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*

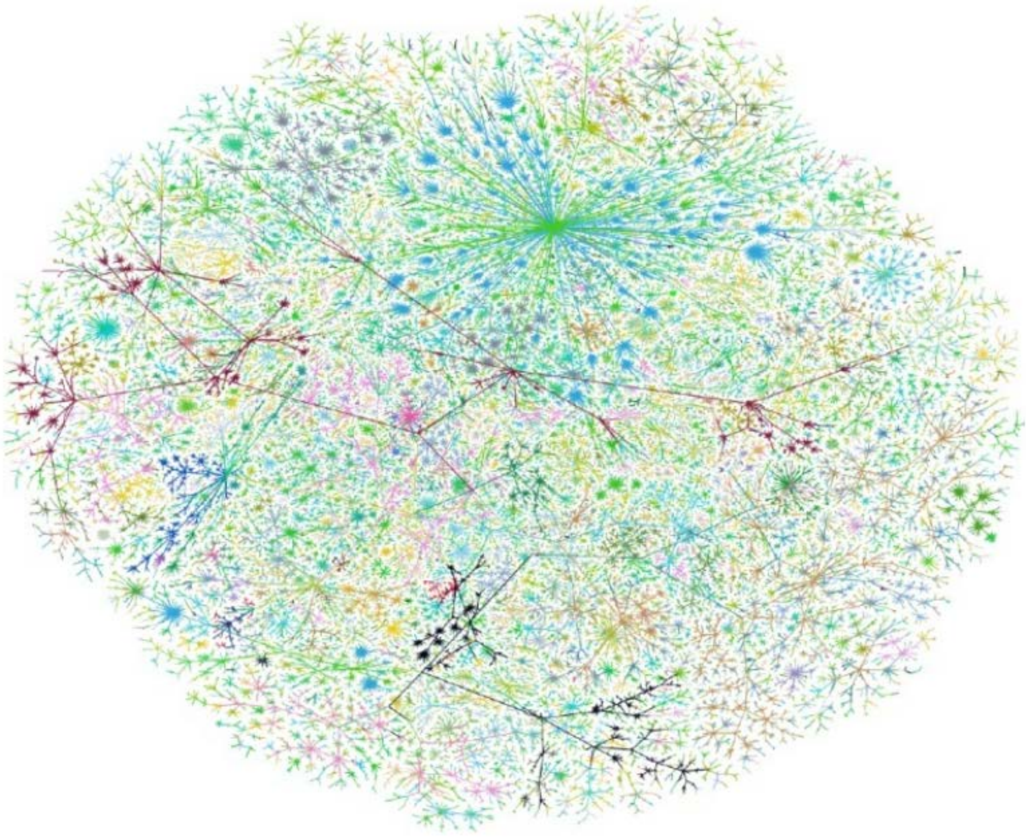
xinhuang@comp.hkbu.edu.hk, *laks@ubc.cs.ca*, *xujl@comp.hkbu.edu.hk*

Tutorial Outline

- Introduction, Motivations, and Challenges
- Networks & Community Detection
- Community Search (4 Parts)
  - Densely-connected community search
  - Attributed community search
  - Social circle discovery
  - Querying geo-social groups
- Future Work & Open Problems

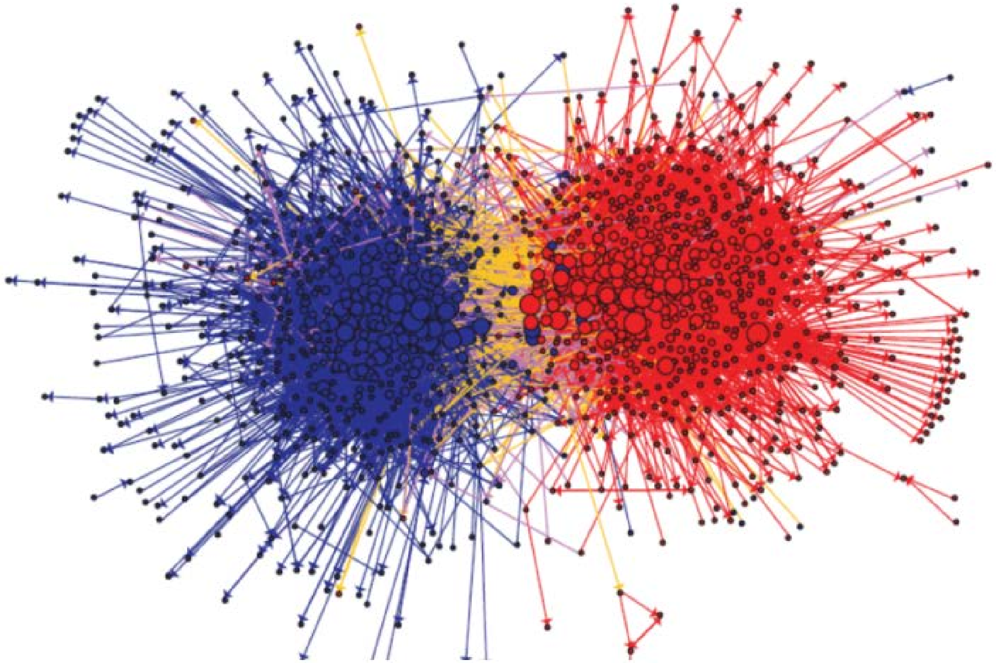
Networks

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



Communities

- **Communities** naturally exist in **networks.**



Blogosphere

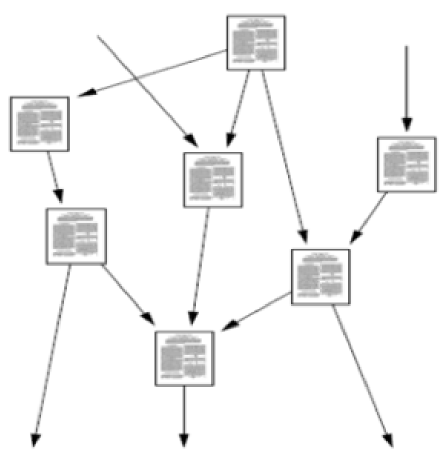
Community Structure

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

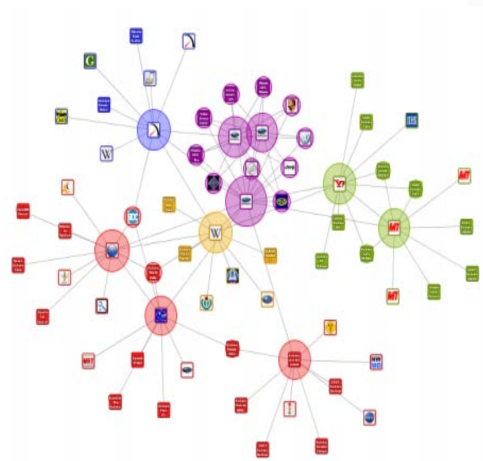
- Many reasons for communities to be formed:

Social Networks

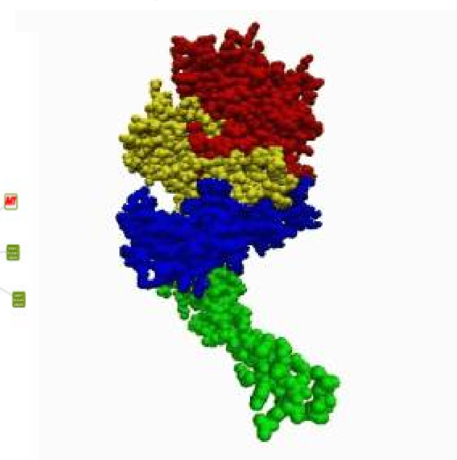
Citation Networks



World Wide Web

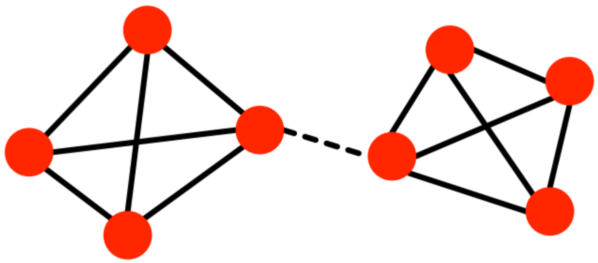


Biological Networks



Basis of Community Formation

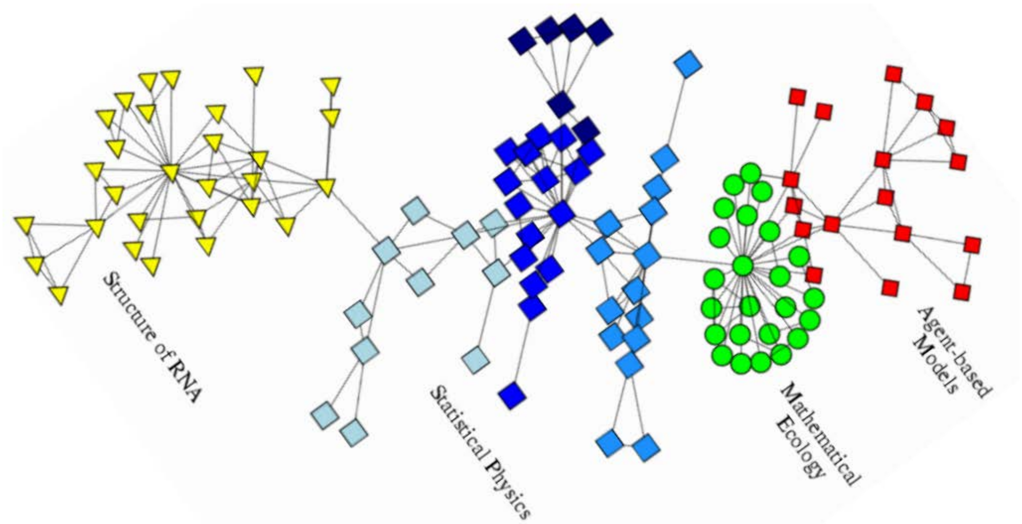
- **The strength of weak ties** *[Mark Granovetter,1973]* and **the models of small-world** *[Strogatz and Watts, Nature'98]* both suggest
  - **Strong ties** are well embedded in the network
  - **Weak ties** span long ranges



- **Given a network, how do we find all communities?**

Community Detection

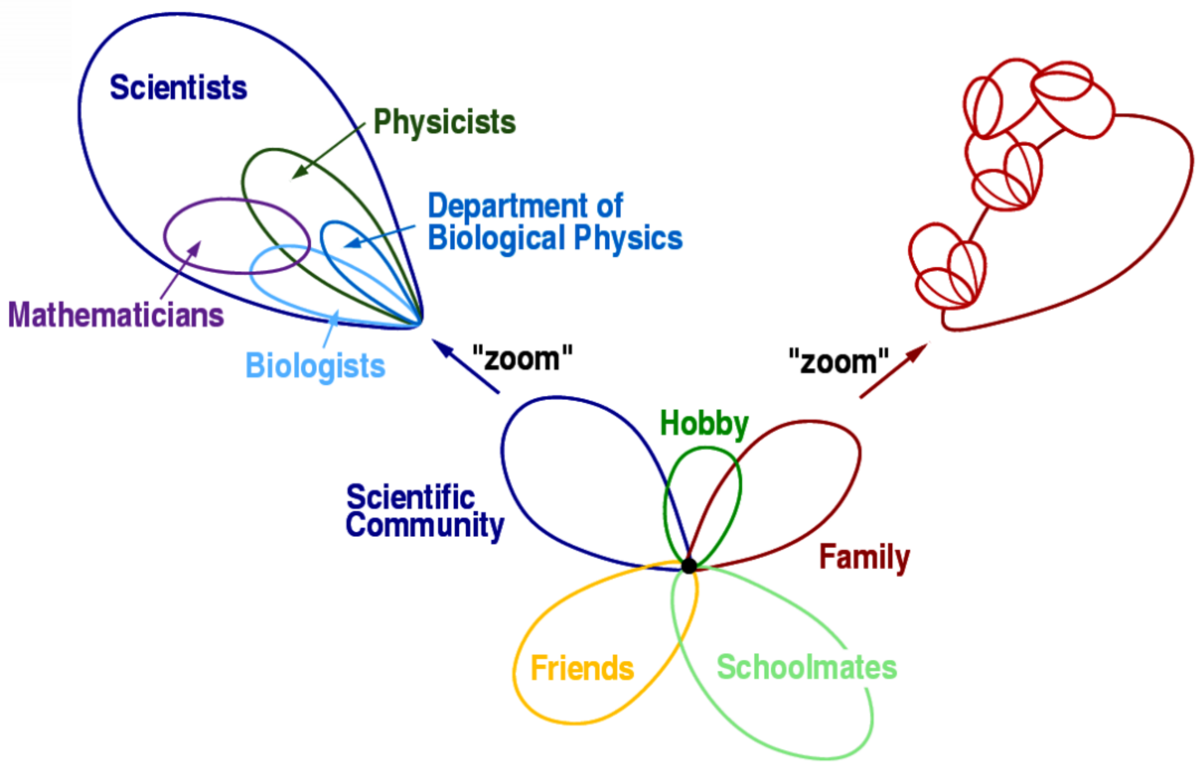
- **Q: Given a network, how do we find all communities?**
- **A: Find weak ties and identify communities**
  - **Betweenness centrality** *[Girvan and Newman, PNAS'02],*
  - **Modularity** *[Newman, PNAS'06]*
  - **Graph partitioning methods** *[Karypis and Kumar, SISC'08]*



SFI collaboration network [Newman]

Overlapping Communities

- **Communities** defined by **different nodes** in a network may be quite different.



Community Search

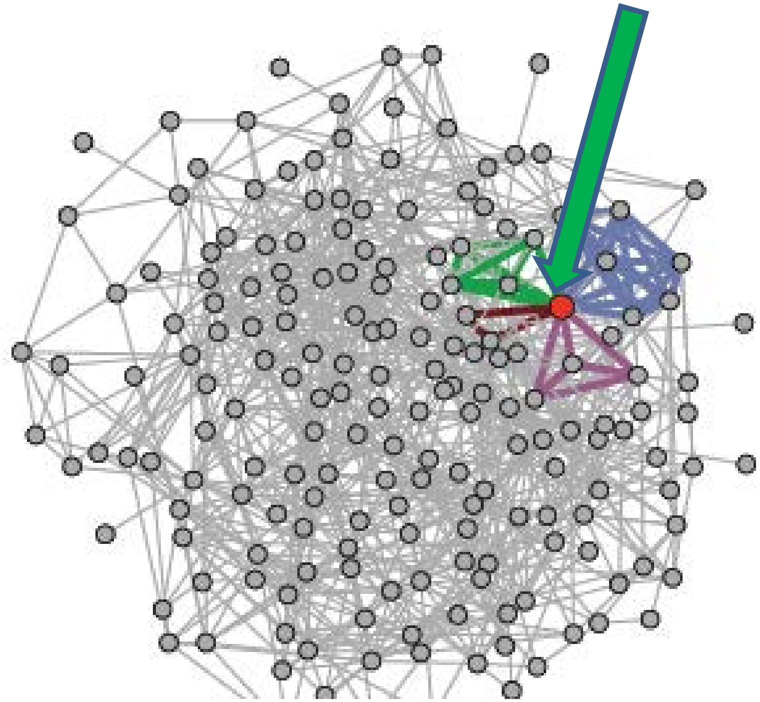
- **Problem:** Given a set of **query nodes**, find densely connected communities containing them.

