Community Search over Big Graphs: Models, Algorithms, and Opportunities

Xin Huang*, Laks V.S. Lakshmanan*, Jianliang Xu†

*University of British Columbia, Vancouver, Canada
†Hong Kong Baptist University, Hong Kong, China

xinhuang@comp.hkbu.edu.hk, laks@ubc.cs.ca, xujl@comp.hkbu.edu.hk
Tutorial Outline

• Introduction, Motivations, and Challenges
• Networks & Community Detection
  – Densely-connected community search
  – Attributed community search
  – Social circle discovery
  – Querying geo-social groups
• Future Work & Open Problems
Networks

• **Networks** are everywhere (e.g. chemistry, biology, social networks, the Web, etc.)
Communities

• **Communities** naturally exist in **networks**.

Blogosphere
Community Structure

- **Community structure**: Nodes with a shared latent property, densely inter-connected.

- Many reasons for communities to be formed:
  - Social Networks
  - Citation Networks
  - World Wide Web
  - Biological Networks
Basis of Community Formation

• The strength of weak ties [Mark Granovetter, 1973] and the models of small-world [Strogatz and Watts, Nature’98] both suggest
  – Strong ties are well embedded in the network
  – Weak ties span long ranges

• Given a network, how do we find all communities?
Community Detection

• **Q:** Given a network, how do we find all communities?
  
  • **A:** Find weak ties and identify communities
    
    – **Betweenness centrality** [Girvan and Newman, PNAS’02],
    
    – **Modularity** [Newman, PNAS’06]
    
    – **Graph partitioning methods** [Karypis and Kumar, SISC’08]
Overlapping Communities

• **Communities** defined by different nodes in a network may be quite different.
Community Search

• **Problem:** Given a set of **query nodes**, find densely connected communities containing them.

• State-of-the-art research focus: **Simple** and **static** graphs → **Evolving, attributed,** and **location-based** big graphs
Community Detection v.s. Community Search

• **Community detection**: identify all communities.
  – fundamental & widely studied
  – global computation (expensive)
  – static graphs (hard to handle evolving graphs)

• **Community search**: find query-dependent communities
  – useful & less studied
  – user-centered & personalized search
  – dynamic graphs
Applications

• Social circle discovery

• Planning a cocktail party/conference/workshop

• Infectious disease control

• Tag recommendation

• Protein complex identification
Community Search

5 communities containing "Jiawei Han" in DBLP collaboration network
Planning a cocktail party
Planning a cocktail party
Planning a cocktail party

Recipe for a successful party:

- Participants should be “close” to the organizers (e.g., a friend of a friend).
- Everybody should know sufficiently many in the party (on an average?).
- The graph should be connected.
- The number of participants should not be too small but...
- ...not too large either!!
- ....
- social distance not too large.

Not an easy task…
Protein Complex Identification

- Given: a protein-protein interaction network
- A set of proteins that regulate a gene that a biologist wishes to study.
- What other proteins should she study? those contained in a compact dense subgraph containing the given proteins.
Challenges

• Complexity of underlying community models

• Responsiveness requirements of query processing

• Dynamic network structures

• Massive volume of big graphs
Related Work

• **Community Detection** (Finding all communities in the entire network)
  – non-overlapping community detection [*Girvan and Newman, PNAS’02*]
  – overlapping community detection [*Ahn et al, Nature’10*]

• **Community Search** (Finding communities containing given query nodes)
  Different community models are proposed for various types of networks and query processing techniques.
  – **Structural Networks**  ---  Densely-connected community search
  – **Attributed Graphs**  ---  Attributed community search
  – **Ego-networks**  ---  Social circle discovery
  – **Location-based Social Networks**  ---  Querying geo-social groups
Part 1: Densely-connected Community Search

• In the simplest way, a graph represents a structure of interactions within a group of vertices.

• Task: finding densely-connected communities containing query nodes.

