OLAK: An Efficient Algorithm to Prevent Unraveling in Social Networks

Fan Zhang¹, Wenjie Zhang², Ying Zhang¹, Lu Qin¹, Xuemin Lin²

¹ University of Technology Sydney, ² University of New South Wales
Network Unraveling


The engagement of a user is influenced by the number of her friends.
Network Unraveling

An equilibrium: a group has the minimum degree of $k$

$k = 3$

Network Unraveling

An equilibrium: a group has the minimum degree of $k$.

$k = 3$

Network Unraveling

A social group tends to be a $k$-core in the network.

$\kappa = 3$

**k-Core**

- Given a graph $G$, the $k$-core of $G$ is a maximal subgraph where each node has at least $k$ neighbors (i.e., $k$ adjacent nodes, or a degree of $k$).


Applications: community detection, social contagion, user engagement, event detection, ……
$k$-Core Decomposition

- **Core number** of a node $v$: the largest value of $k$ such that there is a $k$-core containing $v$.
- Core decomposition: compute the **core number** of each node in $G$.

The Collapse of Friendster

• Founded in 2002.
• Popular at early 21st century, over 115 million users in 2011.
• Suspended in 2015 for lack of engagement by the online community.


The core number threshold steadily increased.
The Collapse of Friendster

- Founded in 2002.
- Popular at early 21st century, over 115 million users in 2011.
- Suspended in 2015 for lack of engagement by the online community.


The collapse started from the center of the core.
User Engagement

• Founded in 2002.
• Popular at early 21st century, over 115 million users in 2011.
• Suspended in 2015 for lack of engagement by the online community.


Social influence is tightly controlled by the number of friends in current subgraph, like $k$-core.
User Engagement

- Founded in 2002.
- Popular at early 21st century, over 115 million users in 2011.
- Suspended in 2015 for lack of engagement by the online community.


The degeneration property of k-core can be used to quantify engagement dynamics.
Prevent Network Unraveling

$u_1$ is called an anchor.

Prevent Network Unraveling

Anchor: if a node $u$ is an anchor, $u$ will never leave the $k$-core community (i.e., the degree of $u$ is always $+\infty$).

Anchored $k$-Core: the $k$-core with some anchors.
Prevent Network Unraveling

**Follower:** a node $v$ is a follower of an anchor $u$, if $v$ is not in $k$-core but belongs to anchored $k$-core by anchoring $u$.

**Anchored $k$-Core Problem:** Given two integers $k$ and $b$, find $b$ anchors to maximize the number of followers (i.e., maximize the number of nodes in anchored $k$-core).

**NP-Hard**

When $k = 3$ and $b = 1$, $u_1$ is a best anchor with 3 followers for the anchored $k$-core problem.

Theorems for Anchoring One Node

**k-Shell:** the nodes in k-core but not in (k+1)-core.

- **Theorem 1:** if \( v \) is a follower of \( u \), \( v \) belongs to \((k-1)\)-shell.

- **Theorem 2:** if \( u \) has at least 1 follower, \( u \) belongs to \((k-1)\)-shell or \( u \) is a neighbor of a node in \((k-1)\)-shell.

\( k = 3 \)
**OLAK Algorithm for Anchored \( k \)-Core Problem**

**A greedy algorithm:** Computing anchored \( k \)-core for every candidate anchor node to find a best anchor (the one with most followers) in each iteration.

**Onion Layers:** a structure based on \((k-1)\)-shell and the neighbors of \((k-1)\)-shell nodes according to deletion order of these nodes in \( k \)-core computation.

We only need to explore a small portion of the Onion Layers to find all followers for an anchor.
Onion Layers in OLAK Algorithm

\( k = 3 \) in the following example

\( C_k(G) \) is the \( k \)-core of \( G \),
\( \text{deg}(u, N) \) is the degree of \( u \) in \( N \),
\( NB(L, G) \) is the neighbor set of \( L \) in \( G \)