- State-of-the-art research focus:

Simple and **static** graphs →

Evolving, attributed, and

location-based big graphs



Community Detection v.s. Community Search

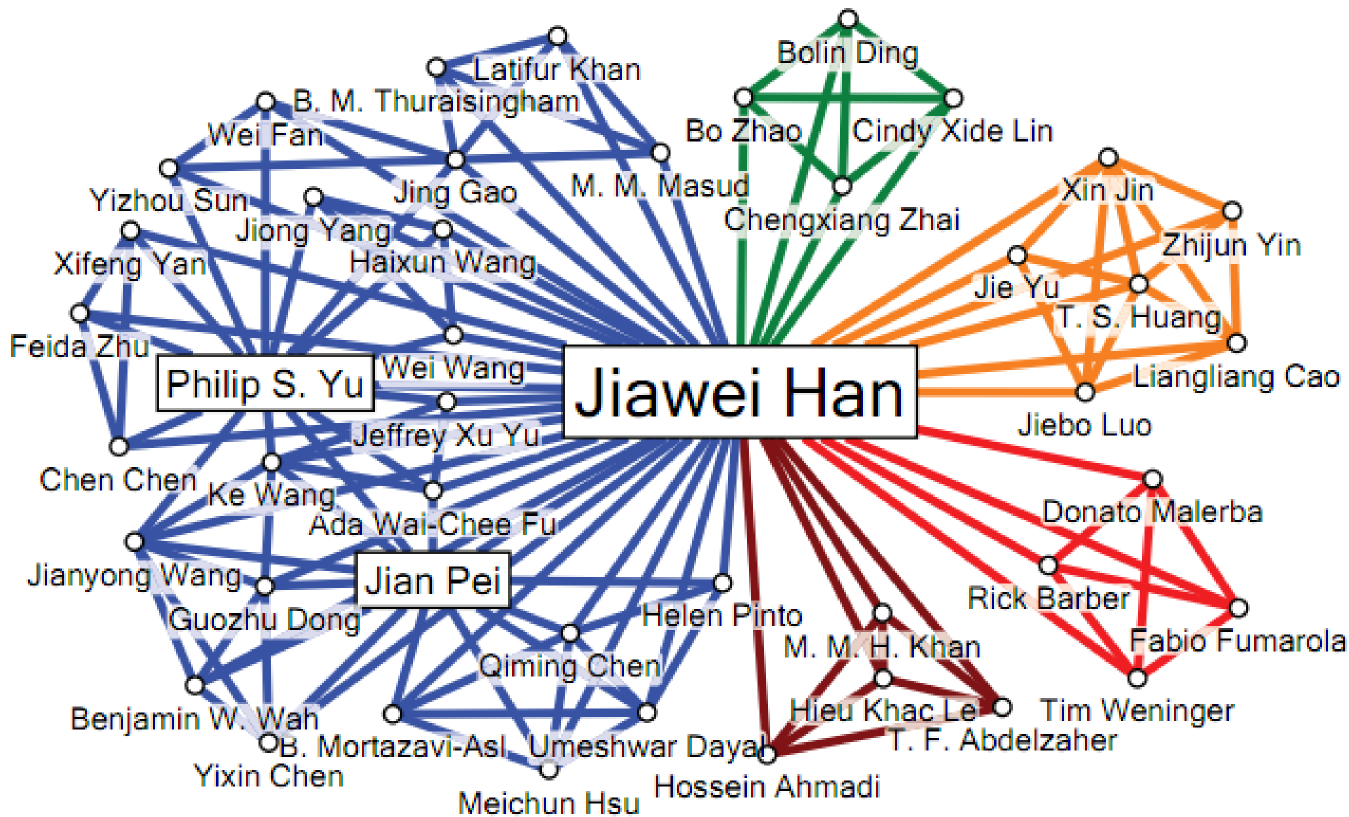
- **Community detection**: **identify all communities.**
  - fundamental & widely studied
  - global computation (expensive)
  - static graphs (hard to handle evolving graphs)

- **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**

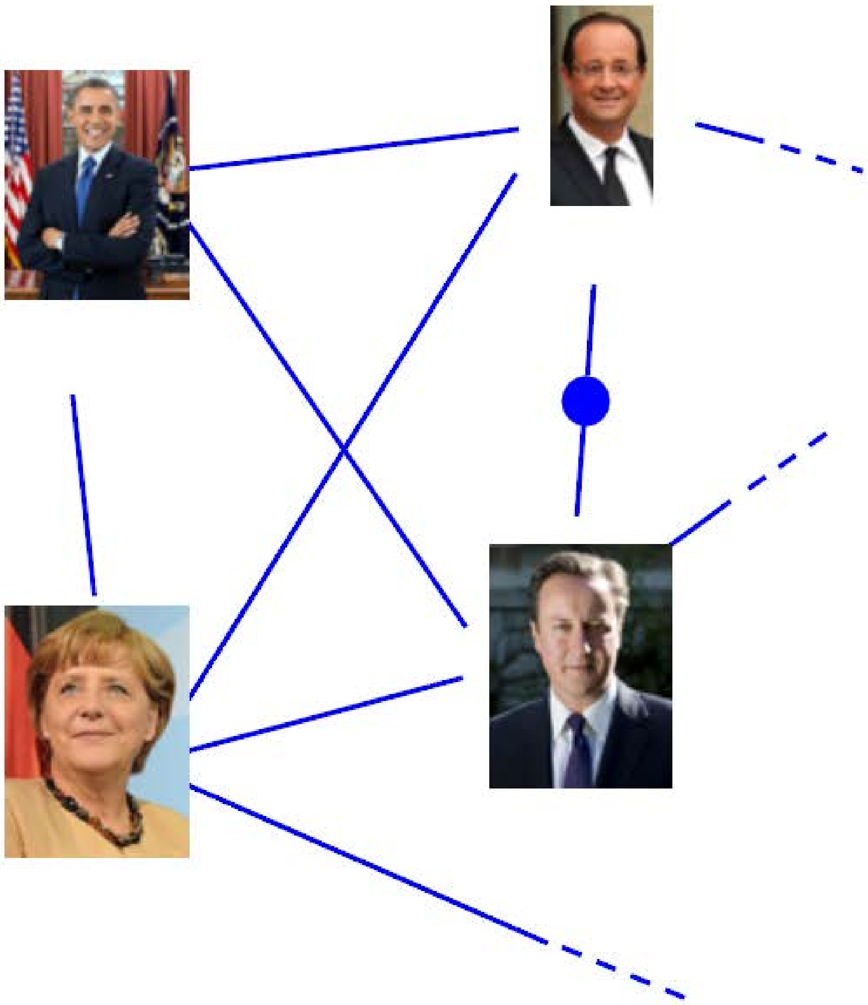
Community Search



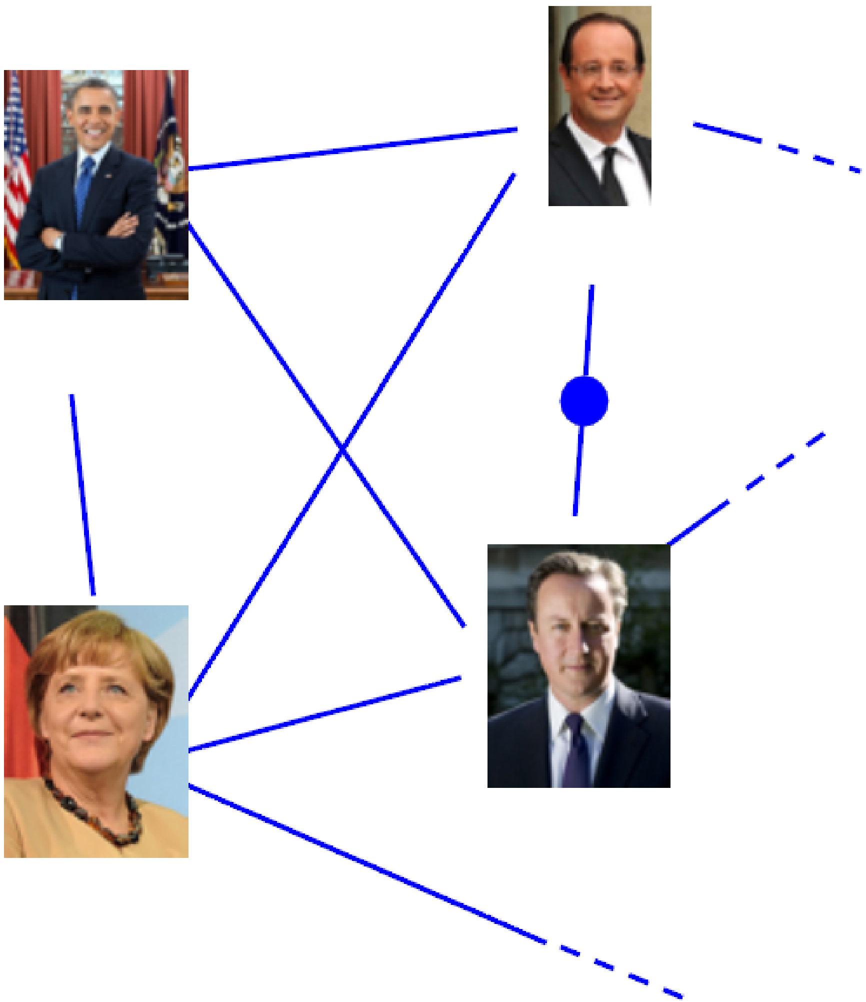
5 communities containing "Jiawei Han" in DBLP collaboration network

Planning a cocktail party

Planning a cocktail party



Planning a cocktail party



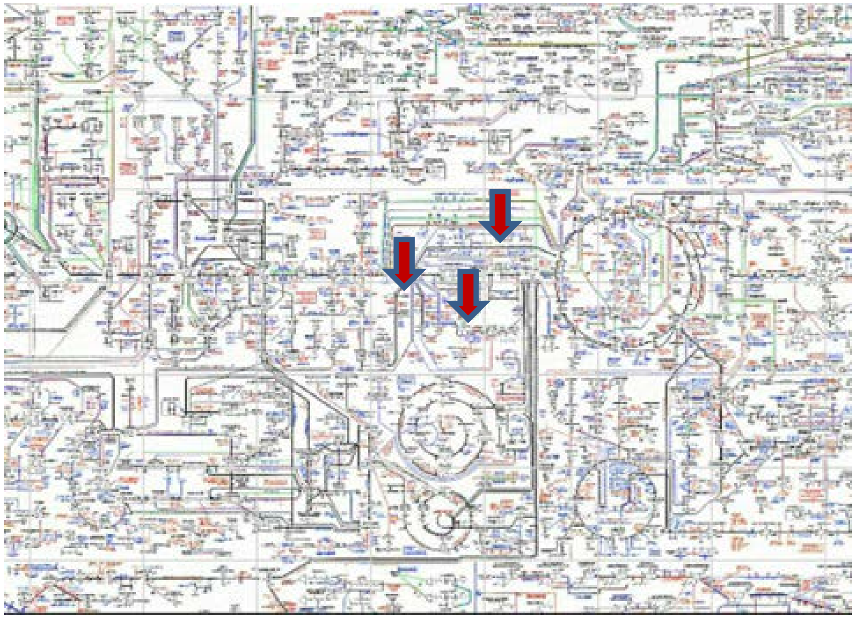
Recipe for a successful party:

- 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

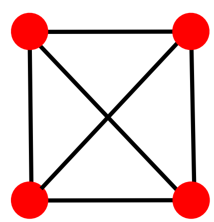
Part 1: Densely-connected Community Search

- 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

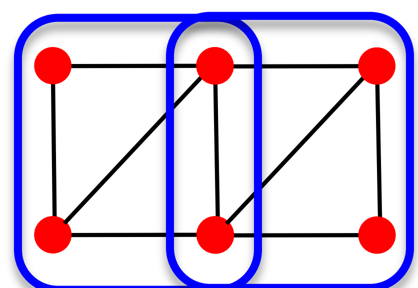
- **α-adjacency-γ-quasi-k-clique community model**
  - **γ-quasi-k-clique:** a k-node graph with at least $\lfloor \gamma k(k-1)/2 \rfloor$ edges.
  - **α-adjacency-γ-quasi-k-clique:** overlap α vertices, where $\alpha \leq k-1$.

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



γ-quasi-k-cliques
(γ=0.8, k=4)

k-cliques
(k=4)

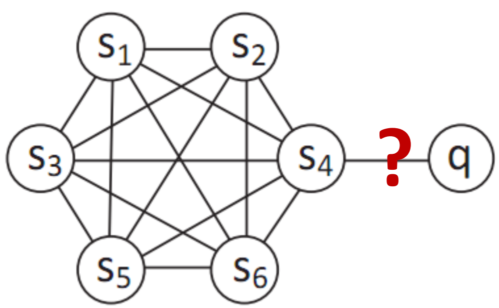


α-adjacency-γ-quasi-k-cliques
(α=2, γ=0.8, k=4)

[Cui et al., SIGMOD'13]

Quasi-Clique based Model

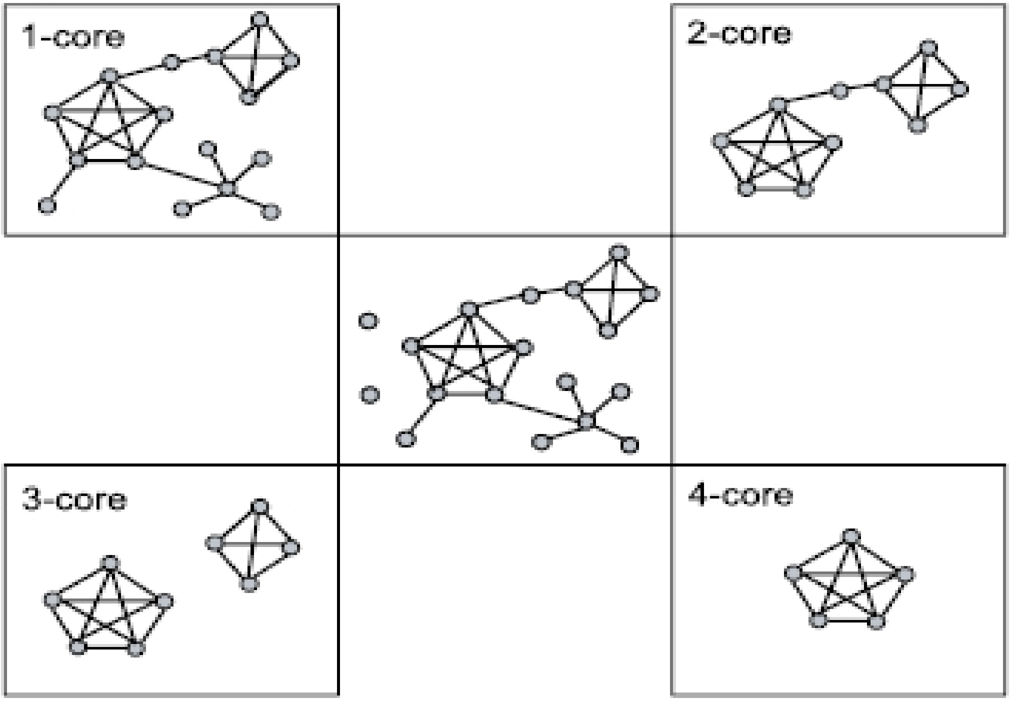
- **Problem:** Given **a query vertex q** in graph, the problem is to find all **α-adjacency-γ-quasi-k-clique** containing q.



