HiDDen: Hierarchical Dense Subgraph Detection with Application to Financial Fraud Detection

Presented by Si Zhang (ASU)
Dense Subgraph: What?

- **Def:** A subgraph of a high density

- **Examples:**
  - Clique: each node connects to every other node in the graph
  - \( \alpha \)-Quasi-Clique: a graph that has \( n \) nodes and at least \( \alpha n(n-1)/2 \) edges
  - K-core: each node has a degree at least \( k \)
Dense Subgraph: Why?

- **Applications**

  - Spam Link Farms [Gibson’05]
  - Community Detection [Sozio’10]
  - Story Identification [Angel’13]
  - Fraud Detection [Hooi’16]
Dense Subgraph: Why?

- Synthetic Identity Detection

Account 1
Account 2
Account 3
Account 4
Account 5
Account 6
Account 7
Account 8
Account 9
Account 10

- Address 1
- Phone 1
- Holder Name 1
- Holder Name 2
- Holder Name 3
- Holder Name 4
- Address 2
- Phone 2
- Phone 3
- Email 1: fortune.777@hotmail.com
- Email 2: fortune.666@hotmail.com
- Email 3: fortune0@bellsouth.net
- Email 4: fortune10@bellsouth.net
- Email 5: pricilen10@bellsouth.net
- Email 6: nicolson@bellsouth.net
- Email 7: nicolson10@bellsouth.net
- Email 8: pricilen@bellsouth.net
Dense Subgraph: How?

- **Density Measures**
  - Edge density: \( d = \frac{2m}{n(n-1)} \)
  - Average degree: \( d = \frac{2m}{n} \)
  - Triangle density: \( d = \frac{\text{# of triangles}}{n(n-1)(n-2)/6} \)

- **Existing Methods**
  - Densest subgraph: greedy method [Charikar’00]
  - k-clique [Tsourakakis’15], k-core, k-plex
  - Denser than the densest [Tsourakakis’13]

- **Key Idea:** to flatly extract one or more partitions in a graph
Why Hierarchical Dense Subgraphs?

- A more comprehensive view of dense subgraph structures
- Example:
Challenges: Hierarchical Dense Subgraphs

- **C1. Optimization Formulation**
  - Flat detection: quadratic optimization constrained on simplex

  $\max_{x} x^T Ax$

  To maximize the number of edges in the subgraph

  \[ \sum_{i=1}^{n} x_i^\beta = 1, \quad x_i \geq 0 \]

  - **Question:** how to formulate multiple hierarchies together?

- **C2. Optimization Algorithm**
  - Flat detection: non-convex or polynomial approximation

  - **Question:** how to develop an effective and scalable algorithm?
Challenges: Hierarchical Dense Subgraphs

- **C3. Generalizations**
  - **Question**: How to generalize to bipartite graphs?

- **Question**: How to detect for a set of certain query nodes?
Outline

- Motivations
- Q1: HiDDen Formulation
- Q2: HiDDen Algorithm
- Q3: HiDDen Generalizations
- Experimental Results
- Conclusions
Prob. Def: Hierarchical Dense Subgraph Detection

- **Given:**
  - (1) adjacency matrix $A$;
  - (2) missing edge penalty $p$
  - (3) number of hierarchies $K$;
  - (4) density increase ratio $\eta$.

- **Output:** subgraph node indicator vectors $x_1, x_2, \ldots, x_K$.

- **An Illustrative Example**
Prob. Def: Query-Specific Hierarchical Dense Subgraph Detection

- **Given:**
  - (1) adjacency matrix $A$; (2) missing edge penalty $p$
  - (3) number of hierarchies $K$; (4) density increase ratio $\eta$;
  - (5) query node set $V_s$.

- **Output:** subgraph node indicator vectors $x_1, x_2, \ldots, x_K$.

- **An Illustrative Example**
HiDDen Formulation: Density Measure

- Intuition:
  - #1: Maximize the number of existing edges
  - #2: Minimize the penalty of the missing edges

- Mathematical Details:

\[
\max_{x} \quad J(x) = x^T A x - px^T (1_{n \times n} - I - A) x
\]

s.t.  
\[x \in \{0,1\}^n\]

- Correctness:
  - Equivalent to edge surplus density w.r.t quasi-clique

- Relaxation:

\[x \in \{0,1\}^n \quad \rightarrow \quad 0 \leq x \leq 1\]
HiDDen Formulation: Constraints for Hierarchies

- **Constraints:**
  - #1 – Density variety: densities in two hierarchies exhibit a difference
  - #2 – Nested node set: larger subgraphs contain smaller subgraphs

- **Mathematical Details:**
  - Density variety:
    \[
    \frac{(x^k)^T A x^k}{(x^k)^T (1_{n \times n} - I) x^k} \geq \eta \frac{(x^{k-1})^T A x^{k-1}}{(x^{k-1})^T (1_{n \times n} - I) x^{k-1}}
    \]
    Example: \(d_3 \geq 1.1 \times d_2\)
  - Nested node set:
    \[V^{k+1} \subseteq V^k \subseteq V^{k-1}\]
    Example: \(V^3 \subseteq V^2 \subseteq V^1 \subseteq V\)
HiDDen Formulation: Objective Function

