \)
Cascading Departure (CasD) Model

**Algorithm:** CasD model

**Input:** Undirected graph \( G = (V, E) \) and node \( v \in V \)

**Output:** Set of removed nodes \( R \)

1: \( c = [c_1, c_2, \ldots, c_{|V|}] = k\text{-core}\_decomposition(G) \)

2: \( \tilde{V} = V \setminus \{v\} \) {Remove node \( v \) and the incident edges}

3: repeat

4: \{Recompute the core number of each node \( i \in \tilde{V} \)} \( \tilde{c} = k\text{-core}\_decomposition(G) \)

5: \{Normalize the core numbers \( \tilde{c}_i \) into the interval \([0, 1]\)} \( \tilde{c}_i^\text{norm} = \frac{\tilde{c}_i - \min(\tilde{c})}{\max(\tilde{c}) - \min(\tilde{c})}, \forall i \in \tilde{V} \)

6: for all \( i \in \tilde{V} \) do

7: if \( \tilde{c}_i < c_i \) then

8: Remove node \( i \) from \( G \) with probability:

\[ \Pr (\tilde{V} = \tilde{V} \setminus \{i\}) = 1 - \tilde{c}_i^\text{norm} \] (also \( R = R \cup \{i\} \))

9: end if

10: end for

11: until No more nodes are removed

12: return Set of affected (removed) nodes \( R \)
Cascading Departure (CasD) Model

**Algorithm:** CasD model

**Input:** Undirected graph \( G = (V, E) \) and node \( v \in V \)

**Output:** Set of removed nodes \( R \)

1: \( c = [c_1, c_2, \ldots, c_{|V|}] = k\text{-core}\_decomposition(G) \)

2: \( \tilde{V} = V \setminus \{v\} \) \{Remove node \( v \) and the incident edges\}

3: repeat

4: \{Recompute the core number of each node \( i \in \tilde{V} \} \)

\( \tilde{c} = k\text{-core}\_decomposition(G) \)

5: \{Normalize the core numbers \( \tilde{c}_i \) into the interval \([0, 1]\} \)

\( \tilde{c}_{i,\text{norm}} = \frac{\tilde{c}_i - \min(\tilde{c})}{\max(\tilde{c}) - \min(\tilde{c})}, \forall i \in \tilde{V} \)

6: for all \( i \in \tilde{V} \) do

7: if \( \tilde{c}_i < c_i \) then

8: Remove node \( i \) from \( G \) with probability:

\( \Pr (\tilde{V} = \tilde{V} \setminus \{i\}) = 1 - \tilde{c}_{i,\text{norm}} \) \{also \( R = R \cup \{i\}\} \)

9: end if

10: end for

11: until No more nodes are removed

12: return Set of affected (removed) nodes \( R \)
Cascading Departure (CasD) Model

**Algorithm:** CasD model

**Input:** Undirected graph $G = (V, E)$ and node $v \in V$

**Output:** Set of removed nodes $R$

1: $c = [c_1, c_2, \ldots, c_{|V|}] = k$-core_decomposition($G$)

2: $\tilde{V} = V \setminus \{v\}$ \{Remove node $v$ and the incident edges\}

3: repeat

4: \{Recompute the core number of each node $i \in \tilde{V}$\}

   $\tilde{c} = k$-core_decomposition($G$)

5: \{Normalize the core numbers $\tilde{c}_i$ into the interval $[0, 1]$\}

   $\tilde{c}_i^{\text{norm}} = \frac{\tilde{c}_i - \min(\tilde{c})}{\max(\tilde{c}) - \min(\tilde{c})}$, $\forall i \in \tilde{V}$

6: for all $i \in \tilde{V}$ do

7: if $\tilde{c}_i < c_i$ then

8: Remove node $i$ from $G$ with probability:

   $\Pr (\tilde{V} = \tilde{V} \setminus \{i\}) = 1 - \tilde{c}_i^{\text{norm}}$ \{also $R = R \cup \{i\}$\}

9: end if

10: end for

11: until No more nodes are removed

12: return Set of affected (removed) nodes $R$
Cascading Departure (CasD) Model

**Algorithm: CasD model**

**Input:** Undirected graph $G = (V, E)$ and node $v \in V$

**Output:** Set of removed nodes $R$

1. $c = [c_1, c_2, \ldots, c_{|V|}] = k$-core_decomposition($G$)
2. $\tilde{V} = V \setminus \{v\}$ {Remove node $v$ and the incident edges}
3. repeat
4. {Recompute the core number of each node $i \in \tilde{V}$}
   $$\tilde{c} = k$-core_decomposition($G$)
5. {Normalize the core numbers $\tilde{c}_i$ into the interval $[0, 1]$}
   $$\tilde{c}_i^{\text{norm}} = \frac{\tilde{c}_i - \min(\tilde{c})}{\max(\tilde{c}) - \min(\tilde{c})}, \forall i \in \tilde{V}$$
6. for all $i \in \tilde{V}$ do
7. if $\tilde{c}_i < c_i$ then
8. Remove node $i$ from $G$ with probability:
   $$\Pr (\tilde{V} = \tilde{V} \setminus \{i\}) = 1 - \tilde{c}_i^{\text{norm}} \quad \text{(also } R = R \cup \{i\})$$
9. end if
10. end for
11. until No more nodes are removed
12. return Set of affected (removed) nodes $R$
Cascading Departure (CasD) Model