A 0.8-quasi-7-clique containing q

K-Core

- **K-core**: **every vertex** has **degree *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 ***$D_Q(H)$ <= distance constraint.***

  (2) |V(H)| ***<= 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]*

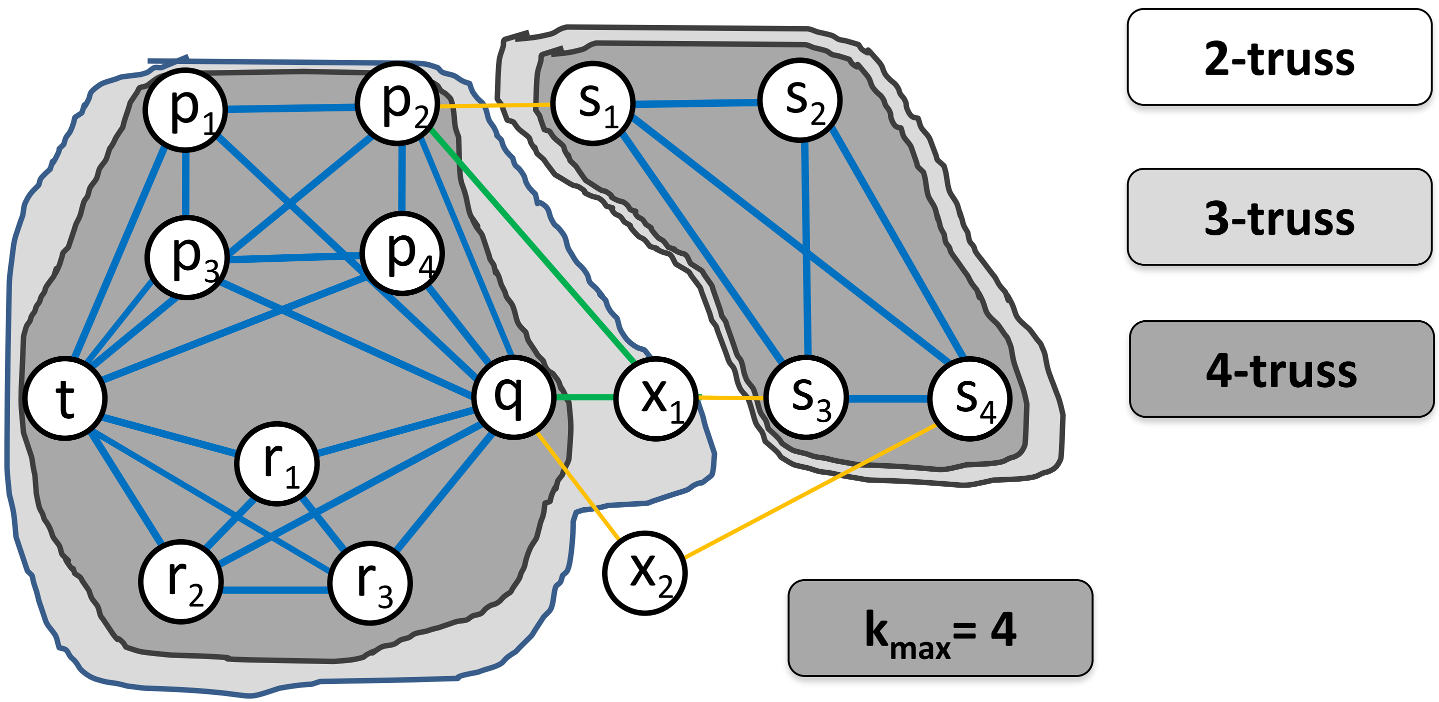
  Minimum-size Community *[Narbieri et al. DMKD’15]*

  Influential Community *[Li et al. PVLDB’15]*

[Jonathan Cohen, 2008]

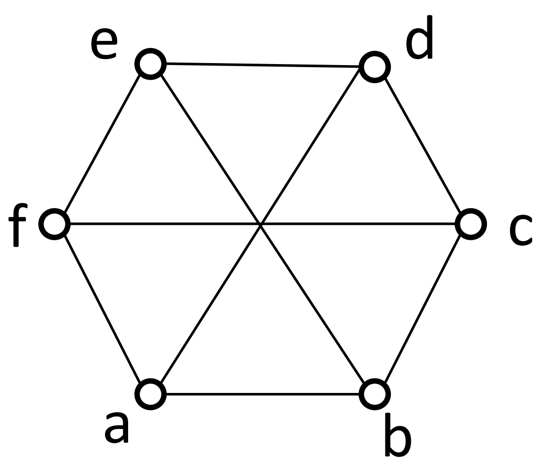
K-Truss

- **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.

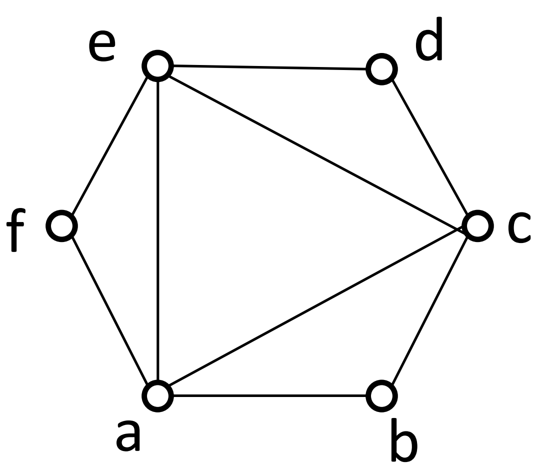


K-Core V.S. K-Truss

- **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

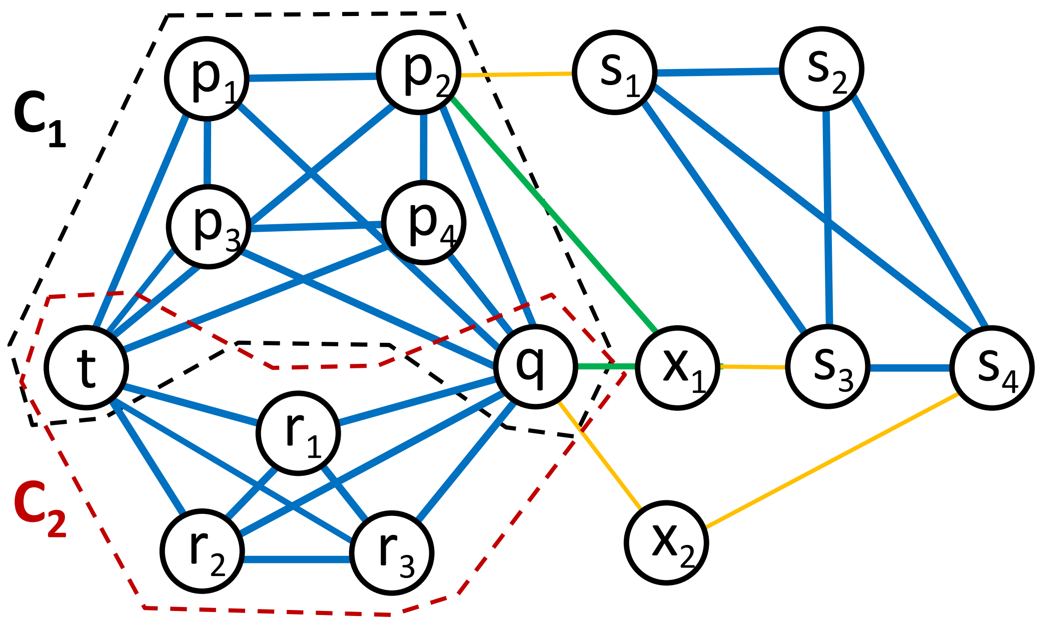


3-truss

[Huang et al., SIGMOD'14]

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**



Two 4-truss communities for q

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.

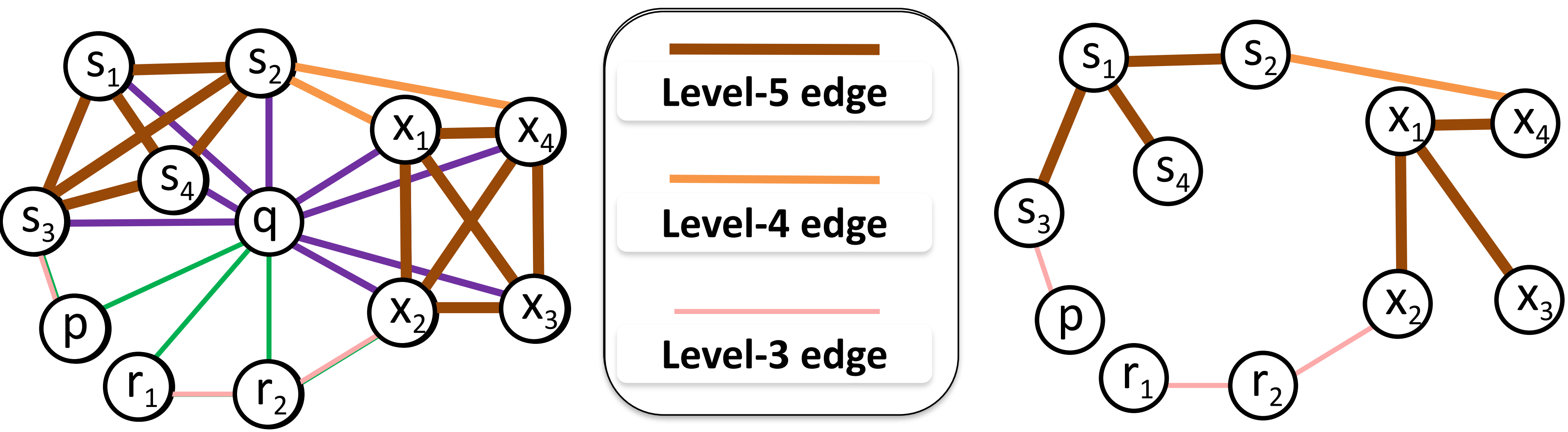
- We take the **k-truss community model** as an example.

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(|Ans|) optimal query time.

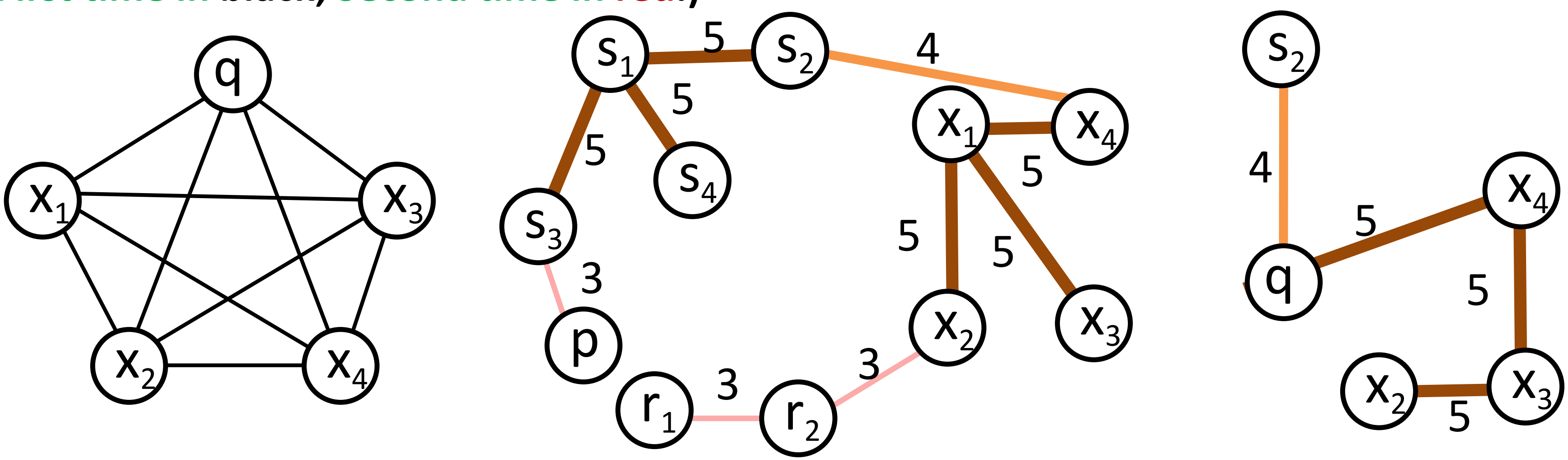


G_q T_q

Query Processing using TCP-Index

- **Rationale**: If y, z are connected via a series of edges with weight $\geq$ k in $\mathbf{T_x}$, then y, z are in the same k-truss community; We use $\mathbf{V_k(x, y)}$ 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.)



A complete 5-truss community

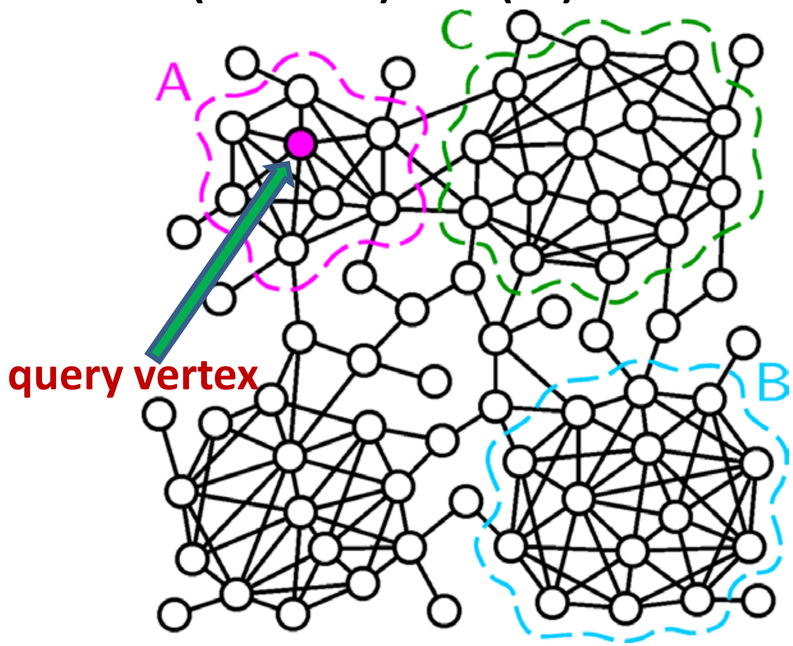
T_q

T_{x_1}

[Wu et al. VLDB'15].

Motivation: Free Rider Effect

- **Free Rider Effect:** far away and irrelevant nodes are included into communities.
- **Classic density:** f(S) =|E(S)|/|S|, E(S)=E∩S^2
- f(A ∪ B) > f(A) .