  – Quasi-clique model [Cui et al. SIGMOD’13]
  – Query-biased densest subgraph model [Wu et al. PVLDB’15]
  – K-core model [Sozio & Gionis KDD’10, Cui et al. SIGMOD’14, Li et al. PVLDB’15, Narbieri et al. DMKD’15]
  – K-truss model [Huang et al. SIGMOD’14, Huang et al. PVLDB’16]
Quasi-Clique based Model

- **α-adjacency-γ-quasi-k-clique community model**
  - **γ-quasi-k-clique**: a k-node graph with at least \( \lfloor \gamma k(k-1)/2 \rfloor \) edges.
  - **α-adjacency-γ-quasi-k-clique**: overlap \( \alpha \) vertices, where \( \alpha \leq k-1 \).

**k-clique**: a complete graph of \( k \) nodes with \( k(k-1)/2 \) edges.

- γ-quasi-k-cliques (\( \gamma = 0.8, k=4 \))
- α-adjacency-γ-quasi-k-cliques (\( \alpha = 2, \gamma = 0.8, k=4 \))
Problem: Given a query vertex $q$ in graph, the problem is to find all $\alpha$-adjacency-$\gamma$-quasi-$k$-clique containing $q$. 

A 0.8-quasi-7-clique containing $q$
K-Core

- **K-core**: every vertex has degree \textit{at least} \( k \) in this subgraph.
K-Core based Model

• Input: a graph G & a set of query nodes Q

• Output: a connected subgraph H containing Q such that
  (1) Query distance $D_Q(H) \leq$ distance constraint.
  (2) $|V(H)| \leq$ size constraint.
  (3) H is a k-core with the largest k by satisfying (1) and (2).

• Other k-core based community models:
  Local search algorithm [Cui et al. SIGMOD’14]
  Minimum-size Community [Narbieri et al. DMKD’15]
  Influential Community [Li et al. PVLDB’15]
**K-Truss**

- **Triangle**: fundamental **building blocks** of networks
- **k-truss** of graph $G$: every edge in $H$ is contained in **at least** $(k-2)$ triangles within $H$. 

[Jonathan Cohen, 2008]
K-Core V.S. K-Truss

- **K-core**: any pair of vertices within an edge may have no common neighbors.
- **K-truss**: any pair of vertices within an edge must have $k-2$ common neighbors.

![3-core](image1)

![3-truss](image2)
K-truss Community Model

• A k-truss community satisfies:
  (1) **K-truss**: each edge within *at least (k-2) triangles*
  (2) **Edge Connectivity**: all pairs of edges connected by triangles
  (3) **Maximal Subgraph**

![Diagram of two 4-truss communities for q](image)

[Huang et al., SIGMOD’14]
Problem Formulation

• **Problem:** Given a graph $G(V, E)$, a query vertex $q$, and an integer $k \geq 3$, find all $k$-truss communities containing $q$. 
Index Based Query Processing Algorithm Framework

- Several different index structures are designed for the efficient search of **k-core** and **k-truss** based communities.

- We take the **k-truss community model** as an example.
Index Based Query Processing Algorithm Framework

• **Index Construction (offline)**
  – They design a novel and compact tree-shaped structure called **TCP-index**.

• **Query Processing (online)**
  – Based on **TCP-index**, k-truss community search can be done in **optimal time** complexity.
TCP-Index Construction

- TCP-Index for vertex $x$ is a tree structure as $T_x$.
  - $T_x$ is a maximum spanning forest.
  - Build $T_x$ with weighted edges level by level.
  - $O(m)$ linear disk space, $O(|\text{Ans}|)$ optimal query time.
Query Processing using TCP-Index

- **Rationale:** If $y$, $z$ are connected via a series of edges with weight $\geq k$ in $T_x$, then $y$, $z$ are in the same $k$-truss community; We use $V_k(x, y)$ to denote all such vertices $z$.
- For example, querying 5-truss communities containing $q$.