Algorithm: CasD model

Input: Undirected graph \( G = (V, E) \) and node \( v \in V \)
Output: Set of removed nodes \( R \)

1: \( c = [c_1, c_2, \ldots, c_{|V|}] = k\text{-core\_decomposition}(G) \)
2: \( \tilde{V} = V \setminus \{v\} \) \{Remove node \( v \) and the incident edges\}
3: repeat
    4: \{Recompute the core number of each node \( i \in \tilde{V} \}\}
        \( \tilde{c} = k\text{-core\_decomposition}(G) \)
    5: \{Normalize the core numbers \( \tilde{c}_i \) into the interval \([0, 1]\)\}
        \( \tilde{c}_i^{\text{norm}} = \frac{\tilde{c}_i - \min(\tilde{c})}{\max(\tilde{c}) - \min(\tilde{c})}, \forall i \in \tilde{V} \)
    6: for all \( i \in \tilde{V} \) do
        7: if \( \tilde{c}_i < c_i \) then
            8: Remove node \( i \) from \( G \) with probability:
                \( \Pr(\tilde{V} = \tilde{V} \setminus \{i\}) = 1 - \tilde{c}_i^{\text{norm}} \) (also \( R = R \cup \{i\} \))
        9: end if
    10: end for
8: until No more nodes are removed
12: return Set of affected (removed) nodes \( R \)
Cascading Departure (CasD) Model

**Algorithm:** CasD model

**Input:** Undirected graph \( G = (V, E) \) and node \( v \in V \)

**Output:** Set of removed nodes \( R \)

1: \( c = [c_1, c_2, \ldots, c_{|V|}] = k\text{-core}\text{-decomposition}(G) \)
2: \( \tilde{V} = V \setminus \{v\} \) {Remove node \( v \) and the incident edges}
3: repeat
4: \{Recompute the core number of each node \( i \in \tilde{V} \}
5: \( \tilde{c} = k\text{-core}\text{-decomposition}(G) \)
6: \{Normalize the core numbers \( \tilde{c}_i \) into the interval \([0, 1]\)\}
7: \( \tilde{c}_i^{\text{norm}} = \frac{\tilde{c}_i - \min(\tilde{c})}{\max(\tilde{c}) - \min(\tilde{c})}, \forall i \in \tilde{V} \)
8: for all \( i \in \tilde{V} \) do
9: if \( \tilde{c}_i < c_i \) then
10: Remove node \( i \) from \( G \) with probability:
11: \( \Pr(\tilde{V} = \tilde{V} \setminus \{i\}) = 1 - \tilde{c}_i^{\text{norm}} \) (also \( R = R \cup \{i\} \))
12: end if
13: end for
14: until No more nodes are removed
15: return Set of affected (removed) nodes \( R \)
Social Vulnerability Assessment

Our approach

- **Vulnerability assessment** concept combining
  
  1. The CasD model (how the disengagement epidemic is spreading)
  
  2. Observation: skewness of the core number distribution
Observations on Real Graphs

Core number distribution

- Heavy-tailed core size distribution
- Most of the nodes typically have low core number → low engagement
- Randomly selected node → more probable to have low core number
Vulnerability Assessment under Node Departures

Experimental set-up

Two strategies for selecting the node that will depart first:

1. **Random departure**: a randomly selected node leaves the graph
   - What is more probable to occur

2. **Targeted departure**: a node selected among the ones with the highest core number decides to depart
   - Highly engaged individuals disengage
Cumulative fraction of removed nodes during the execution of the model for **random** and **targeted** departures
CasD Model: Application on Real Graphs

Cumulative fraction of removed nodes during the execution of the model for random and targeted departures

- **Robustness** against cascades triggered by random departures of nodes
- **Vulnerability** under cascades triggered by targeted departures of high core (engaged) nodes
Discussion

- New concept of **vulnerability assessment** in social networks
  - User departures instead of failures and attacks
  - Engagement-based instead of degree-based

- $k$-core decomposition-based model

- Departure of highly engaged nodes can trigger big cascade
Identification of Influential Spreaders

Motivation

- **Spreading processes** in complex networks
  - Spread of news and ideas
  - Diffusion of influence
  - Disease propagation
  - Viral marketing (word-of-mouth effect)
  - ...