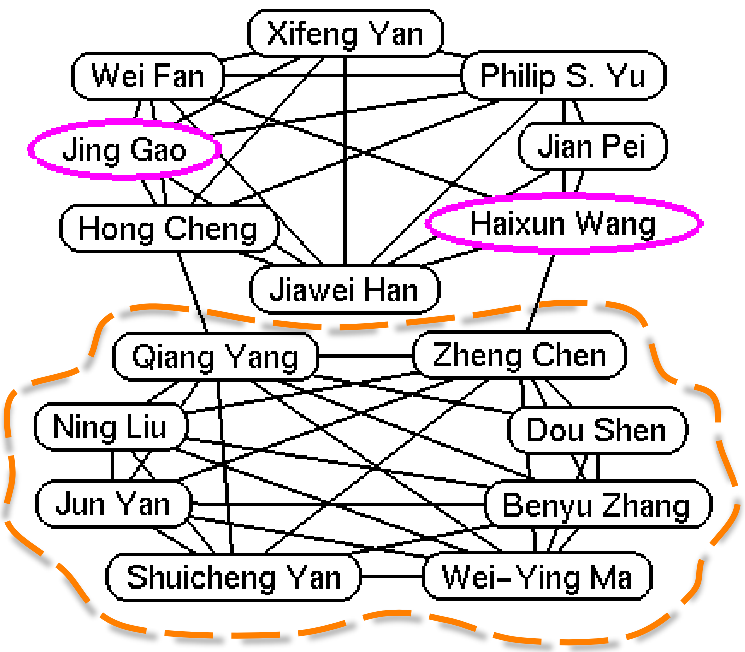
Classic density: $|E|/|V|$

Goodness metrics	A	A ∪ B	A ∪ C
Classic density	2.50	**2.95**	2.83
Edge-surplus	15.3	**26.5**	22.8
Minimum degree	4	4	4
Subgraph modularity	2.0	3.6	**4.6**
Density-isolation	-2.6	**3.8**	1.5
Ext. conductance	0.25	0.14	**0.11**
Local modularity	0.63	0.70	**0.78**

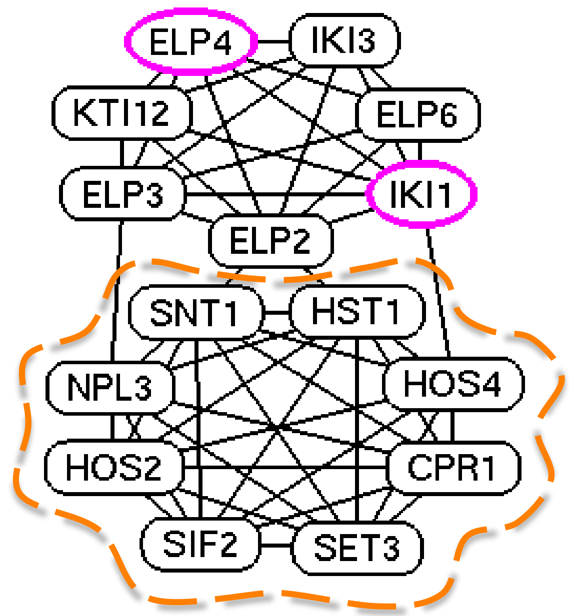
Free Riders: irrelevant to query nodes

[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

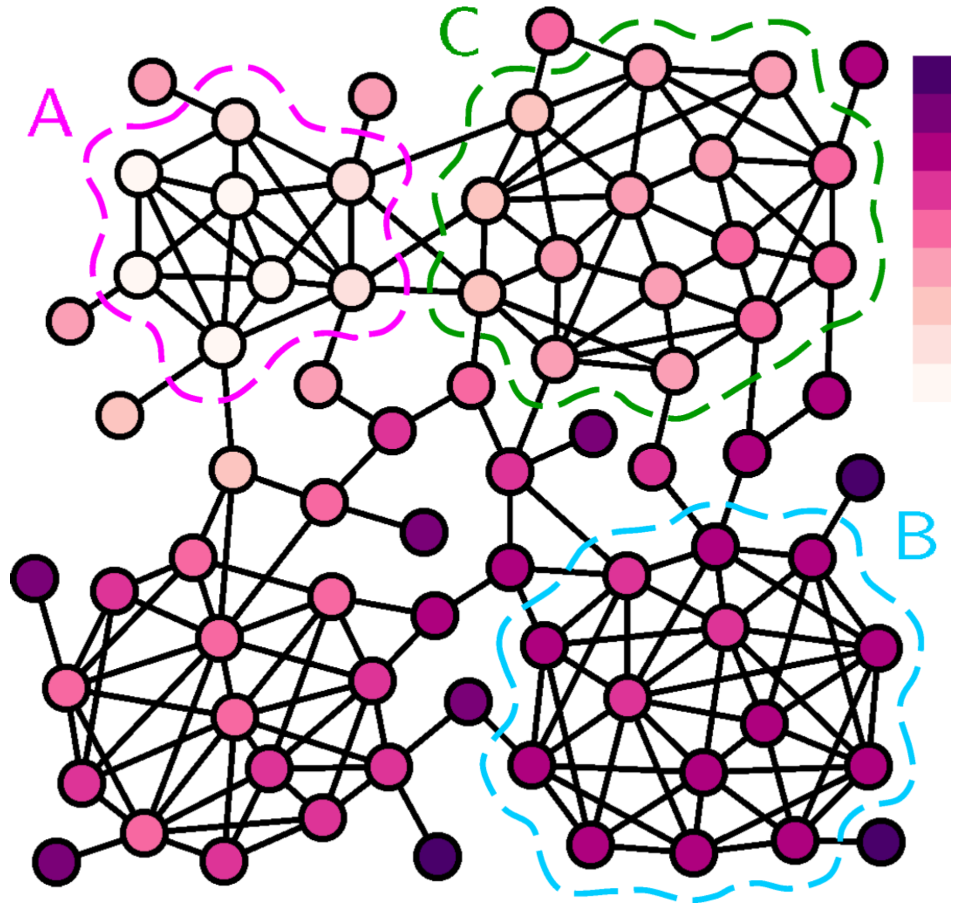
Query biased density:

$$\rho(S) = \frac{e(S)}{\pi(S)}$$

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

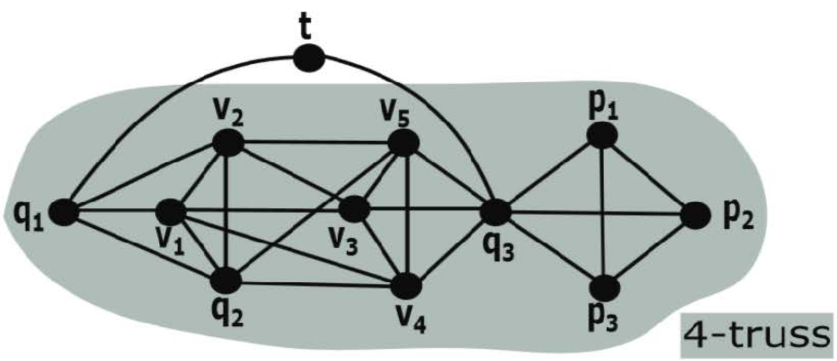
Subgraph **A** becomes the

query biased densest subgraph

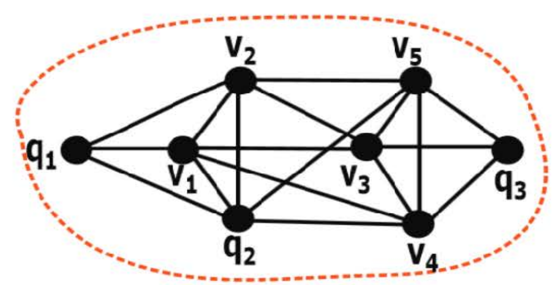


Graph Diameter

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



(a) Graph G



(b) Closest Truss Community for Q={q1, q2, q3}

Closest Truss Community Search

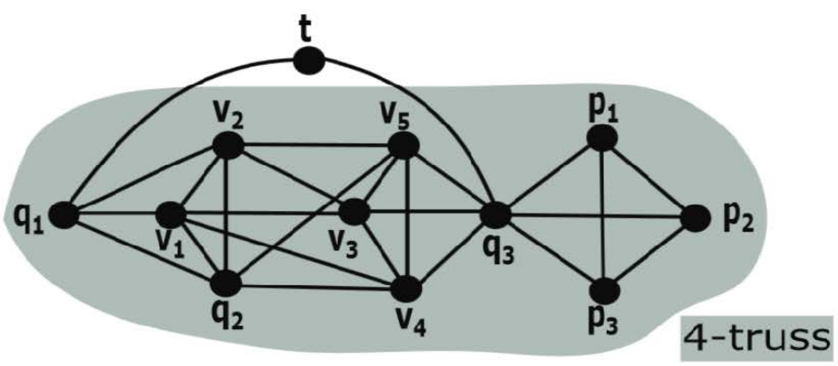
- Input:

  a graph G & a set of query nodes Q

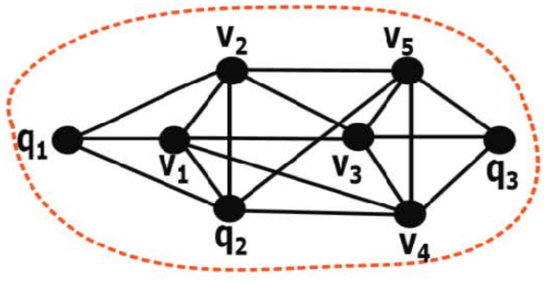
- Output: **a connected subgraph H containing Q such that**

  (1) ***H*** *is* ***a k-truss with the largest k***

  (2) ***H*** *has* ***the smallest diameter among subgraphs satisfying (1).***

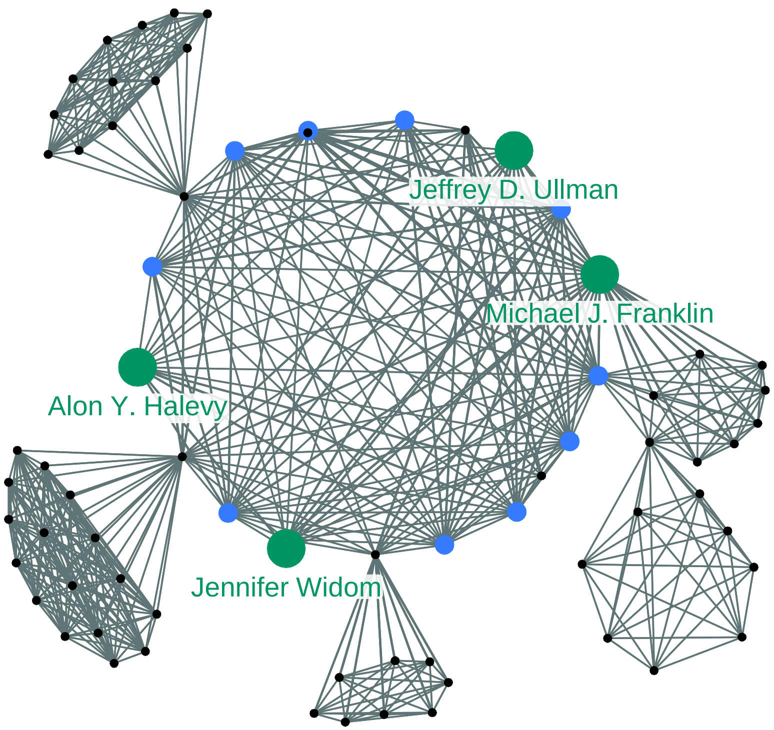


(a) Graph G

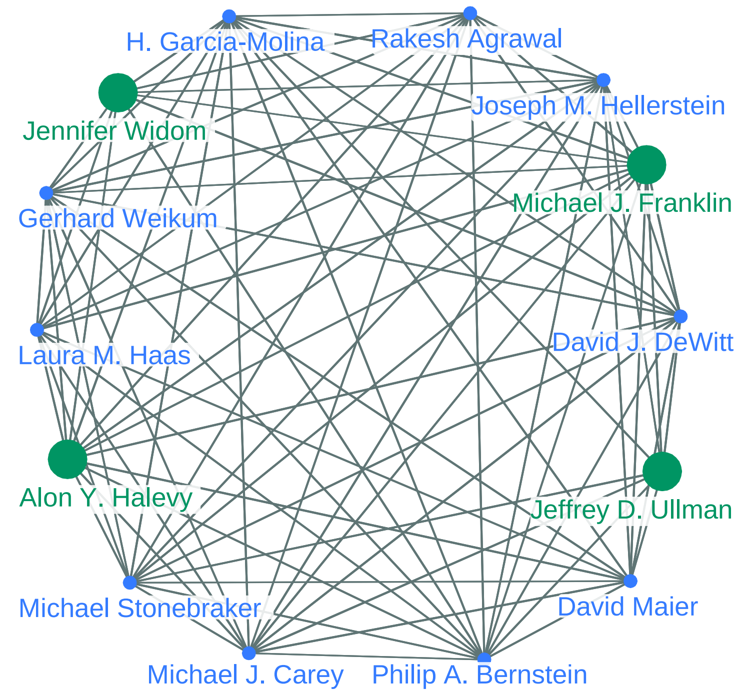


(b) Closest Truss Community for Q={q1, q2, q3}

Case Study: DBLP network



(a) 9-truss



(b) Closest Truss community

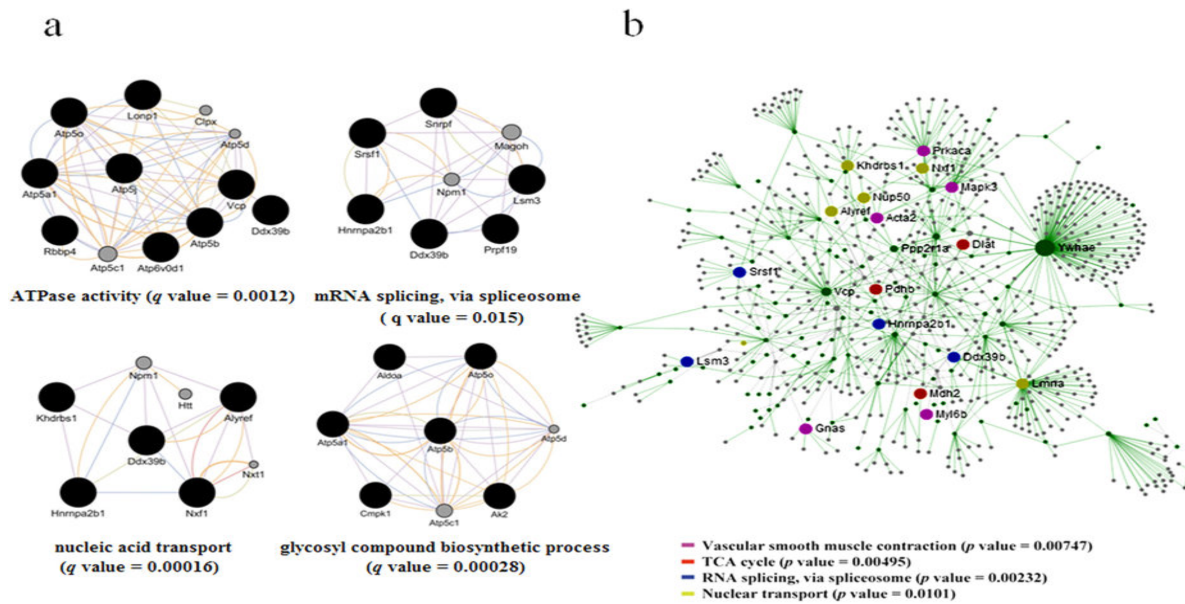
Community search on DBLP network using query Q={ **"Alon Y. Halevy", "Michael J. Franklin", "Jeffrey D. Ullman", "Jennifer Widom"** }

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

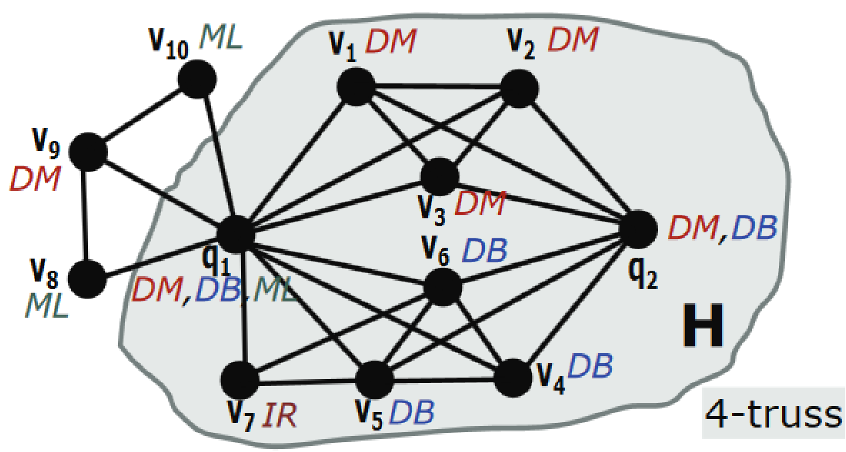
- Motivation: many real social networks contain attributes or predicates on the vertices.
  - Vertices: Person (in social networks), Attributes: name, interests, and skills.
    - Facebook: link relationship, user background
    - Twitter: following/follower-ship, tweets
  - Vertices: Protein (in PPI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