Each edge is accessed only 2 times. Constant!!!
(First time in black; Second time in red.)
Motivation: Free Rider Effect

- **Free Rider Effect:** far away and irrelevant nodes are included into communities.
- **Classic density:** \( f(S) = \frac{|E(S)|}{|S|}, E(S) = E \cap S^2 \)
- \( f(A \cup B) > f(A) \).

Free Riders: irrelevant to query nodes

<table>
<thead>
<tr>
<th>Goodness metrics</th>
<th>A</th>
<th>A \cup B</th>
<th>A \cup C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic density</td>
<td>2.50</td>
<td>2.95</td>
<td>2.83</td>
</tr>
<tr>
<td>Edge-surplus</td>
<td>15.3</td>
<td>26.5</td>
<td>22.8</td>
</tr>
<tr>
<td>Minimum degree</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Subgraph modularity</td>
<td>2.0</td>
<td>3.6</td>
<td>4.6</td>
</tr>
<tr>
<td>Density-isolation</td>
<td>-2.6</td>
<td>3.8</td>
<td>1.5</td>
</tr>
<tr>
<td>Ext. conductance</td>
<td>0.25</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>Local modularity</td>
<td>0.63</td>
<td>0.70</td>
<td>0.78</td>
</tr>
</tbody>
</table>

[Wu et al. VLDB’15].
Free Rider Effect in Real Networks

(a) Co-author network  (b) Biological network

One existing method: classic density

[Wu et al. VLDB’15].
Query Biased Node Weighting

Node Weight: \( \pi(u) = \frac{1}{r(u)} \)

\( r(u) \): proximity value w.r.t. the query

Query biased density:
\[
\rho(S) = \frac{e(S)}{\pi(S)}
\]

\( \pi(S) = \sum_{u \in S} \pi(u) \): sum of node weights

Subgraph A becomes the query biased densest subgraph
Graph Diameter

- **Graph Diameter** of $G$: $\text{diam}(G) = \max_{u,v \in G} \{\text{dist}_G(u, v)\}$
- Fig. (a), shaded, has diameter 4, the longest shortest path span from $q_1$ to $p_1$
- But, Fig. (b) has diameter 3.

[Huang et al. VLDB’16]
Closest Truss Community Search

• Input: a graph $G$ & a set of query nodes $Q$

• Output: a connected subgraph $H$ containing $Q$ such that
  
  1. $H$ is a $k$-truss with the largest $k$
  2. $H$ has the smallest diameter among subgraphs satisfying (1).

[Huang et al. VLDB’16]
Case Study: DBLP network

Community search on DBLP network using query Q=

```
```

(a) 9-truss

(b) Closest Truss community

[Huang et al. VLDB’16]
Desiderata of Good Query Communities

• **Query nodes**: single or multiple.

• **Cohesive structure**: quasi-clique, densest subgraph, k-core, or k-truss.

• **Quality of approximation**: guaranteed or non-guaranteed.

• **Input queries**: parameter-free or user-unfriendly.
Part 2: Attributed Community Search

- Motivation: many real social networks contain attributes or predicates on the vertices.
  - Vertices: **Person (in social networks)**, Attributes: **name, interests, and skills**.
    - Facebook: link relationship, user background
    - Twitter: following/follower-ship, tweets
  - Vertices: **Protein (in PPI networks)**, Attributes: **GO (Gene-Ontology)** terms representing **molecular functions, biological processes, and cellular components**.
### Community Search in Attributed Graph

- **Structure + Semantics**: In addition to the network structure, users may aim to search for **attribute-related communities**, or **attributed communities**.

- **Input**: a graph $G$ where nodes are associated with attributes and an input query $Q$ consisting of nodes $V_q$ and attributes $W_q$.

- **Output**: a connected community $H$ containing $Q$ such that most community members are *densely inter-connected* and have *similar attributes*.