- Identification of **influential spreaders** (goal of this work)
  - Able to diffuse information to a large part of the network
  - Understand and control spreading dynamics
    - E.g, vaccinate individuals with good spreading properties in epidemic control
Identification of Influential Spreaders

Motivation

- **Spreading processes** in complex networks
  - Spread of news and ideas
  - Diffusion of influence
  - Disease propagation
  - Viral marketing (word-of-mouth effect)
  - ...

- Identification of **influential spreaders** *(goal of this work)*
  - Able to diffuse information to a large part of the network
  - Understand and control spreading dynamics

  - E.g, vaccinate individuals with good spreading properties in epidemic control
Identification of Influential Spreaders: the Process

Typically, a two-step approach:

1. Consider a **topological** or **centrality** criterion of the nodes
   - Rank the nodes accordingly
   - The top-ranked nodes are candidates for the most influential ones

2. Simulate the spreading process over the network to examine the performance of the chosen nodes

[Pei and Makse, ’13]
Identification of Single Influential Spreaders

Related work

- **Straightforward approach**: consider **degree centrality**
  - High degree nodes are expected to be good spreaders
  - Hub nodes can trigger big cascades
Identification of Single Influential Spreaders

Related work

- **Straightforward approach**: consider *degree centrality*
  - High degree nodes are expected to be good spreaders
  - Hub nodes can trigger big cascades

- However, degree is a *local criterion*
  - Bad case: star subgraph
**Identification of Single Influential Spreaders**

**Related work**

- **Straightforward approach**: consider **degree centrality**
  - High degree nodes are expected to be good spreaders
  - Hub nodes can trigger big cascades

- However, degree is a **local criterion**
  - Bad case: star subgraph

- **Core-periphery structure** of real-world networks

Equal degree (8)
The $k$-core Decomposition Finds Good Spreaders

- Node A: Degree 96, Core number 63
- Node B: Degree 96, Core number 26

The core number is a better spreading predictor compared to the degree.

[Kitsak et al., Nature Phys. '10]
The $k$-core Decomposition Finds Good Spreaders

- The core number is better spreading predictor compared to the degree
- [Kitsak et al., Nature Phys. ’10]
Motivation and Contributions

- The $k$-core decomposition often returns a relatively large number of candidate influential spreaders
  - Only a small fraction corresponds to highly influential nodes
- Can we further **refine** the set of the most influential spreaders?
Motivation and Contributions

- The $k$-core decomposition often returns a relatively large number of candidate influential spreaders
  - Only a small fraction corresponds to highly influential nodes
- Can we further refine the set of the most influential spreaders?

Main contributions

- Propose the $K$-truss decomposition for locating influential nodes
  - Triangle-based extension of the $k$-core decomposition
- Experimental evaluation
  - Better spreading behavior
  - Faster and wider epidemic spreading
**K-truss Decomposition**

**Definitions**

**K-truss subgraph** $T_K$ [Cohen, ’08], [Wang and Cheng, ’12]

$K$-truss $T_K = (V_{T_K}, E_{T_K}), K \geq 2$: the largest subgraph of $G$ where every edge is contained in at least $K - 2$ triangles within the subgraph

Maximal $K$-truss subgraph

- The $K$-truss subgraph defined for the maximum value $K_{max}$ of $K$
- The nodes of this subgraph define set $\mathcal{T}$
### $K$-truss Decomposition

#### Definitions

<table>
<thead>
<tr>
<th>$K$-truss subgraph $T_K$</th>
<th>[Cohen, '08], [Wang and Cheng, '12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$-truss $T_K = (V_{T_K}, E_{T_K}), K \geq 2$: the largest subgraph of $G$ where every edge is contained in at least $K - 2$ triangles within the subgraph</td>
<td></td>
</tr>
</tbody>
</table>

#### Maximal $K$-truss subgraph

- The $K$-truss subgraph defined for the maximum value $K_{\text{max}}$ of $K$
- The nodes of this subgraph define set $\mathcal{T}$
The maximal $k$-core and $K$-truss subgraphs overlap

- $K$-truss is a subgraph of $k$-core (core of the $k$-core)
- Heuristic to improve execution time

We argue that set $T$ contains highly influential nodes
**K-truss Decomposition**

Maximal $k$-core and $K$-truss subgraphs

- Set $C$: nodes of maximal $k$-core
  - 3-core subgraph
- Set $T$: nodes of maximal $K$-truss
  - 4-truss subgraph

- The maximal $k$-core and $K$-truss subgraphs overlap
  - $K$-truss is a subgraph of $k$-core (core of the $k$-core)
  - Heuristic to improve execution time

- We argue that set $T$ contains highly influential nodes
Experimental Set-up

Baseline methods

- **truss** method: nodes belonging to set $\mathcal{T}$ (Proposed method)
- **core** method: nodes belonging to set $\mathcal{C} - \mathcal{T}$
- **top degree** method: nodes with highest degree $\mathcal{T}$
  - Choose $|\mathcal{C}| - |\mathcal{T}|$ nodes for fair comparison
Experimental Set-up
How to simulate the spreading process?