**Algorithm : OnionPeeling**(\( G, k \))

1 \( N := C_{k-1}(G); i := 0; \)
2 \( P := \{ u \mid \text{deg}(u, N) < k \land u \in N \}; \)
3 \( \text{while } P \neq \emptyset \text{ do} \)
4 \( \quad i := i + 1; L_i := P; \)
5 \( \quad N := N \setminus P; \)
6 \( \quad P := \{ u \mid \text{deg}(u, N) < k \land u \in N \}; \)
7 \( L_0 := \{ u \mid u \in NB(L_1, G) \setminus \{ N \cup L_1 \} \}; \)
8 \( \text{return } L_0 \)
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$, $\text{deg}(u, N)$ is the degree of $u$ in $N$, $\text{NB}(L, G)$ is the neighbor set of $L$ in $G$
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

core: $C_k(G)$ is the $k$-core of $G$,
degree: $\text{deg}(u, N)$ is the degree of $u$ in $N$,
neighbor set: $NB(L, G)$ is the neighbor set of $L$ in $G$

Algorithm: OnionPeeling($G, k$)

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$N := C_{k-1}(G); i := 0;$</td>
</tr>
<tr>
<td>2</td>
<td>$P := {u \mid \text{deg}(u, N) &lt; k &amp; u \in N};$</td>
</tr>
<tr>
<td>3</td>
<td>while $P \neq \emptyset$ do</td>
</tr>
<tr>
<td>4</td>
<td>$i := i + 1; L_i := P;$</td>
</tr>
<tr>
<td>5</td>
<td>$N := N \setminus P;$</td>
</tr>
<tr>
<td>6</td>
<td>$P := {u \mid \text{deg}(u, N) &lt; k &amp; u \in N};$</td>
</tr>
<tr>
<td>7</td>
<td>$L_0 := {u \mid u \in NB(L_1^i, G) \setminus {N \cup L_1^i}};$</td>
</tr>
<tr>
<td>8</td>
<td>return $L_0^i$</td>
</tr>
</tbody>
</table>

Onion Layer Structure ($L_0^3$)
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$, $deg(u, N)$ is the degree of $u$ in $N$, $NB(L, G)$ is the neighbor set of $L$ in $G$

---

Algorithm: OnionPeeling($G$, $k$)

Input: $G$: a social network, $k$: degree constraint
Output: onion layers $\mathcal{L}$ (i.e., $L_0^k$)

1. $N := C_{k-1}(G)$; $i := 0$;
2. $P := \{u \mid deg(u, N) < k \& u \in N\}$;
3. while $P \neq \emptyset$ do
   4. $i := i + 1$; $L_i := P$;
   5. $N := N \setminus P$;
   6. $P := \{u \mid deg(u, N) < k \& u \in N\}$;
7. $L_0 := \{u \mid u \in NB(L_1^i, G) \setminus \{N \cup L_1^i\}\}$;
8. return $L_0^i$
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$, $\text{deg}(u,N)$ is the degree of $u$ in $N$, $\text{NB}(L,G)$ is the neighbor set of $L$ in $G$

**Algorithm : OnionPeeling($G$, $k$)**

**Input**: $G$: a social network, $k$: degree constraint  
**Output**: onion layers $\mathcal{L}$ (i.e., $L_k^0$)

1. $N := C_{k-1}(G); i := 0$;
2. $P := \{u \mid \text{deg}(u,N) < k \land u \in N\}$;
3. while $P \neq \emptyset$ do
   4. $i := i + 1; L_i := P$;
   5. $N := N \setminus P$
   6. $P := \{u \mid \text{deg}(u,N) < k \land u \in N\}$;
7. $L_0 := \{u \mid u \in \text{NB}(L_i^1,G) \setminus \{N \cup L_i^1\}\}$;
8. return $L_0$
Onion Layers in OLAK Algorithm

\( k = 3 \) in the following example

\( C_k(G) \) is the \( k \)-core of \( G \),
\( \text{deg}(u, N) \) is the degree of \( u \) in \( N \),
\( NB(L, G) \) is the neighbor set of \( L \) in \( G \)

---

Algorithm: OnionPeeling\((G, k)\)

- **Input**: \( G \) : a social network, \( k \) : degree constraint
- **Output**: onion layers \( \mathcal{L} \) (i.e., \( L_0^k \))