An example of collaboration attributed network
Community Search in Attributed Graph

An example attributed graph $G$

(b) $H_2$. 4-truss community on $V_q = \{q_1, q_2\}$, $W_q = \{DB, DM\}$

(c) $H_3$. 4-truss community on $V_q = \{q_1, q_2\}$, $W_q = \{DM\}$
Keyword Search

- Input: given a query consisting of nodes and attributes (keywords), e.g., $W=\{q_1, \text{DB}\}$
- Output: finds the substructure (trees or subgraphs) with minimum communication cost that connect the input keywords/nodes, where the communication cost is based on diameter, weight of spanning tree or steiner tree.

An example attributed graph $G$

Keyword Search with query $W=\{q_1, \text{DB}\}$
### A Comparison of Representative Works

Keyword Search (KS), Team Formation (TF), Densely-connected Community Search (DCS) and Attributed Community Search (ACS)

<table>
<thead>
<tr>
<th>Method</th>
<th>Topic</th>
<th>Participation Condition</th>
<th>Attribute Function</th>
<th>Cohesiveness Constraint</th>
<th>Communication Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>[17]</td>
<td>KS</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>[29]</td>
<td>TF</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>[19]</td>
<td>TF</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[28]</td>
<td>TF</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[39]</td>
<td>DCS</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[15]</td>
<td>DCS</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[26]</td>
<td>DCS</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[31]</td>
<td>DCS</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[46]</td>
<td>DCS</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>[18]</td>
<td>ACS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>[25]</td>
<td>ACS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Graph type</td>
<td>Community Detection</td>
<td>Community Search</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------</td>
<td>------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-attributed</td>
<td>[1000+ papers]</td>
<td>[10+ papers]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attributed</td>
<td>[100+ papers]</td>
<td>K-core-based: ACQ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>K-truss-based: ATC</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Attributed Community Query (ACQ)

Given a graph $G$, a vertex $q$, a set $S$ of keywords and an integer $k$, find the sub-graphs s.t. each $G_q$ satisfies:

- **Connectivity**: $G_q$ is connected and it contains $q$;
- **Structure cohesiveness**: minimum degree $\geq k$;
- **Keyword cohesiveness**: the number of keywords in $S$ shared by other vertices in $G_q$ is maximized

$q=$Jack, $k=2$, $S=$\{research, sports, tour\}

[Research & Sports]

[Fang et al. PVLDB’16]
Densely-connected Community Search [1,2]

• Who is in Jim Gray’s community?
  – “k-core” (with Local algo. [2]), nodes connected by k=4 or more edges

• Why are these people considered as Jim’s community?
• What is the theme of this community?

A community can have $10^5$ nodes!

Attributed Community (AC)

• Previous CS solutions overlook **keywords**
  – e.g., a researcher’s interest
Attributed Community (AC)

- In fact, Jim has 2 distinct attributed communities (AC).

Common keyword set (AC label)
Part 3: Social Circle Discovery

- **Social circles**: communities formed by only friends

Social Circle in Facebook
An Ego-network

- **Ego-network**: an induced subgraph of a network only by her friends.
Social Circle Discovery

• **Examples:** online social networks allow users to manually categorize their friends into social circles within their ego network (e.g., circles on Google+)

• **Social circle discovery:** the task is to automatically identify all social circles for a given user.

• **Applications:**
  – content filtering
  – privacy protection
  – sharing groups of users that others may wish to follow
Learning to discover social circles

- **An unsupervised community model** predicts hard memberships to multiple, overlapping circles, using both **user profile** and **network structure**.

\[ p((x, y) \in E) \propto \exp \left\{ \sum_{C_k \ni \{x, y\}} \langle \phi(x, y), \theta_k \rangle - \sum_{C_k \not\ni \{x, y\}} \alpha_k \langle \phi(x, y), \theta_k \rangle \right\} \]

Training is done by maximum likelihood, using QPBO and L-BFGS.
Datasets: Ground-truth Social Circles