**Susceptible-Infected-Recovered (SIR) model**

1. Set candidate node as infected (I state)
2. An infected node can infect its susceptible neighbors with probability $\beta$
   - Set $\beta$ close to the epidemic threshold $\tau = \frac{1}{\lambda_1}$ [Chakrabarty et al., ’08]
3. An infected node can recover (stop being active) with probability $\gamma$
   - Set $\gamma = 0.8$
4. Count the total number of infected individuals (avg. over multiple runs)

State diagram of the SIR model

[Barrat et al., ’08]
Datasets and Properties
Characteristics of the $K$-truss subgraphs

| Network Name       | Nodes    | Edges        | $k_{max}$ | $K_{max}$ | $|C|$ | $|T|$ | $\tau$    |
|--------------------|----------|--------------|-----------|-----------|------|------|-----------|
| EMAIL-ENRON        | 33,696   | 180,811      | 43        | 22        | 275  | 45   | 0.00840   |
| EPINIONS           | 75,877   | 405,739      | 67        | 33        | 486  | 61   | 0.00540   |
| WIKI-VOTE          | 7,066    | 100,736      | 53        | 23        | 336  | 50   | 0.00720   |
| EMAIL-EuAll        | 224,832  | 340,795      | 37        | 20        | 292  | 62   | 0.00970   |
| SLASHDOT           | 82,168   | 582,533      | 55        | 36        | 134  | 96   | 0.00074   |
| WIKI-TALK          | 2,388,953| 4,656,682    | 131       | 53        | 700  | 237  | 0.00870   |

- $\tau = 1/\lambda_1$: epidemic threshold of the graph ($\lambda_1$: largest eigenvalue of $A$)

- Set $T$ has significantly smaller size compared to set $C$
### Datasets and Properties

**Characteristics of the $K$-truss subgraphs**

| Network Name     | Nodes     | Edges    | $k_{max}$ | $K_{max}$ | $|C|$ | $|T|$ | $\tau$  |
|------------------|-----------|----------|-----------|-----------|------|------|--------|
| EMAIL-ENRON      | 33,696    | 180,811  | 43        | 22        | 275  | 45   | 0.00840|
| EPINIONS         | 75,877    | 405,739  | 67        | 33        | 486  | 61   | 0.00540|
| WIKI-VOTE        | 7,066     | 100,736  | 53        | 23        | 336  | 50   | 0.00720|
| EMAIL-EuAll      | 224,832   | 340,795  | 37        | 20        | 292  | 62   | 0.00970|
| SLASHDOT         | 82,168    | 582,533  | 55        | 36        | 134  | 96   | 0.00074|
| WIKI-TALK        | 2,388,953 | 4,656,682| 131       | 53        | 700  | 237  | 0.00870|

- $\tau = 1/\lambda_1$: epidemic threshold of the graph ($\lambda_1$: largest eigenvalue of $A$)

- Set $\mathcal{T}$ has significantly smaller size compared to set $\mathcal{C}$
Evaluation the Spreading Process

Apply SIR simulation starting from a single node \( v \) each time

- Number of infected nodes at each time step of the process
- Total number of infected nodes \( M_v \)
- The time step where the epidemic fades out
### Average Number of Infected Nodes

<table>
<thead>
<tr>
<th>Method</th>
<th>Time Step</th>
<th>Final step</th>
<th>( \sigma )</th>
<th>Max step</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMAIL-ENRON</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>truss</td>
<td>8.44</td>
<td>46.66</td>
<td>204.08</td>
<td>418.77</td>
</tr>
<tr>
<td>core</td>
<td>4.78</td>
<td>31.97</td>
<td>152.55</td>
<td>367.28</td>
</tr>
<tr>
<td>top degree</td>
<td>6.89</td>
<td>34.13</td>
<td>155.48</td>
<td>360.89</td>
</tr>
<tr>
<td><strong>EPINIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>truss</td>
<td>4.17</td>
<td>19.70</td>
<td>75.04</td>
<td>204.14</td>
</tr>
<tr>
<td>core</td>
<td>3.45</td>
<td>14.72</td>
<td>55.27</td>
<td>158.56</td>
</tr>
<tr>
<td>top degree</td>
<td>4.22</td>
<td>16.03</td>
<td>58.84</td>
<td>166.23</td>
</tr>
<tr>
<td><strong>Wiki-Vote</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>truss</td>
<td>2.92</td>
<td>6.92</td>
<td>15.27</td>
<td>28.73</td>
</tr>
<tr>
<td>core</td>
<td>1.92</td>
<td>4.78</td>
<td>10.65</td>
<td>20.66</td>
</tr>
<tr>
<td>top degree</td>
<td>2.43</td>
<td>5.46</td>
<td>12.05</td>
<td>23.05</td>
</tr>
<tr>
<td><strong>EMAIL-EUALL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>truss</td>
<td>11.62</td>
<td>62.25</td>
<td>240.97</td>
<td>584.87</td>
</tr>
<tr>
<td>core</td>
<td>9.85</td>
<td>40.82</td>
<td>158.72</td>
<td>433.81</td>
</tr>
<tr>
<td>top degree</td>
<td>17.96</td>
<td>39.93</td>
<td>144.69</td>
<td>503.18</td>
</tr>
</tbody>
</table>