- **Objective function:**

$$\max_{x_1, x_2, \ldots, x_K} \sum_{k=1}^{K} (x^k)^T [(1 + p)A - p(1_{n \times n} - I)]x^k$$

s.t.

$$\frac{(x^j)^T Ax^j}{(x^j)^T (1_{n \times n} - I)x^j} \geq \frac{(x^{j-1})^T Ax^{j-1}}{(x^{j-1})^T (1_{n \times n} - I)x^{j-1}}$$

$$x^{j+1} \leq x^j \leq x^{j-1}$$

$$\forall \ j = 1, 2, \ldots, K$$

- **Observation:** a non-convex quadratic constrained quadratic programming problem (QCQP)

- **Question:** can we simplify the problem?
HiDDen Formulation: QCQP Relaxation

- Constraint #1 Relaxation:
  - Relax it to a regularization, i.e.,
    \[
    \frac{(x^j)^T A x^j}{(x^j)^T (1_{n\times n} - I) x^j} \geq \eta \frac{(x^{j-1})^T A x^{j-1}}{(x^{j-1})^T (1_{n\times n} - I) x^{j-1}}
    \]
  
  Relax

  \[
  \max_{x^j} (x^j)^T A x^j - C^{j-1} (x^j)^T (1_{n\times n} - I) x^j
  \]

  where \( C^{j-1} = \eta \frac{(x^{j-1})^T A x^{j-1}}{(x^{j-1})^T (1_{n\times n} - I) x^{j-1}} \) is a constant w.r.t \( x^j \)

  - Relax to a quadratic optimization
  - Intrinsically increase the missing edge penalties in each hierarchy
HiDDen Formulation: Overall Objective Function

- Overall objective function

\[
\begin{align*}
\min_{x_1, x_2, \ldots, x_K} & 
- (1 + p)(x^1)^T A x^1 + p(\|x^1\|_1^2 - \|x^1\|_2^2) \\
\text{for } 1^{\text{st}} \text{ hierarchy} & \\
- (1 + p + \beta) \sum_{k=2}^{K} (x^k)^T A x^k + \sum_{k=2}^{K} (p + \beta C^{k-1}) (\|x^k\|_1^2 - \|x^k\|_2^2) \\
\text{for } k^{\text{th}} \text{ hierarchy} & \\
s. t. & x^{j+1} \leq x^j \leq x^{j-1}, \quad \forall j = 1, 2, \ldots, K \\
- p & \text{ is the parameter of missing edge penalty} \\
- \beta & \text{ controls the importance of the constraint relaxation} \\
- p + \beta C^{j-1} & \text{ is the increased penalty for the } k^{\text{th}} \text{ hierarchy}
\end{align*}
\]
Outline

▪ Motivations

▪ Q1: HiDDen Formulation

▪ Q2: HiDDen Algorithm

▪ Q3: HiDDen Generalizations

▪ Experimental Results

▪ Conclusions
HiDDen Algorithm

- **Observation:** a non-convex quadratic optimization problem
- **Solution:** alternative projected gradient descent method
  - \( \nabla x_1 f = -2(1 + p)Ax^1 + 2p\|x^1\|_1 1 - 2px^1 \)
  - \( \nabla x^k f = -2(1 + p + \beta)Ax^k + 2(p + \beta C^{k-1}) (\|x^k\|_1 1 - x^k) \)
  - Armijo’s rule line search
  - Stopping criterion: adopted from [Lin 2007]
- **Benefits:**
  - Converge to a stationary point
  - Time complexity: \( O(mK) \)
- **Question:** how to generalize to bipartite graph & query-specific
Outline

- Motivations
- Q1: HiDDen Formulation
- Q2: HiDDen Algorithm
- Q3: HiDDen Generalizations
- Experimental Results
- Conclusions
**HiDDen Generalizations: Bipartite Graph**

- **Key idea:** indicator vectors for two types of nodes, $x^k$ & $y^k$ ($k = 1, \ldots, K$)

- **Objective function:**

$$
\min_{x^1, \ldots, x^K, y^1, \ldots, y^K} \left\{ -(1 + p)(x^1)^T A y^1 + p \|x^1\|_1 \|y^1\|_1 \right. \\
\left. -(1 + p + \beta) \sum_{k=2}^{K} (x^k)^T A y^k + \sum_{k=2}^{K} (p + \beta c^{k-1}) \|x^k\|_1 \|y^k\|_1 \right\}
$$

**for 1st hierarchy**

**for k-th hierarchy**

- **Solution:** alternative projected gradient descent method
  - Alternate between $x^1, \ldots, x^K$ and $y^1, \ldots, y^K$
  - Stopping criterion: similar to previous