- Datasets are collected from real-world networks Facebook, Google+, and Twitter

<table>
<thead>
<tr>
<th></th>
<th>ego-networks circles</th>
<th>nodes</th>
<th>edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>10</td>
<td>193</td>
<td>4,039</td>
</tr>
<tr>
<td>Google+</td>
<td>133</td>
<td>479</td>
<td>107,614</td>
</tr>
<tr>
<td>Twitter</td>
<td>1,000</td>
<td>4,869</td>
<td>81,306</td>
</tr>
</tbody>
</table>

All data are available on [snap.stanford.edu/data/](http://snap.stanford.edu/data/).
Detected Circles

Facebook:

Google+:

Blue = true positive; gray = true negative; red = false positive; yellow = false negative; green = detected circles for which we have no groundtruth.
Social Contagion

• Case Study (Facebook)  
  [Ugander et al, PNAS’12]

Social circles can affect the process of information diffusion on social contagion.

Consider an existing Facebook user invites the non-existing Facebook user to join Facebook.

We want to study the success rate that this non-existing user will join Facebook.

---

Friendship neighborhood:

Existing Facebook user

Non-Existing Facebook user
Social circles can affect the process of information diffusion on social contagion. Consider an existing Facebook user invites the non-existing Facebook user to join Facebook. We want to study the success rate that this non-existing user will join Facebook.

Social Contagion

- Case Study (Facebook)  
  [Ugander et al, PNAS’12]

The number of connected components is related to the success rate.

[Ugander et al. PNAS’12].
Top-K Structural Diversity Search

• **The structural diversity** of a node is defined to be the number of connected components in its ego-network.

• **Problem**
  – Find k nodes with the greatest structural diversity in a social network (Node Ranking).

• **Application**
  – Political campaign, promotion of health practices, marketing
Part 4: Querying Geo-Social Groups

• Boom in geo-social networks
  – Foursquare, Facebook, Weibo, DaZhongDianPing, Yelp, Flickr
  – Social networks coupled with user locations

• Group-based activity planning and business
  – Find a group of friends at the conference for gathering
  – Find a group of nearby friends for sports, ridesharing, groupon...
Geo-Social Group Queries (GSGQ)

- Given an LBSN $G=(V, E)$, a query user $v_q \in V$ and an integer $c \geq 1$, find a group of users $V' \subseteq V$ containing $v_q$ and satisfying:
  - Social constraint: $G[V'] \in G$ is a $c$-core
  - Spatial constraint:
    - Range: all users of the group are in a given spatial range
    - $k$NN: the closest group with $k$ other users (NP-hard!)

- Range: $c = 2$, $V' = \{v_1, v_2, v_5, v_6\}$
- $k$NN: $c = 2$, $k = 2$ $V' = \{v_1, v_2, v_6\}$
Key Concept

• **Core Bounding Rectangle (CBR):** Given $G=(V, E)$, a node $v$, an integer $c \geq 1$, $CBR_{v,c}$ is a rectangle that covers $v$ and in which any user group containing $v$ cannot form a $c$-core.
  - $CBR_{v,c1} \subseteq CBR_{v,c2}$, if $c1 < c2$
  - Construction cost: $O(|E| \log |V|)$

**Pruning:** exclude $v$ from result group if query range $\subset CBR$
Geo-Social K-Cover Group Queries

• **Problem:** Given an LBSN $G(V, E)$, a set of query points $P=\{p_1, p_2, \ldots, p_m\}$, and an integer $k \geq 1$, find a group of users $V' \subseteq V$ satisfying:

1) **Spatial constraint:** $P \subseteq \bigcup_{u \in V'} u.R$

2) **Social constraint:** $G[V'] \in G$ is a $c$-core

3) **Size requirement:** $|V'|$ is minimum

• $c = 2$, $P=\{p_1, p_2, p_3, p_4\}$, $V' = \{u_1, u_3, u_4\}$

![Diagram](a) Social networks  ![Diagram](b) Associated regions
Applications

• **Spatial task outsourcing**: identify a group of workers whose service regions collectively cover the locations of spatial tasks

• **Travel Recommendation**: find a minimum group of tourists for a self-drive tour of a set of POIs

• **Collaborative team organization**: find a collaborative team to promote products in several market areas