- **Performance of truss method**
  - Higher infection rate during the first steps of the process
  - The total number of infected nodes is larger
  - The epidemic dies out earlier
### Average Number of Infected Nodes

<table>
<thead>
<tr>
<th>Method</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>Final step</th>
<th>(\sigma)</th>
<th>Max step</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMAIL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- truss</td>
<td>8.44</td>
<td>46.66</td>
<td>204.08</td>
<td>418.77</td>
<td>355.84</td>
<td>2,596.52</td>
<td>136.7</td>
<td>33</td>
</tr>
<tr>
<td>- core</td>
<td>4.78</td>
<td>31.97</td>
<td>152.55</td>
<td>367.28</td>
<td>364.13</td>
<td>2,465.60</td>
<td>199.6</td>
<td>37</td>
</tr>
<tr>
<td>- top degree</td>
<td>6.89</td>
<td>34.13</td>
<td>155.48</td>
<td>360.89</td>
<td>357.08</td>
<td>2,471.67</td>
<td>354.8</td>
<td>36</td>
</tr>
<tr>
<td><strong>EPINIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- truss</td>
<td>4.17</td>
<td>19.70</td>
<td>75.04</td>
<td>204.14</td>
<td>329.08</td>
<td>2,567.69</td>
<td>227.8</td>
<td>37</td>
</tr>
<tr>
<td>- core</td>
<td>3.45</td>
<td>14.72</td>
<td>55.27</td>
<td>158.56</td>
<td>280.03</td>
<td>2,325.37</td>
<td>327.2</td>
<td>43</td>
</tr>
<tr>
<td>- top degree</td>
<td>4.22</td>
<td>16.03</td>
<td>58.84</td>
<td>166.23</td>
<td>289.49</td>
<td>2,414.99</td>
<td>331.7</td>
<td>47</td>
</tr>
<tr>
<td><strong>Wiki-Vote</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- truss</td>
<td>2.92</td>
<td>6.92</td>
<td>15.27</td>
<td>28.73</td>
<td>42.46</td>
<td>560.66</td>
<td>114.9</td>
<td>52</td>
</tr>
<tr>
<td>- core</td>
<td>1.92</td>
<td>4.78</td>
<td>10.65</td>
<td>20.66</td>
<td>32.40</td>
<td>466.01</td>
<td>104.5</td>
<td>57</td>
</tr>
<tr>
<td>- top degree</td>
<td>2.43</td>
<td>5.46</td>
<td>12.05</td>
<td>23.05</td>
<td>35.55</td>
<td>502.88</td>
<td>104.5</td>
<td>62</td>
</tr>
<tr>
<td><strong>EMAIL-EuALL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- truss</td>
<td>11.62</td>
<td>62.25</td>
<td>240.97</td>
<td>584.87</td>
<td>725.42</td>
<td>5,018.52</td>
<td>487.94</td>
<td>36</td>
</tr>
<tr>
<td>- core</td>
<td>9.85</td>
<td>40.82</td>
<td>158.72</td>
<td>433.81</td>
<td>644.76</td>
<td>4,579.84</td>
<td>498.71</td>
<td>38</td>
</tr>
<tr>
<td>- top degree</td>
<td>17.96</td>
<td>39.93</td>
<td>144.69</td>
<td>503.18</td>
<td>548.25</td>
<td>4,137.56</td>
<td>1,174.84</td>
<td>39</td>
</tr>
</tbody>
</table>

#### Performance of truss method

- **Higher** infection rate during the first steps of the process
- The total number of infected nodes is **larger**
- The epidemic dies out **earlier**
## Average Number of Infected Nodes