Community Search in Attributed Graph

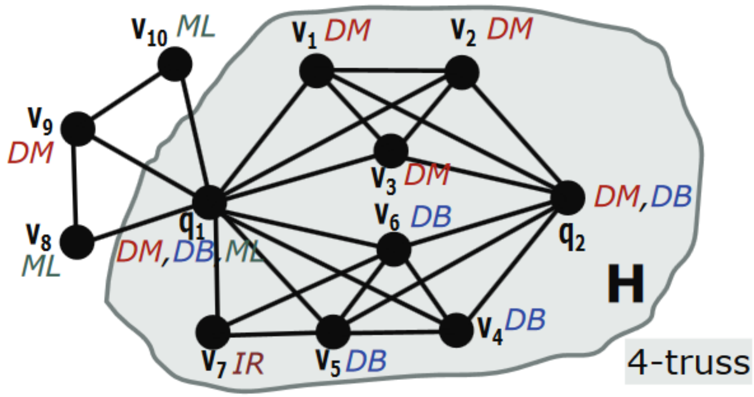
- **Structure + Semantics:** In addition to the **network structure**, users may aim to search for **attribute-related communities**, or **attributed communities**.
- Input: **a graph G** where nodes are associated with attributes

  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***

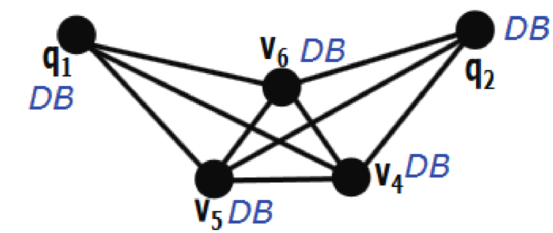


An example of collaboration attributed network

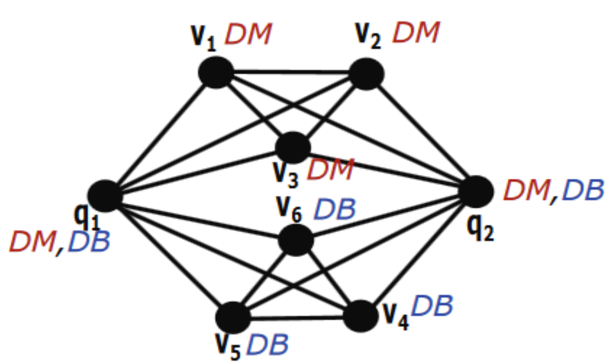
Community Search in Attributed Graph



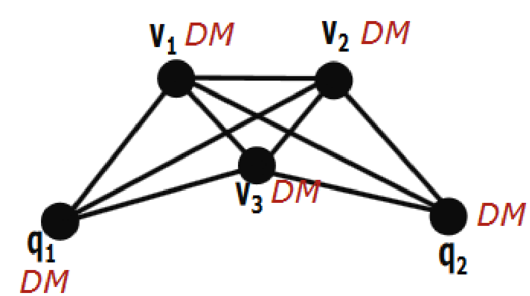
An example attributed graph G



(a) H_1. 4-truss community on $V_q = \{q_1, q_2\}, W_q = \{DB\}$



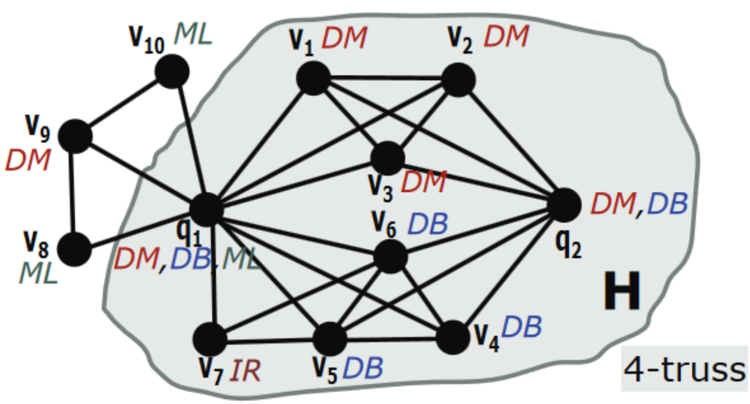
(b) H_2. 4-truss community on $V_q = \{q_1,q_2\}, W_q = \{DB, DM\}$



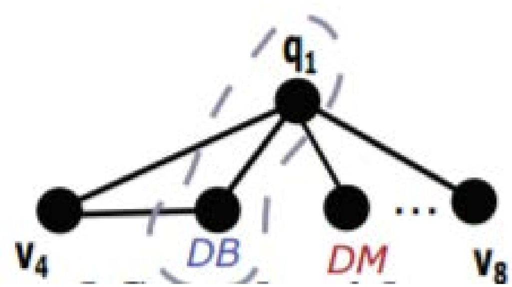
(c) H_3. 4-truss community on $V_q = \{q_1, q_2\}, W_q = \{DM\}$

Keyword Search

- Input: given a query consisting of nodes and attributes (keywords), e.g., W={q_1, 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, DB}

A Comparison of Representative Works

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

Method	Topic	Participation Condition	Attribute Function	Cohesiveness Constraint	Communication Cost
[6]	KS	χ	✓	χ	✓
[17]	KS	χ	✓	χ	✓
[30]	KS	χ	✓	χ	✓
[29]	TF	χ	✓	χ	✓
[19]	TF	χ	✓	✓	✓
[28]	TF	χ	✓	χ	✓
[39]	DCS	✓	χ	✓	✓
[14]	DCS	✓	χ	✓	χ
[15]	DCS	✓	χ	✓	χ
[5]	DCS	✓	χ	✓	χ
[26]	DCS	✓	χ	✓	✓
[31]	DCS	χ	χ	✓	χ
[46]	DCS	✓	χ	✓	✓
[18]	ACS	✓	✓	✓	χ
[25]	ACS	✓	✓	✓	✓

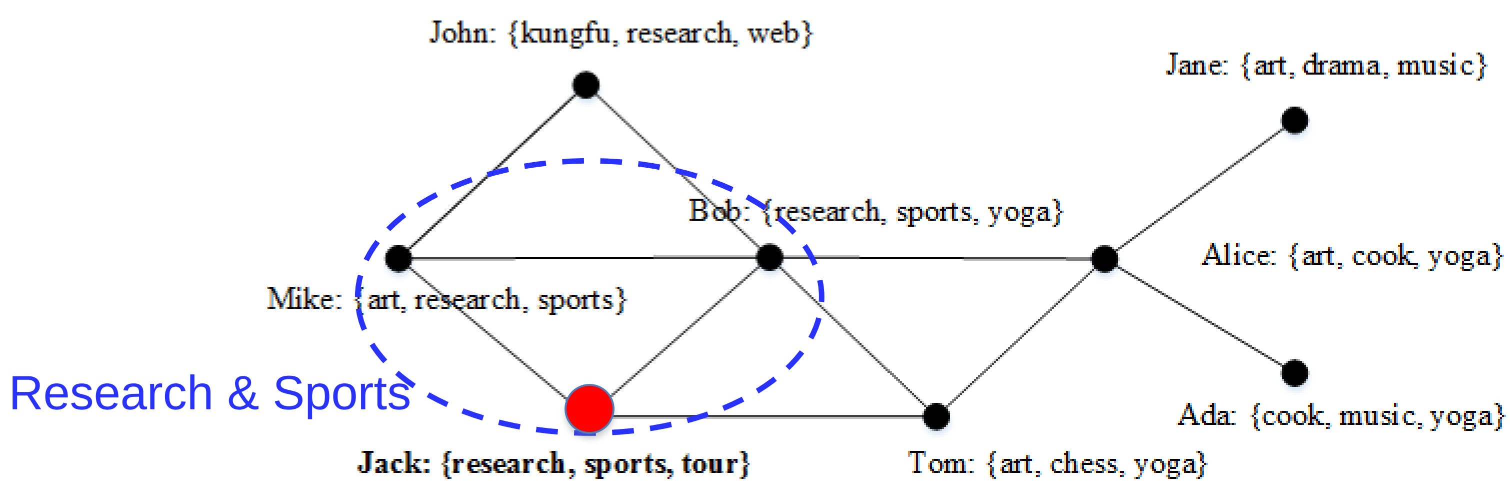
The Number of Related Works

Graph type	Community Detection	Community Search
Non-attributed	[1000+ papers]	[10+ papers]
Attributed	[100+ papers]	K-core-based: ACQ K-truss-based: ATC

[Fang et al. PVLDB'16].

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 ≥ *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}

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 e

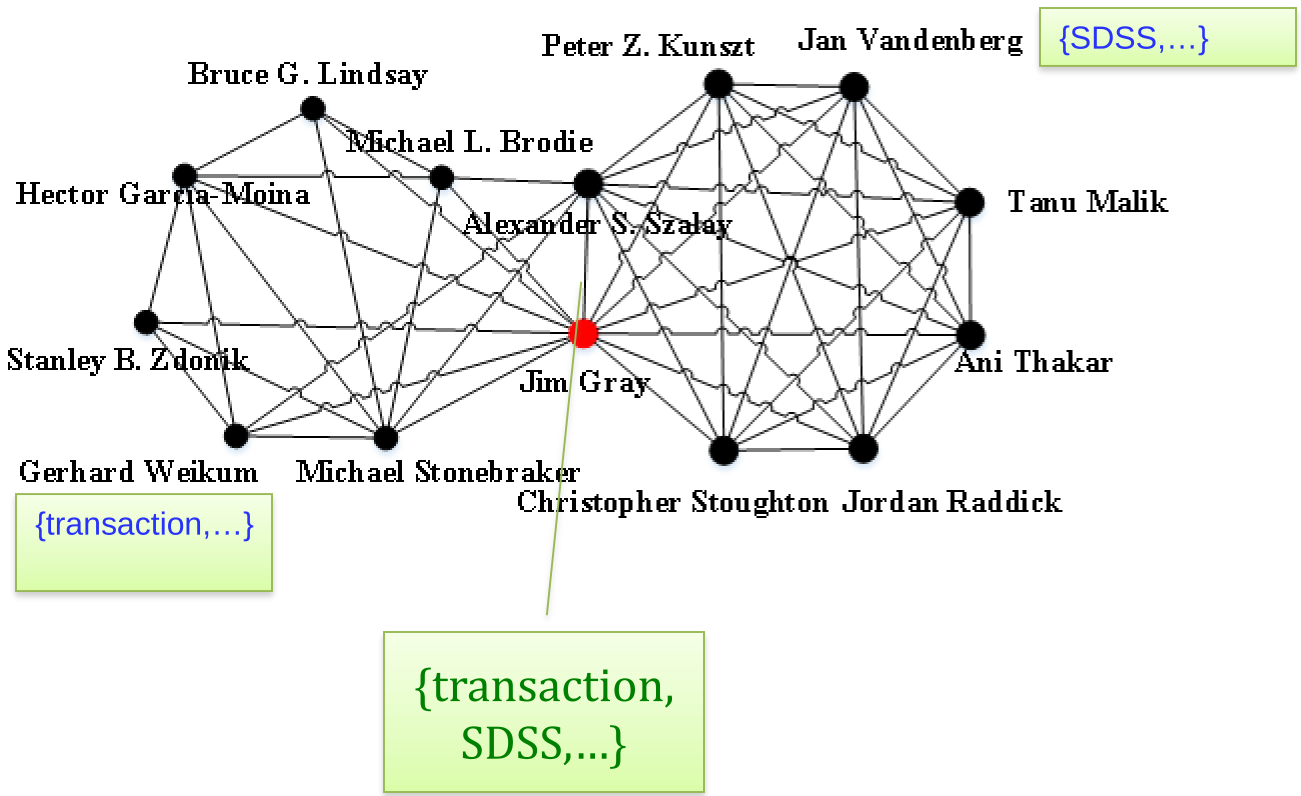
A community can have 10^5 nodes!

- **Why** are these people considered as Jim's community?
- What is the **theme** of this community?

[1] Sozio, Mauro, and Aristides Gionis. "The community-search problem and how to plan a successful cocktail party." *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010.
[2] Cui, Wanyun, et al. "Local search of communities in large graphs." *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014.

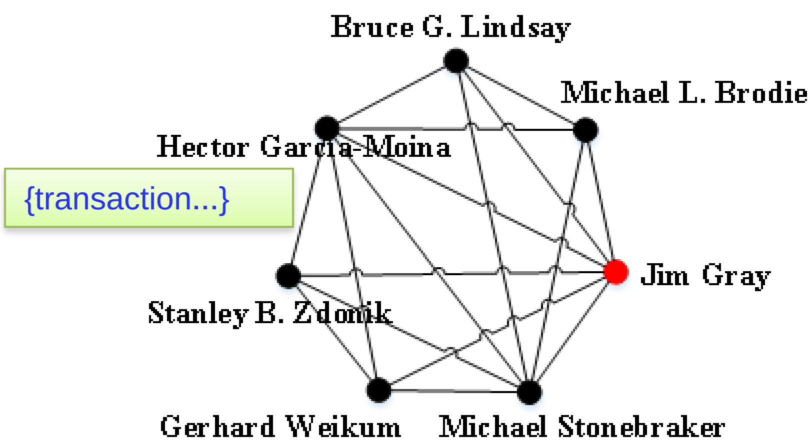
Attributed Community (AC)

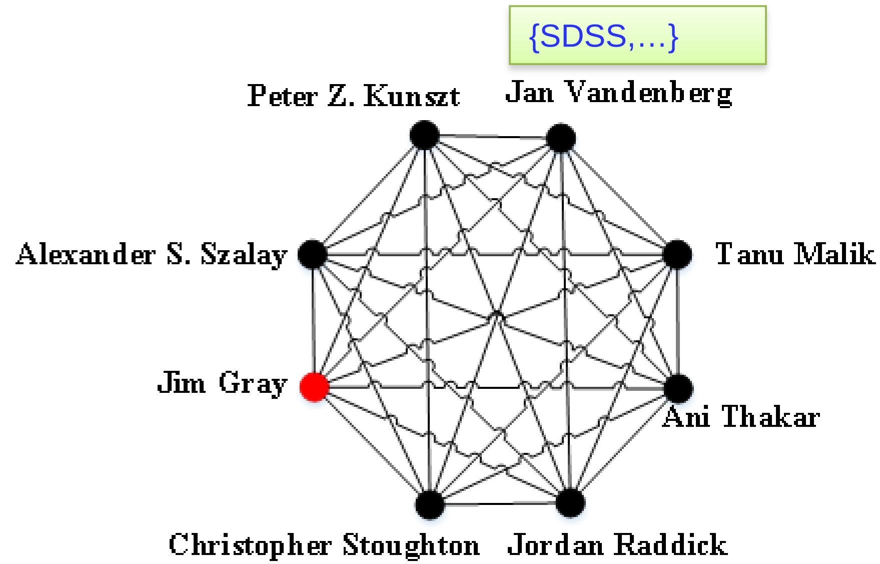
- Previous CS solutions overlook <u>keywords</u>
  - e.g., a researcher's interest



Attributed Community (AC)

- In fact, Jim has 2 distinct <u>attributed communities (AC)</u>.