[Li et al. ICDE 2016]
Other Geo-Social Group Queries

• Spatial-Aware Community (SAC) Search
  – Y. Fang, et al., “Effective Community Search over Large Spatial Graphs” [PVLDB’17]
  – **Problem**: Given a graph $G(V, E)$, an integer $c$, and a query vertex $q \in V$, find a subgraph $G_q \subseteq G$:
    1. **Connectivity**: $q \in G_q$ is connected
    2. **Structure cohesiveness**: $\forall v \in G_q$, $\deg_{G_q}(v) \geq c$
    3. **Spatial cohesiveness**: smallest *minimum covering circle*

• $q=Q$ and $c=2$,
  $G_q = \{Q, C, D\}$
Open Problems & Future Directions

• **Heterogeneous Information Networks**

  • Scalability
    – I/O-efficient algorithms & distributed computing
    – Stream graphs

• **Public-Private Social Networks**

• **Community Search on Uncertain Graphs**
  – Probabilistic k-core & Probabilistic k-truss
Heterogeneous Information Networks

• Information network: A network where each node represents an entity (e.g., actor in a social network) and each link (e.g., tie) a relationship between entities.

• Homogeneous vs. heterogeneous networks
  – Homogeneous networks
    • Single object type and single link type
    • Single model social networks (e.g., friends)
  – Heterogeneous, multi-typed networks
    • Multiple object and link types
    • Healthcare network: patients, doctors, disease, hospitals, treatments
Heterogeneous Information Networks
Open Problems & Future Directions

• Heterogeneous Information Networks

• **Scalability**
  – I/O-efficient algorithms & distributed computing
  – Stream graphs

• Public-Private Social Networks

• Community Search on Uncertain Graphs
  – Probabilistic k-core & Probabilistic k-truss
Scalability

- **Scaling community search techniques** to the *massive and rapidly growing network datasets* of the Big Data era.

- **I/O efficient algorithms**: k-core decomposition and k-truss decomposition.

- **Distributed graph computing**: Pregel and Blogel.

- **Streaming graphs**: handling community indexes in highly evolving graphs.
Scalability

Data-Parallel

Table

- Row
- Row
- Row
- Row

Result

Graph-Parallel

Property Graph

- Pregel
- GraphLab
- Giraph
Open Problems & Future Directions

• Heterogeneous Information Networks

• Scalability
  – I/O-efficient algorithms & distributed computing
  – Stream graphs

• Public-Private Social Networks

• Community Search on Uncertain Graphs
  – Probabilistic k-core & Probabilistic k-truss
Public-Private Social Networks

• **Background:** In Facebook social network, 52.6% of 1.4 million New York City Facebook users hid their friends list.

微博悄悄关注(Secretly follow in Weibo networks)

• Public-Private graph model contains a **public graph**, in which each node is also associated with a **private graph**.

  – The public graph is visible to everyone, but each private graph is visible only to the corresponding user.
Open Problems & Future Directions

• Heterogeneous Information Networks

• Scalability
  – I/O-efficient algorithms & distributed computing
  – Stream graphs

• Public-Private Social Networks

• **Community Search on Uncertain Graphs**
  – Probabilistic k-core & Probabilistic k-truss
Not all real-world networks are deterministic graphs.

Probabilistic/Uncertain Graphs: each edge has an existence probability.
Probabilistic Graphs: Examples

- Topologies of wireless sensor networks (WSNs)
  - Vertices: sensor nodes
  - Edges: wireless links between sensor nodes
  - Uncertainties: probabilities of wireless links functioning
Discovery of communities in uncertain graphs

• Benefits:
  – Find most influential communities in social networks.
  – Functional module identification for helping critical clinical diagnosis of diseases such as cancer in biology.

• K-core and k-truss have been studied in probabilistic graphs.

• An exciting question is how to generalize various community models and search techniques to probabilistic graphs.
References


References II


Thank you!

Questions?

xinhuang@comp.hkbu.edu.hk