<table>
<thead>
<tr>
<th>Method</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>Final step</th>
<th>σ</th>
<th>Max step</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMAIL-ENRON</td>
<td>truss</td>
<td>8.44</td>
<td>46.66</td>
<td>204.08</td>
<td>418.77</td>
<td>355.84</td>
<td>2,596.52</td>
<td>136.7</td>
</tr>
<tr>
<td></td>
<td>core</td>
<td>4.78</td>
<td>31.97</td>
<td>152.55</td>
<td>367.28</td>
<td>364.13</td>
<td>2,465.60</td>
<td>199.6</td>
</tr>
<tr>
<td></td>
<td>top degree</td>
<td>6.89</td>
<td>34.13</td>
<td>155.48</td>
<td>360.89</td>
<td>357.08</td>
<td>2,471.67</td>
<td>354.8</td>
</tr>
<tr>
<td>EPINIONS</td>
<td>truss</td>
<td>4.17</td>
<td>19.70</td>
<td>75.04</td>
<td>204.14</td>
<td>329.08</td>
<td>2,567.69</td>
<td>227.8</td>
</tr>
<tr>
<td></td>
<td>core</td>
<td>3.45</td>
<td>14.72</td>
<td>55.27</td>
<td>158.56</td>
<td>280.03</td>
<td>2,325.37</td>
<td>327.2</td>
</tr>
<tr>
<td></td>
<td>top degree</td>
<td>4.22</td>
<td>16.03</td>
<td>58.84</td>
<td>166.23</td>
<td>289.49</td>
<td>2,414.99</td>
<td>331.7</td>
</tr>
<tr>
<td>Wiki-Vote</td>
<td>truss</td>
<td>2.92</td>
<td>6.92</td>
<td>15.27</td>
<td>28.73</td>
<td>42.46</td>
<td>560.66</td>
<td>114.9</td>
</tr>
<tr>
<td></td>
<td>core</td>
<td>1.92</td>
<td>4.78</td>
<td>10.65</td>
<td>20.66</td>
<td>32.40</td>
<td>466.01</td>
<td>104.5</td>
</tr>
<tr>
<td></td>
<td>top degree</td>
<td>2.43</td>
<td>5.46</td>
<td>12.05</td>
<td>23.05</td>
<td>35.55</td>
<td>502.88</td>
<td>104.5</td>
</tr>
<tr>
<td>EMAIL-EUALL</td>
<td>truss</td>
<td>11.62</td>
<td>62.25</td>
<td>240.97</td>
<td>584.87</td>
<td>725.42</td>
<td>5,018.52</td>
<td>487.94</td>
</tr>
<tr>
<td></td>
<td>core</td>
<td>9.85</td>
<td>40.82</td>
<td>158.72</td>
<td>433.81</td>
<td>644.76</td>
<td>4,579.84</td>
<td>498.71</td>
</tr>
<tr>
<td></td>
<td>top degree</td>
<td>17.96</td>
<td>39.93</td>
<td>144.69</td>
<td>503.18</td>
<td>548.25</td>
<td>4,137.56</td>
<td>1,174.84</td>
</tr>
</tbody>
</table>

- **Performance of truss method**
  - **Higher** infection rate during the first steps of the process
  - The total number of infected nodes is **larger**
  - The epidemic dies out **earlier**
### Average Number of Infected Nodes

<table>
<thead>
<tr>
<th>Method</th>
<th>Truss</th>
<th>Core</th>
<th>Top Degree</th>
<th>Time Step</th>
<th>Final Step</th>
<th>σ</th>
<th>Max Step</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EMAIL-ENRON</strong></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8.44</td>
<td>4.78</td>
<td>6.89</td>
<td>204.08</td>
<td>355.84</td>
<td>2,596.52</td>
<td>136.7</td>
</tr>
<tr>
<td></td>
<td>46.66</td>
<td>31.97</td>
<td>34.13</td>
<td>418.77</td>
<td>364.13</td>
<td>2,465.60</td>
<td>199.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>152.55</td>
<td>155.48</td>
<td>367.28</td>
<td>357.08</td>
<td>2,471.67</td>
<td>354.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>33</td>
</tr>
<tr>
<td><strong>EPINIONS</strong></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.17</td>
<td>3.45</td>
<td>4.22</td>
<td>75.04</td>
<td>329.08</td>
<td>2,567.69</td>
<td>227.8</td>
</tr>
<tr>
<td></td>
<td>19.70</td>
<td>14.72</td>
<td>16.03</td>
<td>204.14</td>
<td>280.03</td>
<td>2,325.37</td>
<td>327.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>55.27</td>
<td>58.84</td>
<td>158.56</td>
<td>289.49</td>
<td>2,414.99</td>
<td>331.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>37</td>
</tr>
<tr>
<td><strong>WIKI-VOTE</strong></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.92</td>
<td>1.92</td>
<td>2.43</td>
<td>15.27</td>
<td>42.46</td>
<td>560.66</td>
<td>114.9</td>
</tr>
<tr>
<td></td>
<td>6.92</td>
<td>4.78</td>
<td>5.46</td>
<td>28.73</td>
<td>32.40</td>
<td>466.01</td>
<td>104.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.65</td>
<td>12.05</td>
<td>23.05</td>
<td>35.55</td>
<td>502.88</td>
<td>104.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>52</td>
</tr>
<tr>
<td><strong>EMAIL-EUALL</strong></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11.62</td>
<td>9.85</td>
<td>17.96</td>
<td>240.97</td>
<td>725.42</td>
<td>5,018.52</td>
<td>487.94</td>
</tr>
<tr>
<td></td>
<td>62.25</td>
<td>40.82</td>
<td>39.93</td>
<td>584.87</td>
<td>644.76</td>
<td>4,579.84</td>
<td>498.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>158.72</td>
<td>144.69</td>
<td>532.87</td>
<td>548.25</td>
<td>4,137.56</td>
<td>1,174.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>36</td>
</tr>
</tbody>
</table>