---

Arizona State University
HiDDen Generalizations: Query–Specific

- Intuition: constrain $x_i^k = 1$, for $i \in V_s$
- Challenges: could lead to a mixed integer problem
- Key Idea: relax to $x_i^k \geq \delta$, where $\delta \in (0, 1)$ is relatively large
- Objective function:
  \[
  \min_{x_1, x_2, \ldots, x_K} -(1 + p)(x^1)^T A x^1 + p(\|x^1\|_1^2 - \|x^1\|_2^2) - (1 + p + \beta) \sum_{k=2}^{K} (x^k)^T A x^k \\
  + \sum_{k=2}^{K} (p + \beta C^{j-1}) \left(\|x^k\|_1^2 - \|x^k\|_2^2\right)
  \]
  \[
  s.t. \quad x^{j+1} \leq x^j \leq x^{j-1}, \quad \forall \ j = 1, 2, \ldots, K
  \]
  \[
  x_i^{K+1} = \delta, \text{ if } i \in V_s; \text{ otherwise, } x_i^{K+1} = 0
  \]
  - Example: query for node-1 and node-2
    $x_1^k \geq 0.9, x_2^k \geq 0.9, \quad \text{for } k = 1, \ldots, K$
Outline

- Motivations
- Q1: HiDDen Formulation
- Q2: HiDDen Algorithm
- Q3: HiDDen Generalizations
- Experimental Results
- Conclusions
Experimental Setup

▪ Datasets:
  – DBLP co-author network (nodes: 38,624, edges: 200,332)
  – Autonomous system network (nodes: 6,474, edges: 25,142)
  – Financial network (account nodes: 29,851, PII nodes: 61,159)
  – Trafficking network (traffickers: 1416, word nodes: 4225)

▪ Evaluation Objectives:
  – Effectiveness: density of each hierarchy and density variety
  – Efficiency: running time and scalability

▪ Comparison Methods:

<table>
<thead>
<tr>
<th>HiDDen (Our Methods)</th>
<th>Baseline Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>□ HiDDen-Basic (quadratic programming separately)</td>
<td>□ GreedyOQC [Tsourakakis’13]</td>
</tr>
<tr>
<td>□ HiDDen-OPT (alternative gradient descent)</td>
<td>□ MURMS-Uni [Ding’08]</td>
</tr>
<tr>
<td></td>
<td>□ R1NdM [Belachew’15]</td>
</tr>
</tbody>
</table>
R1. Effectiveness Results – Unipartite Network

**Observation:** densities are higher and increase up to 1
R2. Case Study on Co-Authorship Network

- Differences between 1\textsuperscript{st} and 5\textsuperscript{th} hierarchy:

- Differences between 5\textsuperscript{th} and 10\textsuperscript{th} hierarchy:

**Observation:** (1) difference in research area; (2) most of people in 5\textsuperscript{th} hierarchy are in mid-career

**Observation:** 10\textsuperscript{th} hierarchy contain only flagship researchers
R3. Effectiveness Results – Bipartite Network

**Observation:** densities exhibit a good variety and are up to 1
R4. Case Study on Financial Network

- Differences among hierarchies for synthetic identity fraud detection problem:

**Observation:** multiple hierarchies of dense subgraph can more accurately detect the synthetic identity fraud.
R5. Case Study on Trafficking Network

- Differences among hierarchies for trafficking problem:

<table>
<thead>
<tr>
<th>Hierarchy</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>7th Hierarchy</td>
<td>33 traffickers: some of them are from same family; 8 words</td>
</tr>
<tr>
<td>3rd Hierarchy</td>
<td>815 traffickers (nearly half); words (30 in total): prostitution, girls, victims, police, underage, sex, trafficked, recruited, minor, adult, drugs, arrested, money, women, hotel</td>
</tr>
<tr>
<td>1st Hierarchy</td>
<td>1326 traffickers and 284 words</td>
</tr>
</tbody>
</table>

**Observation:** (1) most of the traffickers are forcing the underage girls for prostitution in hotels in exchange for cash, drugs, and other items; (2) some are from same family
Observation: HiDDen gains a better balance between running time and avg. density, as well as density variety.
R7. Scalability of HiDDen

Observation: HiDDen has a linear time complexity w.r.t # of edges
Outline

- Motivations ✓
- Q1: HiDDen Formulation ✓
- Q2: HiDDen Algorithm ✓
- Q3: HiDDen Generalizations ✓
- Experimental Results ✓
- Conclusions
Conclusions

- **Hierarchical Dense Subgraph Detection**
  - **Q1:** Formulation
  - **A1:** HiDDen
  - **Q2:** Algorithm
  - **A2:** Alternative Projected Gradient Descent Method
  - **Q3:** Generalizations
    - **A3:** Algorithms for bipartite graphs & query-specific

- **Results**
  - HiDDen outperform other baseline methods in density and variety
  - HiDDen has a linear time complexity

\[
\min_{x_1, x_2, \ldots, x_K} -(1 + p)(x^1)^T A x^1 + p (\|x^1\|_1^2 - \|x^1\|_2^2) - (1 + p + \beta) \sum_{k=2}^{K} (x^k)^T A x^k + \sum_{k=2}^{K} (p + \beta C^{-1}) (\|x^k\|_1^2 - \|x^k\|_2^2) \\
\text{s.t. } x^{j+1} \leq x^j \leq x^{j-1}, \quad \forall j = 1, 2, \ldots, K
\]