1. \( N := C_{k-1}(G); i := 0 \);
2. \( P := \{ u | \text{deg}(u, N) < k \land u \in N \} \);
3. while \( P \neq \emptyset \) do
   4. \( i := i + 1; L_i := P \);
   5. \( N := N \setminus P \);
   6. \( P := \{ u | \text{deg}(u, N) < k \land u \in N \} \);
7. \( L_0 := \{ u | u \in NB(L_1^i, G) \setminus \{ N \cup L_1^i \} \} \);
8. return \( L_0^i \)
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$, $\text{deg}(u, N)$ is the degree of $u$ in $N$, $\text{NB}(L, G)$ is the neighbor set of $L$ in $G$

```
Algorithm : OnionPeeling($G$, $k$)

Input : $G$ : a social network, $k$ : degree constraint
Output : onion layers $\mathcal{L}$ (i.e., $L_0^k$)

1. $N := C_{k-1}(G); i := 0$
2. $P := \{u \mid \text{deg}(u, N) < k \& u \in N\}$
3. while $P \neq \emptyset$ do
   4. $i := i + 1; L_i := P$
   5. $N := N \setminus P$
   6. $P := \{u \mid \text{deg}(u, N) < k \& u \in N\}$
7. $L_0 := \{u \mid u \in \text{NB}(L_1^i, G) \setminus \{N \cup L_1^i\}\}$
8. return $L_0$
```
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$,
$\text{deg}(u, N)$ is the degree of $u$ in $N$,
$\text{NB}(L, G)$ is the neighbor set of $L$ in $G$

Algorithm: OnionPeeling($G, k$)

```
Input : $G$: a social network, $k$: degree constraint
Output : onion layers $\mathcal{L}$ (i.e., $L^8$)
1 $N := C_{k-1}(G); i := 0$
2 $P := \{ u \mid \text{deg}(u, N) < k \& u \in N \}$
3 while $P \neq \emptyset$ do
4     $i := i + 1; L_i := P$
5     $N := N \setminus P$
6     $P := \{ u \mid \text{deg}(u, N) < k \& u \in N \}$
7 $L_0 := \{ u \mid u \in \text{NB}(L_1, G) \setminus \{ N \cup L_1 \} \}$
8 return $L_0$
```
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$,
$\text{deg}(u,N)$ is the degree of $u$ in $N$,
$NB(L,G)$ is the neighbor set of $L$ in $G$

Algorithm: OnionPeeling($G$, $k$)

1. $N := C_{k-1}(G)$; $i := 0$
2. $P := \{u \mid \text{deg}(u,N) < k & u \in N\}$
3. while $P \neq \emptyset$ do
   4. $i := i + 1$; $L_i := P$
   5. $N := N \setminus P$
   6. $P := \{u \mid \text{deg}(u,N) < k & u \in N\}$
7. $L_0 := \{u \mid u \in NB(L_1^i, G) \setminus \{N \cup L_1^i\}\}$
8. return $L_0^i$
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$,
$\text{deg}(u, N)$ is the degree of $u$ in $N$,
$\text{NB}(L, G)$ is the neighbor set of $L$ in $G$

Algorithm : OnionPeeling($G, k$)

<table>
<thead>
<tr>
<th>Input</th>
<th>$G$: a social network, $k$: degree constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>onion layers $L$ (i.e., $L^S_0$)</td>
</tr>
<tr>
<td>1</td>
<td>$N := C_{k-1}(G); i := 0;$</td>
</tr>
<tr>
<td>2</td>
<td>$P := {u \mid \text{deg}(u, N) &lt; k &amp; u \in N};$</td>
</tr>
<tr>
<td>3</td>
<td>while $P \neq \emptyset$ do</td>
</tr>
<tr>
<td>4</td>
<td>$i := i + 1; L_i := P;$</td>
</tr>
<tr>
<td>5</td>
<td>$N := N \setminus P;$</td>
</tr>
<tr>
<td>6</td>
<td>$P := {u \mid \text{deg}(u, N) &lt; k &amp; u \in N};$</td>
</tr>
<tr>
<td>7</td>
<td>$L_0 := {u \mid u \in \text{NB}(L_1^i, G) \setminus {N \cup L_1^i}};$</td>
</tr>
<tr>
<td>8</td>
<td>return $L_0^i$</td>
</tr>
</tbody>
</table>

Onion Layer Structure ($L^3_0$)
Onion Layers in OLAK Algorithm

$k = 3$ in the following example

$C_k(G)$ is the $k$-core of $G$, $\text{deg}(u, N)$ is the degree of $u$ in $N$, $\text{NB}(L, G)$ is the neighbor set of $L$ in $G$.