- **Performance of truss method**
  - **Higher** infection rate during the first steps of the process
  - The total number of infected nodes is **larger**
  - The epidemic dies out **earlier**
Comparison to the Optimal Spreading

Methodology

1. Rank nodes according to the total infection size $M_v$
   - $OPT_1 \geq OPT_2 \geq \ldots \geq OPT_{|V|}$, where $OPT_1 = \arg \max_{v \in V} M_v$

2. Consider window $W$ over the ranked nodes
   $$P_W^T = \frac{|T_W|}{|W|} \div \frac{|T|}{|V|}$$
   - $T_W$ is the set of nodes $v \in T$ located in $W$
Comparison to the Optimal Spreading (1/2)

Methodology

1. Rank nodes according to the total infection size $M_v$
   - $OPT_1 \geq OPT_2 \geq \ldots \geq OPT_{|V|}$, where $OPT_1 = \arg \max_{v \in V} M_v$

2. Consider window $W$ over the ranked nodes

$$P_W^T = \frac{|T_W|/|T|}{|W|/|V|}$$

- $T_W$ is the set of nodes $v \in T$ located in $W$
Comparison to the Optimal Spreading (2/2)

**Results**

**EMAIL-ENRON**

![Graph showing comparison between optimal spreading and other methods for EMAIL-ENRON network](image)

**WIKI-VOTE**

![Graph showing comparison between optimal spreading and other methods for WIKI-VOTE network](image)

- The nodes detected by the K-truss decomposition are better distributed among the most efficient spreaders.
- The spread reaches the maximum value (i.e., 100%) relatively early and for small window sizes, compared to other methods.
Comparison to the Optimal Spreading (2/2)

Results

**EMAIL-ENRON**

![Graph showing comparison between two metrics](image1)

- $P_W^T$ reaches the maximum value (i.e., 100%) relatively early and for small window sizes, compared to $P_W^C$.

- The nodes detected by the $K$-truss decomposition are better distributed among the most efficient spreaders.

**WIKI-VOTE**

![Graph showing comparison between two metrics](image2)
Discussion

- The $K$-truss decomposition can help towards identifying single influential spreaders
  - Faster and wider epidemic spreading
  - Well distributed nodes among those that are achieving the optimal spreading
Concluding Remarks
Degeneracy-based Graph Mining

Graph mining and core decomposition

1. Models, dynamics and properties of social networks

2. Algorithmic tools for graph analytics
Summary of Contributions

**C1:** $k$-core decomposition for modeling the engagement dynamics

**C2:** Model for vulnerability assessment under node departures in social networks

**C3:** The $K$-truss decomposition method locates highly influential nodes

**C4:** Accelerating graph clustering and community detection with the $k$-core decomposition

**C5:** Graph mining and core decomposition for text analytics
Summary of Contributions

**C1:** $k$-core decomposition for modeling the engagement dynamics

**C2:** Model for vulnerability assessment under node departures in social networks

**C3:** The $K$-truss decomposition method locates highly influential nodes

**C4:** Accelerating graph clustering and community detection with the $k$-core decomposition

**C5:** Graph mining and core decomposition for text analytics
Summary of Contributions

C1: $k$-core decomposition for modeling the engagement dynamics

C2: Model for vulnerability assessment under node departures in social networks

C3: The $K$-truss decomposition method locates highly influential nodes

C4: Accelerating graph clustering and community detection with the $k$-core decomposition

C5: Graph mining and core decomposition for text analytics
Summary of Contributions

C1: $k$-core decomposition for modeling the engagement dynamics

C2: Model for vulnerability assessment under node departures in social networks

C3: The $K$-truss decomposition method locates highly influential nodes

C4: Accelerating graph clustering and community detection with the $k$-core decomposition

C5: Graph mining and core decomposition for text analytics
Future Work

- Engagement dynamics on graphs with rich semantics
  - What about ground truth information?
- Prediction algorithms for network vulnerability
- Identification of multiple influential spreaders
Future Work

- Engagement dynamics on graphs with rich semantics
  - What about ground truth information?
- Prediction algorithms for network vulnerability
- Identification of multiple influential spreaders
Future Work

- Engagement dynamics on graphs with rich semantics
  - What about ground truth information?
- Prediction algorithms for network vulnerability
- Identification of multiple influential spreaders
Publications I

F. D. Malliaros and M. Vazirgiannis
To Stay or Not to Stay: Modeling Engagement Dynamics in Social Graphs.
In ACM International Conference on Information and Knowledge Management (CIKM), 2013.