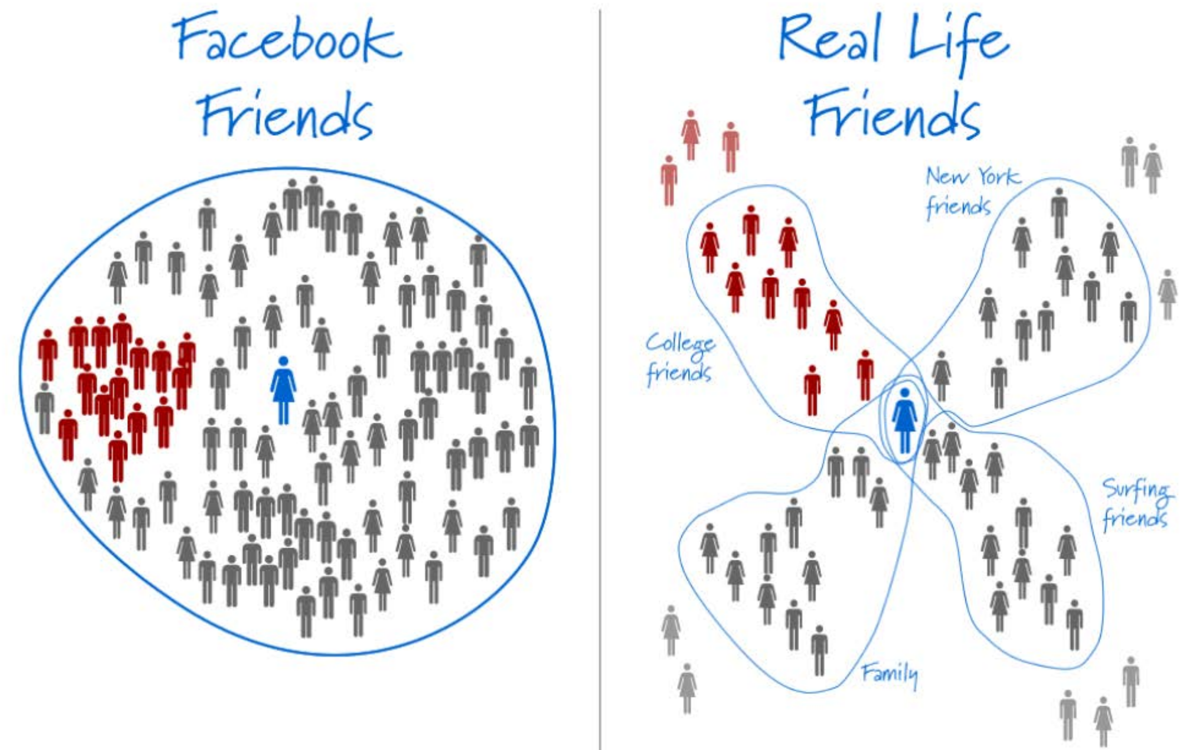
{research, transaction, data, management, system}

{sloan, digital, sky, data, sdss }

Common keyword set (AC label)

Part 3: Social Circle Discovery

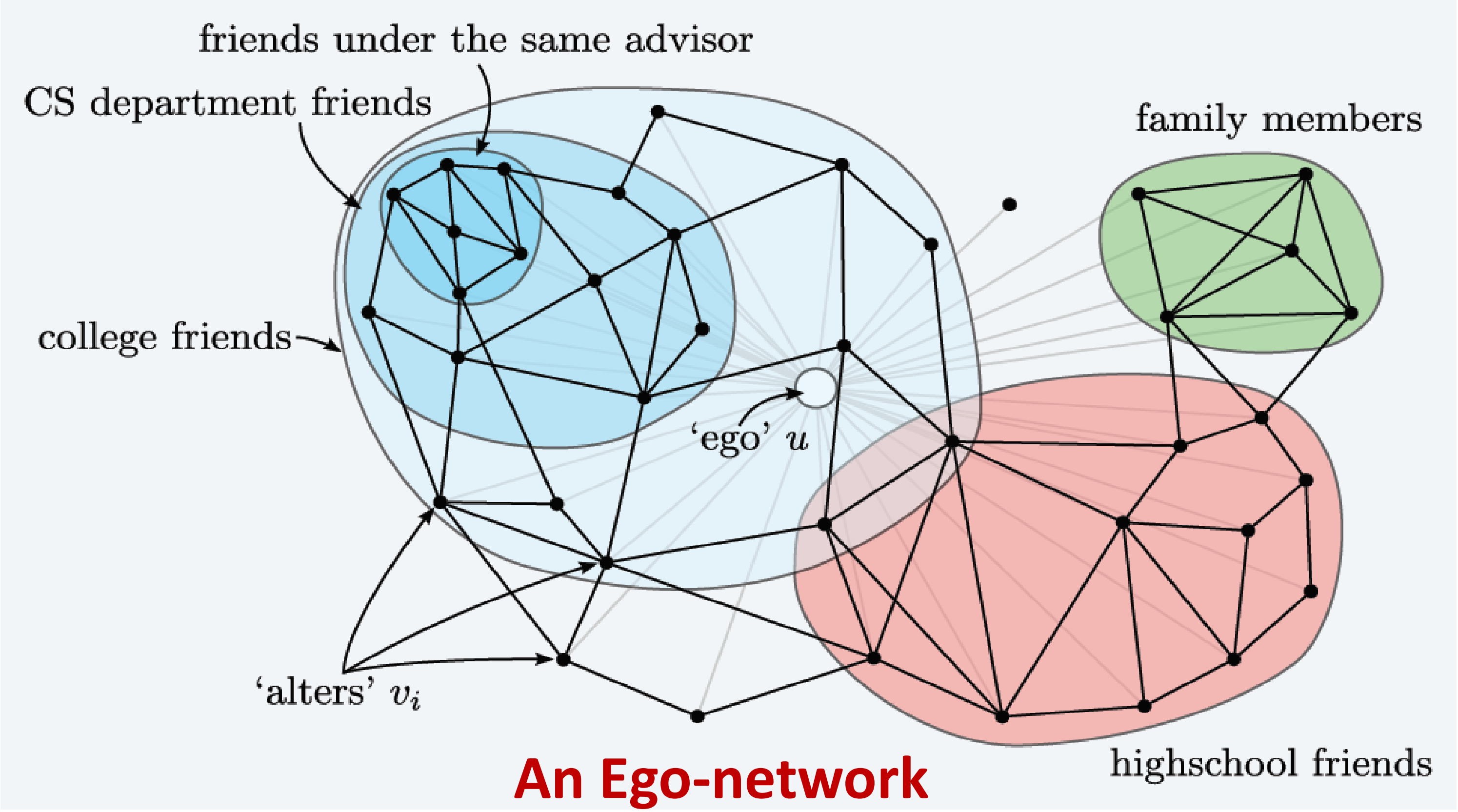
- Social circles: communities formed by only friends



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

[Leskovec and Mcauley, NIPS'12]

Learning to discover social circles

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

$$p((x,y) \in E) \propto \exp\left\{ \underbrace{\sum_{C_k \supseteq \{x,y\}} \langle \phi(x,y), \theta_k \rangle}_{\text{circles containing both nodes}} - \underbrace{\sum_{C_k \not\supseteq \{x,y\}} \alpha_k \langle \phi(x,y), \theta_k \rangle}_{\text{all other circles}} \right\}$$

Training is done by maximum likelihood, using QPBO and L-BFGS.

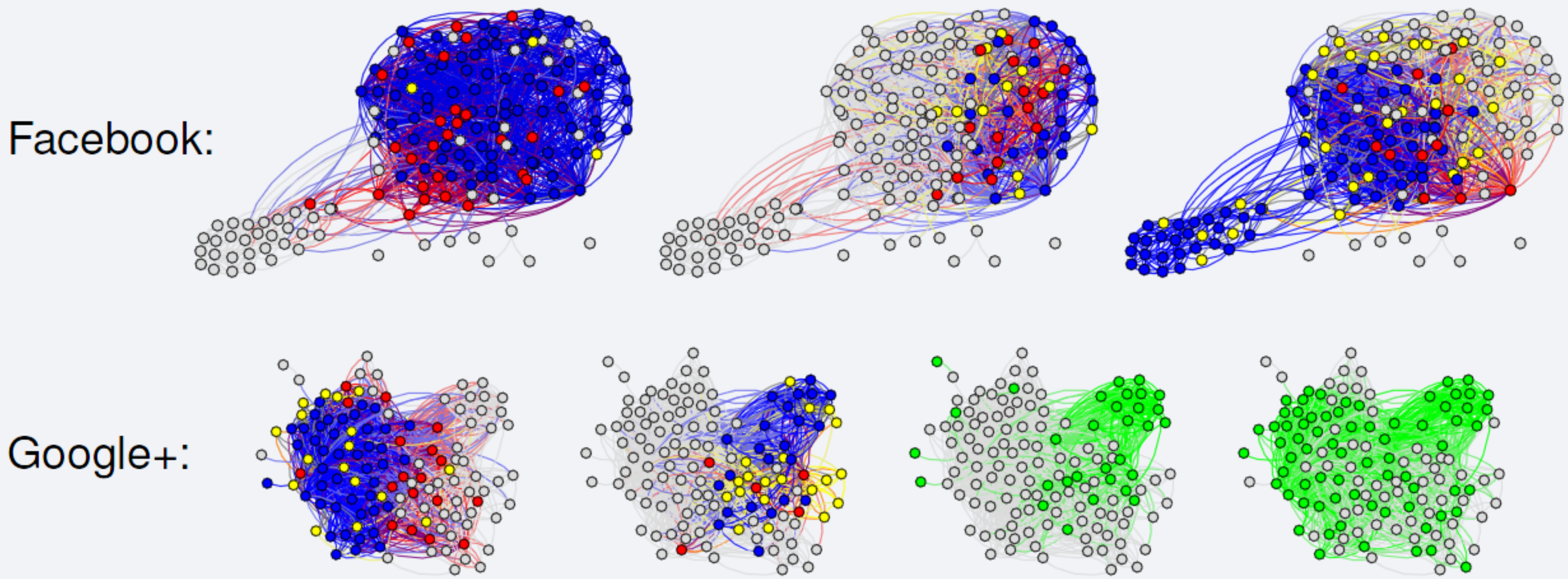
Datasets: Ground-truth Social Circles

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

	ego-networks	circles	nodes	edges
Facebook	10	193	4,039	88,234
Google+	133	479	107,614	13,673,453
Twitter	1,000	4,869	81,306	1,768,149

All data are **available** on `snap.stanford.edu/data/`

Detected Circles



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

[Ugander et al. PNAS’12].

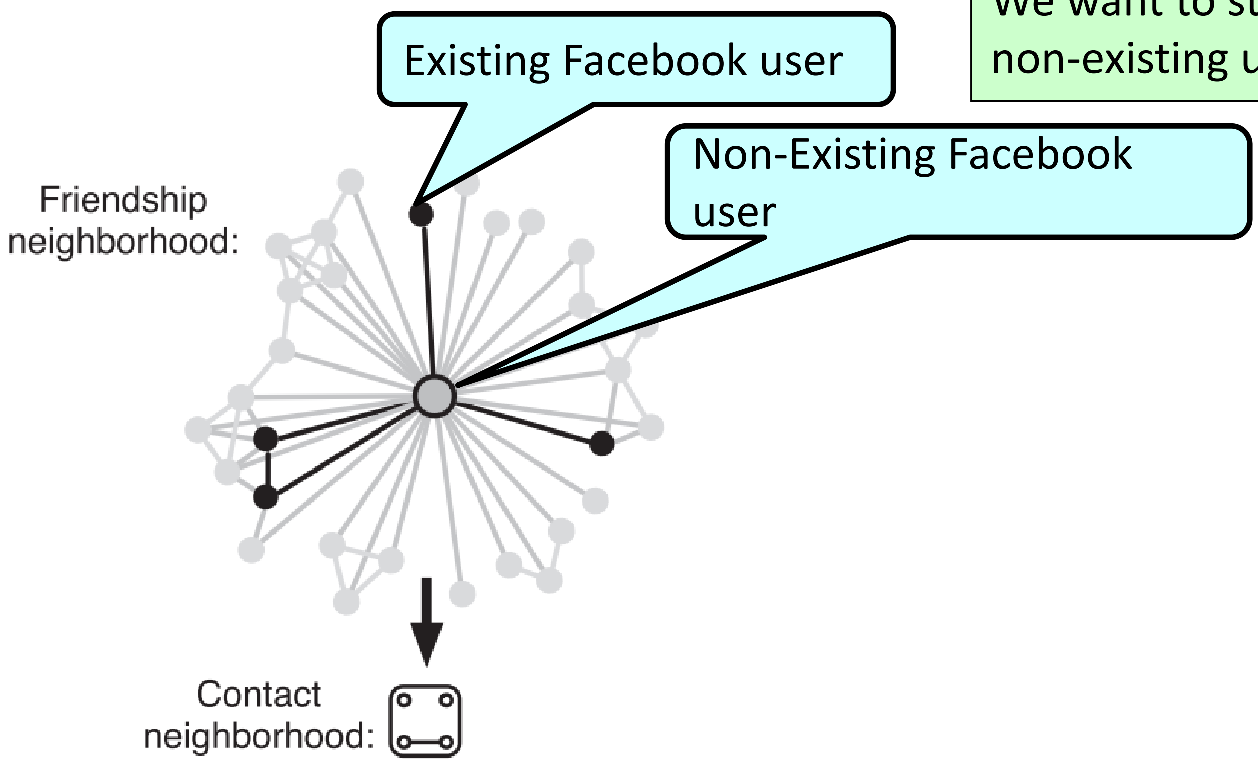
Social Contagion

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

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



Social Contagion

- Case Study (Facebook)

  [Ugander et al, PNAS'12]

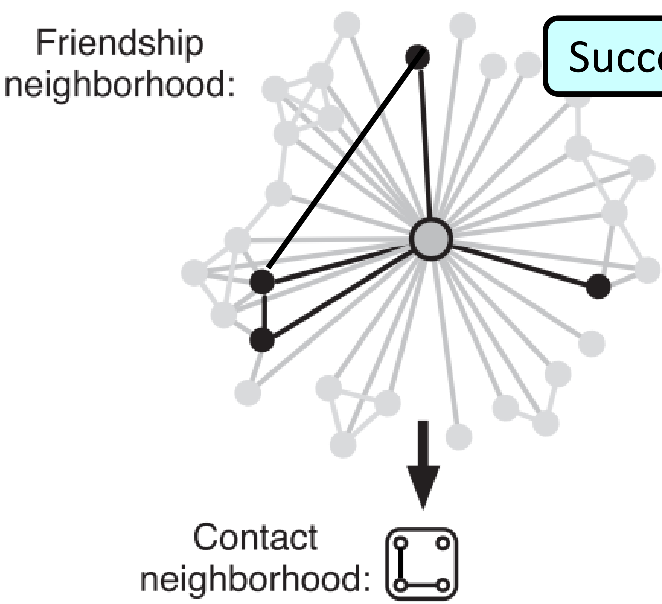
Social circles can affect the process of information diffusion on social contagion