Algorithm: OnionPeeling($G, k$)

Input: $G$: a social network, $k$: degree constraint
Output: onion layers $\mathcal{L}$ (i.e., $L_0^k$)

1. $N := C_{k-1}(G)$; $i := 0$
2. $P := \{u | \text{deg}(u, N) < k \land u \in N\}$;
3. while $P \neq \emptyset$ do
   4. $i := i + 1$; $L_i := P$
   5. $N := N \setminus P$
   6. $P := \{u | \text{deg}(u, N) < k \land u \in N\}$
7. $L_0 := \{u \mid u \in \text{NB}(L_i, G) \setminus \{N \cup L_i\}\}$
8. return $L_0^i$

After OnionPeeling algorithm, $N$ is the $k$-core of $G$. 
Theorems for Anchored k-Core

**Support Path:** there is a support path from \( u \) to \( v \) if \( u \) can downward spread to \( v \) in Onion Layers through neighboring edges. Horizontal or upward spreads are NOT allowed.

**Theorem 3:** if \( v \) is a follower of \( u \), there is a support path from \( u \) to \( v \).
Onion Layer Search to Find Followers

If we anchor the node $v_1$, only $v_2$ and $v_3$ become candidate followers, $v_4$ and $v_7$ cannot be followers of $v_1$.

Reason: $v_4$ and $v_7$ will still be deleted in the deletion order of producing onion layers (i.e., producing k-core), i.e., $v_4$ and $v_7$ cannot have larger degrees after anchoring $v_1$. 

$k = 3$
Theorems for Anchored k-Core

**Theorem 4:** if the degree upperbound of \( u \) is less than \( k \) in the Onion Layer Search, we can early terminate the spread on \( u \).

**Theorem 5:** if \( v \) is a follower of \( u \), \( v \) cannot have more followers than \( u \).

\( v_2 \) or \( v_3 \) cannot have more followers than \( v_1 \).

\( k = 3 \)
Follower Number Upper Bound

Let $W(x)$ denote the neighbors of a vertex $x$ in lower layers, i.e., $W(x) = \{u \mid u \in NB(x) \cap \mathcal{L} \text{ and } l(u) > l(x)\}$. We use $UB(x)$ to denote the upper bound of $|\mathcal{F}(x)|$, where

$$UB(x) = \begin{cases} \sum_{u \in W(x)} (UB(u) + 1) & \text{if } |W(x)| > 0; \\ 0 & \text{otherwise}. \end{cases} \quad (1)$$

$L$ is the Onion Layers,
$l(u)$ is the layer number of $u$,
$NB(u)$ is the neighbor set of $u$,
$F(x)$ is the follower set of $x$,

$k = 3$

Theorem 6: An anchor $x$ cannot have more followers than $UB(x)$. 

Onion Layer Structure ($L_0^3$)
Follower Number Upper Bound

Let $W(x)$ denote the neighbors of a vertex $x$ in lower layers, i.e., $W(x) = \{u \mid u \in NB(x) \cap \mathcal{L} \text{ and } l(u) > l(x)\}$. We use $UB(x)$ to denote the upper bound of $|\mathcal{F}(x)|$, where

$$UB(x) = \begin{cases} \sum_{u \in W(x)} (UB(u) + 1) & \text{if } |W(x)| > 0; \\ 0 & \text{otherwise.} \end{cases}$$

$L$ is the Onion Layers,
$l(u)$ is the layer number of $u$,
$NB(u)$ is the neighbor set of $u$,
$F(x)$ is the follower set of $x$,

**Theorem 6:** An anchor $x$ cannot have more followers than $UB(x)$.  

$k = 3$
Follower Number Upper Bound

Let $W(x)$ denote the neighbors of a vertex $x$ in lower layers, i.e., $W(x) = \{ u \mid u \in NB(x) \cap L$ and $l(u) > l(x) \}$. We use $UB(x)$ to denote the upper bound of $| \mathcal{F}(x) |$, where

$$UB(x) = \begin{cases} \sum_{u \in W(x)} (UB(u) + 1) & \text{if } |W(x)| > 0; \\ 0 & \text{otherwise.} \end{cases}$$ \hspace{1cm} (1)

$L$ is the Onion Layers,
$l(u)$ is the layer number of $u$,
$NB(u)$ is the neighbor set of $u$,
$F(x)$ is the follower set of $x$,

Theorem 6: An anchor $x$ cannot have more followers than $UB(x)$. 
Follower Number Upper Bound