C. Giatsidis, F. D. Malliaros, D. M. Thilikos, and M. Vazirgiannis
CoreCluster: A Degeneracy Based Graph Clustering Framework.
In AAAI Conference on Artificial Intelligence, (AAAI), 2014.

M.-E. G. Rossi, F. D. Malliaros, and M. Vazirgiannis
Spread It Good, Spread It Fast: Identification of Influential Nodes in Social Networks.
In International Conference on World Wide Web Companion (WWW), 2015.

F. D. Malliaros and M. Vazirgiannis
Vulnerability Assessment in Social Networks under Cascade-based Node Departures.
EPL (Europhysics Letters), 11(6), 2015.

F. D. Malliaros and K. Skianis
Graph-Based Term Weighting for Text Categorization.
In IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Social Media and Risk Workshop (SoMeRis), 2015.

F. D. Malliaros, M. Rossi, and M. Vazirgiannis
Locating Influential Nodes in Complex Networks.
Scientific Reports, 6(19307), Nature Publishing Group, 2016.
References I

R. Albert, H. Jeong, and A. L. Barabási
Error and Attack Tolerance of Complex Networks.

R. Albert and A. L. Barabási
Statistical Mechanics of Complex Networks.

V. Batagelj and M. Zaversnik.
An $O(m)$ Algorithm for Cores Decomposition of Networks.

A. Barrat, M. Barthlémy, and A. Vespignani.
Dynamical Processes on Complex Networks.

K. Bhawalkar, J. Kleinberg, K. Lewi, T. Roughgarden, and A. Sharma.
Preventing Unraveling in Social Networks: the Anchored $k$-core Problem.
In *ICALP*, 2011.

R. Blanco and C. Lioma
Graph-Based Term Weighting for Information Retrieval.

Network Robustness and Fragility: Percolation on Random Graphs.
D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and Christos Faloutsos. 
Epidemic Thresholds in Real Networks. 

R. Cohen, K. Erez, D. ben Avraham, and S. Havlin 
Breakdown of the Internet under Intentional Attack. 

J. Cohen 
Trusses: Cohesive Subgraphs for Social Network Analysis. 

D. Easley and J. Kleinberg 
Networks, Crowds, and Markets: Reasoning About a Highly Connected World. 

M. Faloutsos, P. Faloutsos, and C. Faloutsos 
In SIGCOMM, 1999.

A. Harkins 
Network Games with Perfect Complements. 

T. Joachims 
Text Categorization with Support Vector Machines: Learning with Many Relevant Features. 
In ECML, 1998.

D. Kempe, J. Kleinberg, and E. Tardos 
Maximizing the Spread of Influence Through a Social Network. 
In KDD, 2003.
References III

M. Kitsak, L. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. Stanley, and H. Makse
Identification of Influential Spreaders in Complex Networks.
*Nature Physics, 6*(11), 2010.

C. D. Manning, P. Raghavan, and H. Schütze.
Introduction to Information Retrieval.

V. H. Manshadi and R. Johari
Supermodular Network Games.

A. E. Motter and Y.-C. Lai
Cascade-based Attacks on Complex Networks.

S. Pei and H. A. Makse
Spreading Dynamics in Complex Networks.

F. Rousseau and M. Vazirgiannis
Graph-of-Word and TW-IDF: New Approach to Ad Hoc IR.
In *CIKM*, 2013.

F. Rousseau and M. Vazirgiannis
Main Core Retention on Graph-of-Words for Single-Document Keyword Extraction.
In *ECIR*, 2015.

F. Sebastiani
Machine Learning in Automated Text Categorization.
References IV

S. B. Seidman
Network Structure and Minimum Degree.
*Social Networks* 5, 1983.

J. Wang and J. Cheng
Truss Decomposition in Massive Networks.
*Proc. VLDB Endow.*, 5(9), 2012.

S. Wu, A. Das Sarma, A. Fabrikant, S. Lattanzi, and A. Tomkins
Arrival and Departure Dynamics in Social Networks.
In *WSDM*, 2013.
Thank You!

NETWORK AND DATA SCIENCE
Graph Mining
Social Networks Analysis
Spectral Graph Theory
Core Decomposition
Applied Machine Learning
Natural Language Processing

Graph-based text analytics
Social vulnerability assessment
Engagement dynamics
Influential spreaders
Privacy in real networks
Community detection
Spectral robustness estimation
Degeneracy-Based Mining of Social and Information Networks