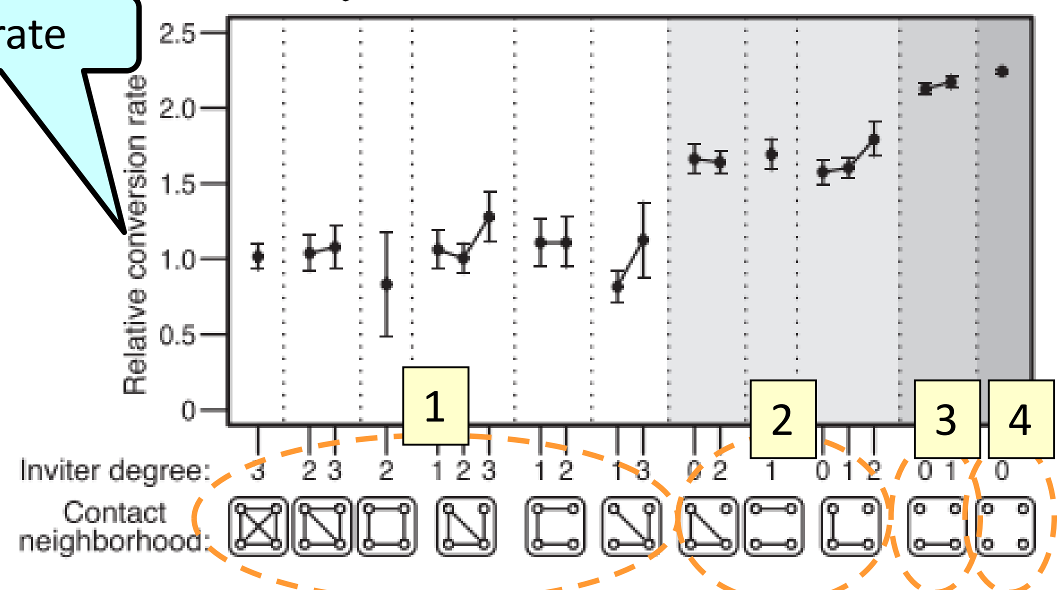
Consider an existing Facebook user invites the non-existing Facebook user to join Facebook.

is

The no. of connected components is related to the success rate

Success rate





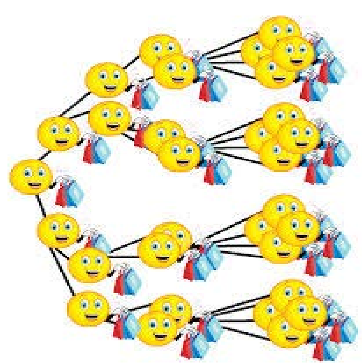
[Huang et al. VLDBJ'15].

Top-K Structural Diversity Search

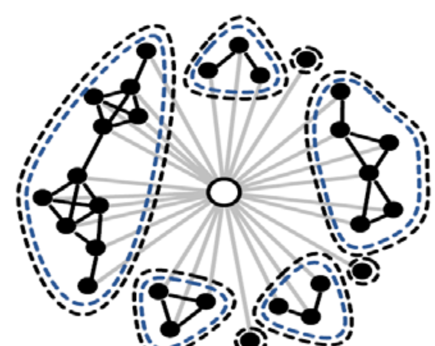
- The **structural diversity** of a node is defined to be **the number of connected components** in its ego-network.

- **Problem**
  - **Find k nodes** with the **greatest structural diversity** in a social network (Node Ranking).
- **Application**
  - Political campaign, promotion of health practices, marketing

Opinions Diffusion



Viral Marketing



Connected Components in the Neighborhood

[Zhu et al. ArXiv’14]

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…

UBER

facebook

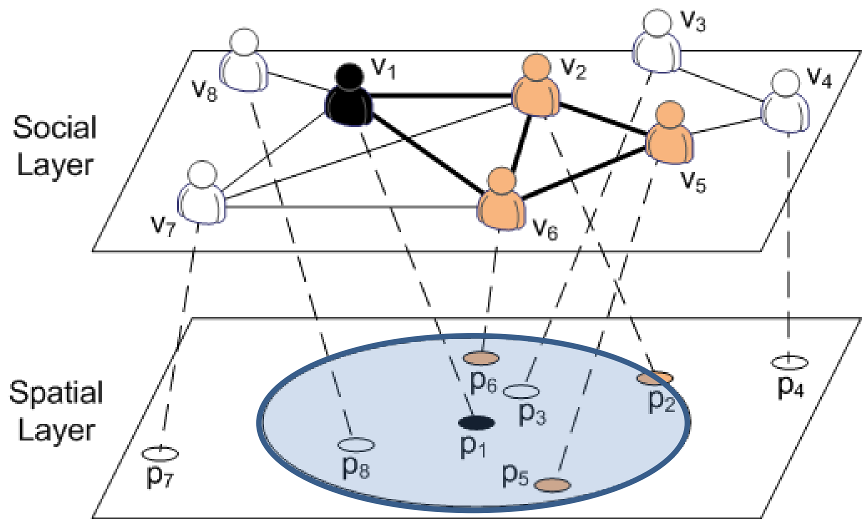
Facebook Places

Who. What. When. And now where.

[Zhu et al. ArXiv'14]

Geo-Social Group Queries (GSGQ)

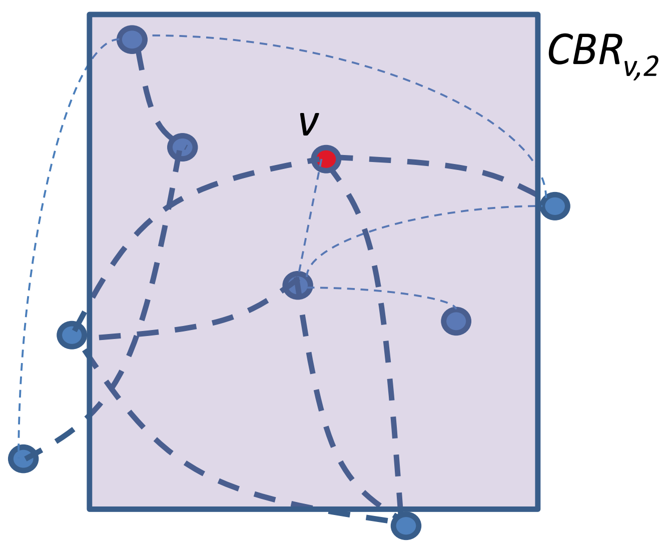
- Given an LBSN G=(V, E), a query user $v_q \in V$ and an integer $c \geq 1$, find a group of users $V' \subseteq V$ containing v_q and satisfying:
  - Social constraint: G[V'] $\in G$ is a c-core
  - Spatial constraint:
    - **Range**: all users of the group are in a given spatial range
    - ***k*NN**: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



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

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: $O(|E| \log |V|)$

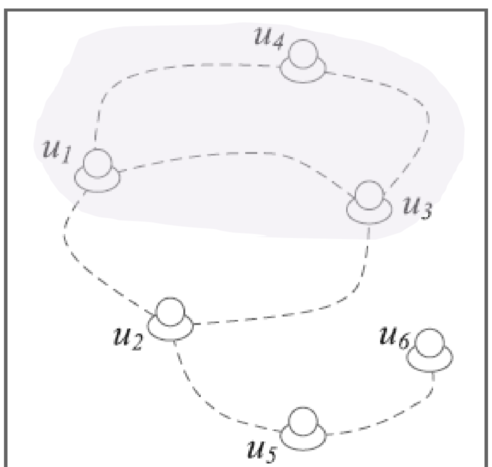


Pruning: exclude v from result group if query range $\subset$ CBR

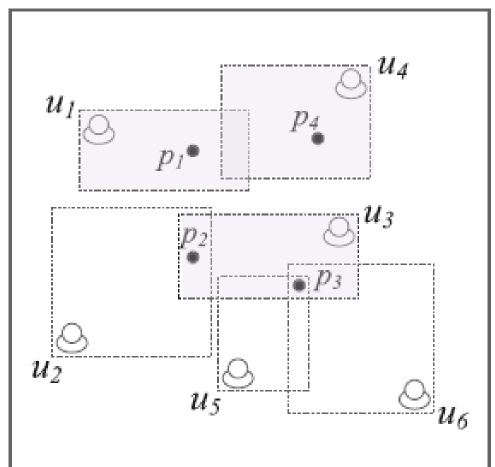
[Li et al. ICDE 2016]

Geo-Social K-Cover Group Queries

- **Problem:** Given an LBSN $G(V, E)$, a set of query points $P=\{p_1, p_2, ..., p_m\}$, and an integer $k \geq 1$, find a group of users $V' \subseteq V$ satisfying:
  1) Spatial constraint: $P \subset \cup_{u \in V'} u.R$
  2) Social constraint: $G[V'] \in G$ is a c-core
  3) Size requirement: $|V'|$ is minimum



(a) Social networks



(b) Associated regions

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

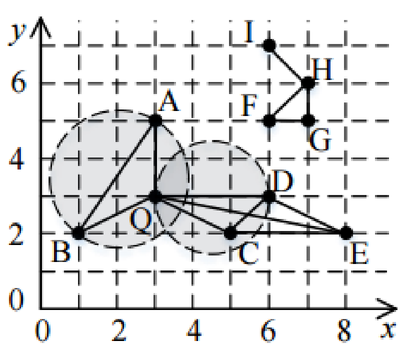
[Li et al. ICDE 2016]

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

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

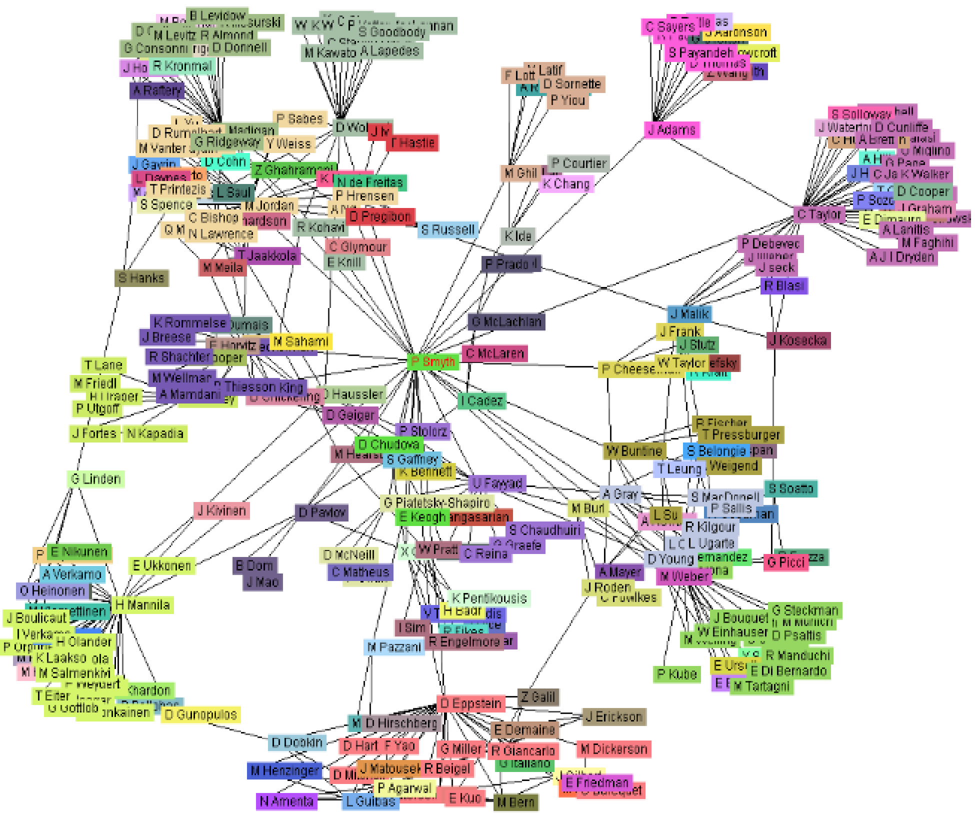
- **Heterogeneous Information Networks**
- **Scalability**
  - I/O-efficient algorithms & distributed computing
  - Stream graphs
- **Public-Private Social Networks**
- **Community Search on Uncertain Graphs**
  - Probabilistic k-core & Probabilistic k-truss

Heterogeneous Information Networks

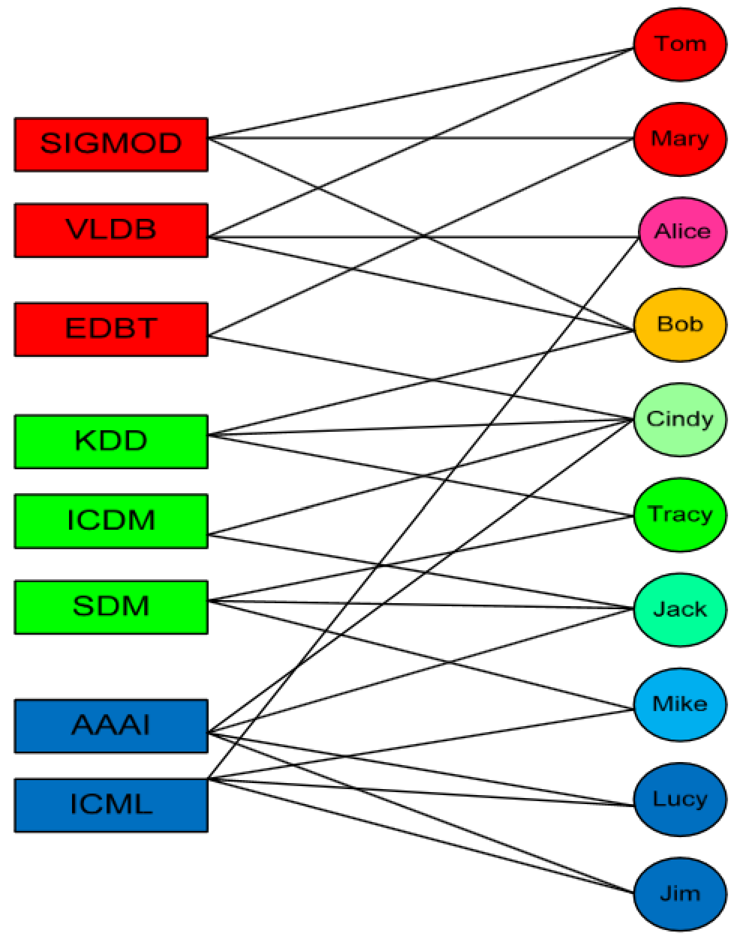
- Information network: A network where each node represents an entity (e.g., actor in a social network) and each link (e.g., tie) a relationship between entities.