Let $W(x)$ denote the neighbors of a vertex $x$ in lower layers, i.e., $W(x) = \{ u \mid u \in NB(x) \cap \mathcal{L} \text{ and } l(u) > l(x) \}$. We use $UB(x)$ to denote the upper bound of $|\mathcal{F}(x)|$, where

$$UB(x) = \begin{cases} \sum_{u \in W(x)} (UB(u) + 1) & \text{if } |W(x)| > 0; \\ 0 & \text{otherwise}. \end{cases}$$  \hspace{1cm} (1)$$

$L$ is the Onion Layers, $l(u)$ is the layer number of $u$, $NB(u)$ is the neighbor set of $u$, $F(x)$ is the follower set of $x$.

Theorem 6: An anchor $x$ cannot have more followers than $UB(x)$. 

$k = 3$

Onion Layer Structure ($L_0^3$)
Follower Number Upper Bound

Let $W(x)$ denote the neighbors of a vertex $x$ in lower layers, i.e., $W(x) = \{ u \mid u \in NB(x) \cap L \text{ and } l(u) > l(x) \}$. We use $UB(x)$ to denote the upper bound of $|\mathcal{F}(x)|$, where

$$UB(x) = \begin{cases} \sum_{u \in W(x)} (UB(u) + 1) & \text{if } |W(x)| > 0; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

$L$ is the Onion Layers,

$l(u)$ is the layer number of $u$,

$NB(u)$ is the neighbor set of $u$,

$F(x)$ is the follower set of $x$,

$\mathcal{F}(x)$ is the follower set of $x$,

$k = 3$ \hspace{1cm} \text{Onion Layer Structure} (L^3_0)

**Theorem 6:** An anchor $x$ cannot have more followers than $UB(x)$. 
Experimental Setting

- **Datasets:**
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Nodes</th>
<th>Edges</th>
<th>$d_{avg}$</th>
<th>$d_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>4,039</td>
<td>88,234</td>
<td>43.7</td>
<td>1045</td>
</tr>
<tr>
<td>Brightkite</td>
<td>58,228</td>
<td>194,090</td>
<td>6.7</td>
<td>1098</td>
</tr>
<tr>
<td>Gowalla</td>
<td>196,591</td>
<td>456,830</td>
<td>4.7</td>
<td>9967</td>
</tr>
<tr>
<td>Yelp</td>
<td>552,339</td>
<td>1,781,908</td>
<td>6.5</td>
<td>3812</td>
</tr>
<tr>
<td>Flickr</td>
<td>105,938</td>
<td>2,316,948</td>
<td>43.7</td>
<td>5465</td>
</tr>
<tr>
<td>YouTube</td>
<td>1,134,890</td>
<td>2,987,624</td>
<td>5.3</td>
<td>28754</td>
</tr>
<tr>
<td>DBLP</td>
<td>1,566,919</td>
<td>6,461,300</td>
<td>8.3</td>
<td>2023</td>
</tr>
<tr>
<td>Pokec</td>
<td>1,632,803</td>
<td>8,320,605</td>
<td>10.2</td>
<td>7266</td>
</tr>
<tr>
<td>LiveJournal</td>
<td>3,997,962</td>
<td>34,681,189</td>
<td>17.4</td>
<td>14815</td>
</tr>
<tr>
<td>Orkut</td>
<td>3,072,441</td>
<td>117,185,083</td>
<td>76.3</td>
<td>33313</td>
</tr>
</tbody>
</table>

- **Environments:**
  - Intel Xeon 2.3GHz CPU and Redhat Linux System.
  - All algorithms are implemented in C++.
Case Studies

Yelp is a crowd-sourced local business review and social networking site.

DBLP is a computer science bibliography website.
Number of Followers

(a) 10 Datasets, k=20, b=20

(b) Pokec, k=20
(c) LiveJournal, k=20
(d) DBLP, b=20
(e) Gowalla, b=20
Efficiency

![Graph showing time cost (sec) for Baseline1, Baseline2, and OLAK across different datasets like Facebook, Brightkite, Gowalla, Yelp, Flickr, YouTube, DBLP, Pokec, and LiveJournal.]

- **Baseline1** △
- **Baseline2** □
- **OLAK** ○

(a) Brightkite, k=20
(b) Brightkite, b=20
(c) Orkut, k=20
(d) Orkut, b=20