- 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



Co-author Network



Conference-Author Network

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

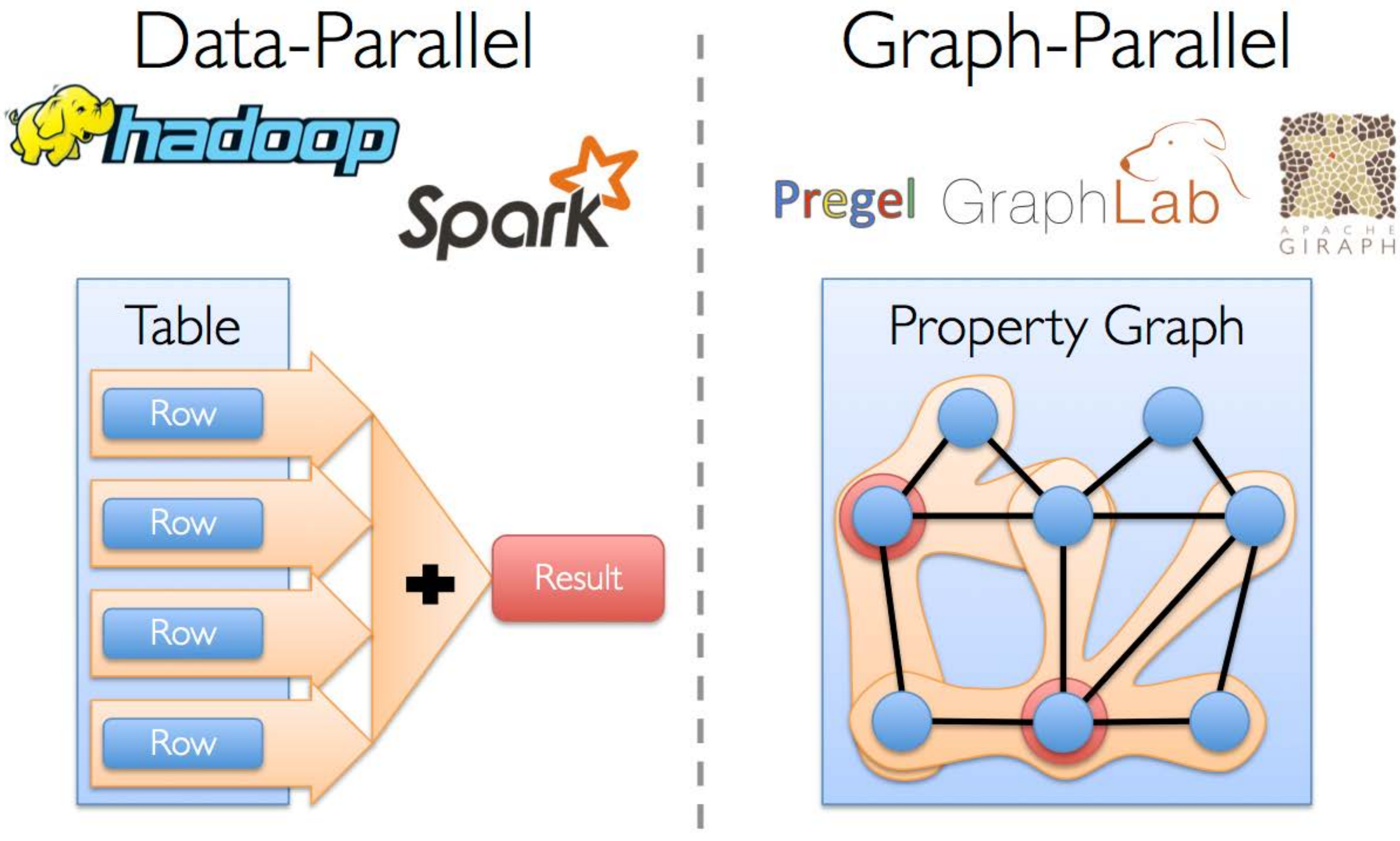
- **Scaling community search techniques** to the *massive and rapidly growing network datasets* of the Big Data era.

- **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



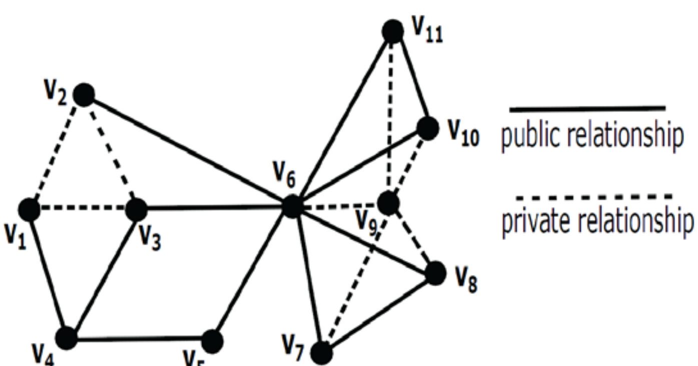
Open Problems & Future Directions

- **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

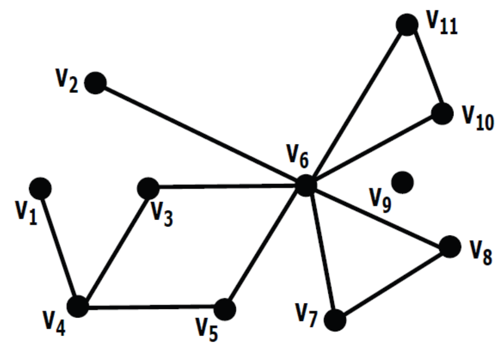
Public-Private Social Networks

- ***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)

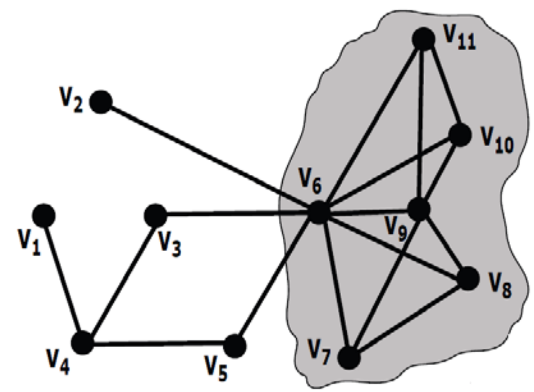
- Public-Private graph model contains a public graph, in which **each node** is also associated with **a private graph**.
  – The public graph is visible to everyone, but each **private graph** is **visible only to the corresponding user**.



A Public-private Graph



Public Graph



In the view of $v9$: $G \cup G_{v_9}$

Open Problems & Future Directions

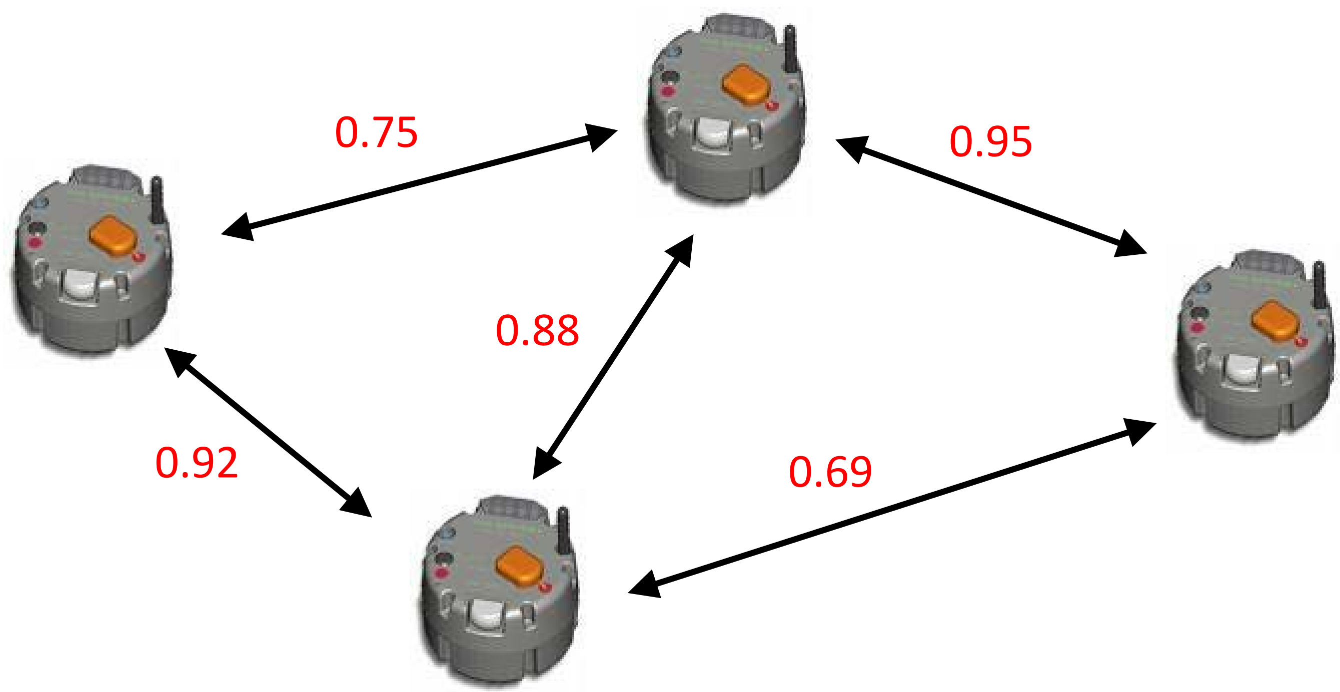
- **Heterogeneous Information Networks**
- **Scalability**
  - I/O-efficient algorithms & distributed computing
  - Stream graphs
- **Public-Private Social Networks**
- **Community Search on Uncertain Graphs**
  - Probabilistic k-core & Probabilistic k-truss

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 I

[1] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: A system for keyword-based search over relational databases. In ICDE, pages 5–16, 2002.

[2] N. Armenatzoglou, R. Ahuja, and D. Papadias. Geo-social ranking: functions and query processing. The VLDB Journal, 24(6):783–799, 2015.

[3] N. Armenatzoglou, S. Papadopoulos, and D. Papadias. A general framework for geo-social query processing. PVLDB, 6(10):913–924, 2013.

[4] B. Bahmani, R. Kumar, and S. Vassilvitskii. Densest subgraph in streaming and mapreduce. PVLDB, 5(5):454–465, 2012.

[5] N. Barbieri, F. Bonchi, E. Galimberti, and F. Gullo. Efficient and effective community search. DMKD, 29(5):1406–1433, 2015.

[6] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using banks. In ICDE, pages 431–440, 2002.

[7] P. Boldi, M. Rosa, and S. Vigna. Hyperanf: approximating the neighbourhood function of very large graphs on a budget. In WWW, pages 625–634, 2011.

[8] F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich. Core decomposition of uncertain graphs. In KDD, pages 1316–1325, 2014.

[9] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova. Clustering attributed graphs: models, measures and methods. Network Science, 3(03):408–444, 2015.

[10] H. Cheng, Y. Zhou, X. Huang, and J. X. Yu. Clustering large attributed information networks: an efficient incremental computing approach. DMKD, 25(3):450–477, 2012.

References II

[11] J. Cheng, Y. Ke, S. Chu, and M. T. O ¨ zsu. Efficient core decomposition in massive networks. In ICDE, pages 51–62, 2011.

[[12] F. Chierichetti, A. Epasto, R. Kumar, S. Lattanzi, and V. Mirrokni. Efficient algorithms for public-private social networks. In KDD, pages 139–148, 2015.

[13] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. JCSS, 55(3):441–453, 1997.

[14] W. Cui, Y. Xiao, H. Wang, Y. Lu, and W. Wang. Online search of overlapping communities. In SIGMOD, pages 277–288, 2013.

[15] W. Cui, Y. Xiao, H. Wang, and W. Wang. Local search of communities in large graphs. In SIGMOD, pages 991–1002, 2014.

[16] R. Dey, Z. Jelveh, and K. Ross. Facebook users have become much more private: A large-scale study. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, pages 346–352. IEEE, 2012.

[17] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin. Finding top-k min-cost connected trees in databases. In ICDE, pages 836–845, 2007.

[18] Y. Fang, R. Cheng, S. Luo, and J. Hu. Effective community search for large attributed graphs. PVLDB, 9(12):1233–1244, 2016.

[19] A. Gajewar and A. D. Sarma. Multi-skill collaborative teams based on densest subgraphs. In SDM, pages 165–176, 2012.

[20] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In PVLDB, pages 670–681, 2002.

References III

[21] X. Huang, H. Cheng, R.-H. Li, L. Qin, and J. X. Yu. Top-k structural diversity search in large networks. PVLDB, 6(13):1618–1629, 2013.

[22] X. Huang, H. Cheng, R.-H. Li, L. Qin, and J. X. Yu. Top-k structural diversity search in large networks. The VLDB Journal, 24(3):319–343, 2015.

[23] X. Huang, H. Cheng, L. Qin, W. Tian, and J. X. Yu. Querying k-truss community in large and dynamic graphs. In SIGMOD, pages 1311–1322, 2014.

[24] X. Huang, H. Cheng, and J. X. Yu. Dense community detection in multivalued attributed networks. Information Sciences, 314:77–99, 2015.

[25] X. Huang and L. V. Lakshmanan. Attribute truss community search. arXiv preprint arXiv:1609.00090, 2016.

[26] X. Huang, L. V. Lakshmanan, J. X. Yu, and H. Cheng. Approximate closest community search in networks. PVLDB, 9(4):276–287, 2015.

[27] X. Huang, W. Lu, and L. V. S. Lakshmanan. Truss decomposition of probabilistic graphs: Semantics and algorithms. In SIGMOD, pages 77–90, 2016.

[28] M. Kargar and A. An. Discovering top-k teams of experts with/without a leader in social networks. In CIKM, pages 985–994, 2011.

[29] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In KDD, pages 467–476, 2009.

[30] G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou. Ease: an effective 3-in-1 keyword search method for unstructured, semi-structured and structured data. In SIGMOD, pages 903–914, 2008.

References IV

[31] R.-H. Li, L. Qin, J. X. Yu, and R. Mao. Influential community search in large networks. PVLDB, 8(5), 2015.

[32] Y. Li, R. Chen, J. Xu, Q. Huang, H. Hu, and B. Choi. Geo-social k-cover group queries for collaborative spatial computing. TKDE, 27(10):2729–2742, 2015.

[33] Y. Li, R. Chen, J. Xu, Q. Huang, H. Hu, and B. Choi. Geo-social k-cover group queries for collaborative spatial computing. In ICDE, pages 1510–1511, 2016.

[34] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In SIGMOD, pages 135–146, 2010.

[35] J. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In NIPS, volume 272, pages 548–556, 2012.

[36] J. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In NIPS, pages 548–556, 2012.

[37] L. Qin, J. X. Yu, L. Chang, and Y. Tao. Querying communities in relational databases. In ICDE, pages 724–735, 2009.

[38] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In WWW, pages 1089–1098,2013.

[39] M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In KDD, pages 939–948, 2010.

[40] Y. Sun and J. Han. Mining heterogeneous information networks: a structural analysis approach. ACM SIGKDD Explorations Newsletter, 14(2):20–28, 2013.

References V

[41] Y. Sun, J. Tang, J. Han, M. Gupta, and B. Zhao. Community evolution detection in dynamic heterogeneous information networks. In Proceedings of the Eighth Workshop on Mining and Learning with Graphs, pages 137–146, 2010.

[42] J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. PNAS, (16):5962–5966.

[43] J. Wang and J. Cheng. Truss decomposition in massive networks. PVLDB, 5(9):812–823, 2012.

[44] Y. Wang and L. Gao. An edge-based clustering algorithm to detect social circles in ego networks. Journal of computers, 8(10):2575–2582, 2013.

[45] D. Wen, L. Qin, Y. Zhang, X. Lin, and J. X. Yu. I/O efficient core graph decomposition at web scale. In ICDE, pages 133–144, 2016.

[46] Y. Wu, R. Jin, J. Li, and X. Zhang. Robust local community detection: On free rider effect and its elimination. PVLDB, 8(7), 2015.

[47] D. Yan, J. Cheng, Y. Lu, and W. Ng. Blogel: A block-centric framework for distributed computation on real-world graphs. PVLDB, 7(14):1981–1992, 2014.

[48] D.-N. Yang, C.-Y. Shen, W.-C. Lee, and M.-S. Chen. On socio-spatial group query for location-based social networks. In KDD, pages 949–957, 2012.

[49] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. PVLDB, 2(1):718–729, 2009.

[50] Q. Zhu, H. Hu, J. Xu, and W.-C. Lee. Geo-social group queries with minimum acquaintance constraint. arXiv preprint arXiv:1406.7367, 2014.

Thank you!

Questions?

xinhuang@comp.hkbu.edu.hk