Mining Large Graphs

ECML/PKDD 2007 tutorial

Part 3: Case studies

Jure Leskovec and Christos Faloutsos

Machine Learning Department

Carnegie Mellon

Joint work with: Lada Adamic, Deepay Chakrabarti, Natalie Glance, Carlos Guestrin, Bernardo Huberman, Jon Kleinberg, Andreas Krause, Mary McGlohon, Ajit Singh, and Jeanne VanBriesen.

Tutorial outline

- Part 1: Structure and models for networks
  - What are properties of large graphs?
  - How do we model them?
- Part 2: Dynamics of networks
  - Diffusion and cascading behavior
  - How do viruses and information propagate?
- Part 3: Case studies
  - 240 million MSN instant messenger network
  - Graph projections: how does the web look like

Part 3: Outline

Case studies

- Microsoft Instant Messenger Communication network
  - How does the world communicate
- Web projections
  - How to do learning from contextual subgraphs
- Finding fraudsters on eBay
- Center piece subgraphs
  - How to find best path between the query nodes

Microsoft Instant Messenger Communication Network

How does the whole world communicate?

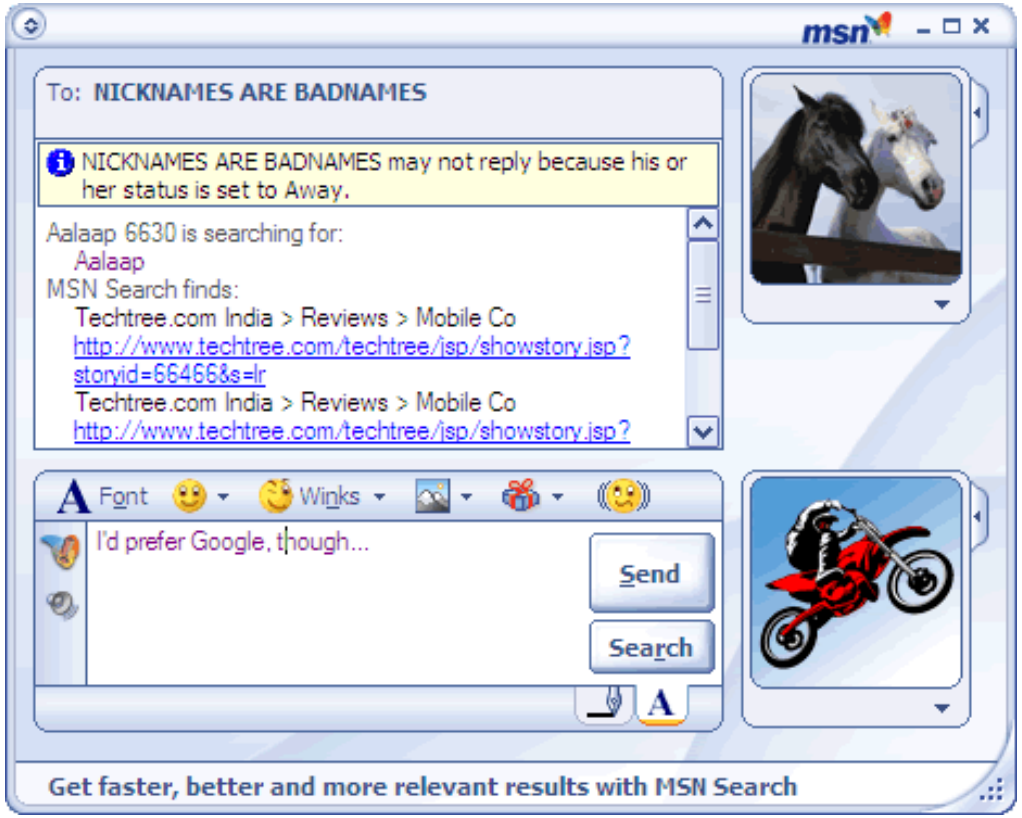
Leskovec and Horvitz: Worldwide Buzz: Planetary-Scale Views on an Instant-Messaging Network, 2007

The Largest Social Network

- What is the largest social network in the world (that one can relatively easily obtain)? ☺

  For the first time we had a chance to look at complete (anonymized) communication of the whole planet (using Microsoft MSN instant messenger network)

Instant Messaging



- Contact (buddy) list
- Messaging window

IM – Phenomena at planetary scale

Observe social phenomena at planetary scale:

- How does communication change with user demographics (distance, age, sex)?
- How does geography affect communication?
- What is the structure of the communication network?

Communication data

The record of communication

- Presence data
  - user status events (login, status change)
- Communication data
  - who talks to whom
- Demographics data
  - user age, sex, location

Data description: Presence

- Events:
  - Login, Logout
  - Is this first ever login
  - Add/Remove/Block buddy
  - Add unregistered buddy (invite new user)
  - Change of status (busy, away, BRB, Idle,…)
- For each event:
  - User Id
  - Time

Data description: Communication

- For every conversation (session) we have a list of users who participated in the conversation
- There can be multiple people per conversation
- For each conversation and each user:
  - User Id
  - Time Joined
  - Time Left
  - Number of Messages Sent
  - Number of Messages Received

Data description: Demographics

- For every user (self reported):
  - Age
  - Gender
  - Location (Country, ZIP)
  - Language
  - IP address (we can do reverse geo IP lookup)

Data collection

- Log size: 150Gb/day
- Just copying over the network takes 8 to 10h
- Parsing and processing takes another 4 to 6h
- After parsing and compressing ~ 45 Gb/day
- Collected data for 30 days of June 2006:
  - Total: 1.3Tb of compressed data

Data statistics

Activity over June 2006 (30 days)

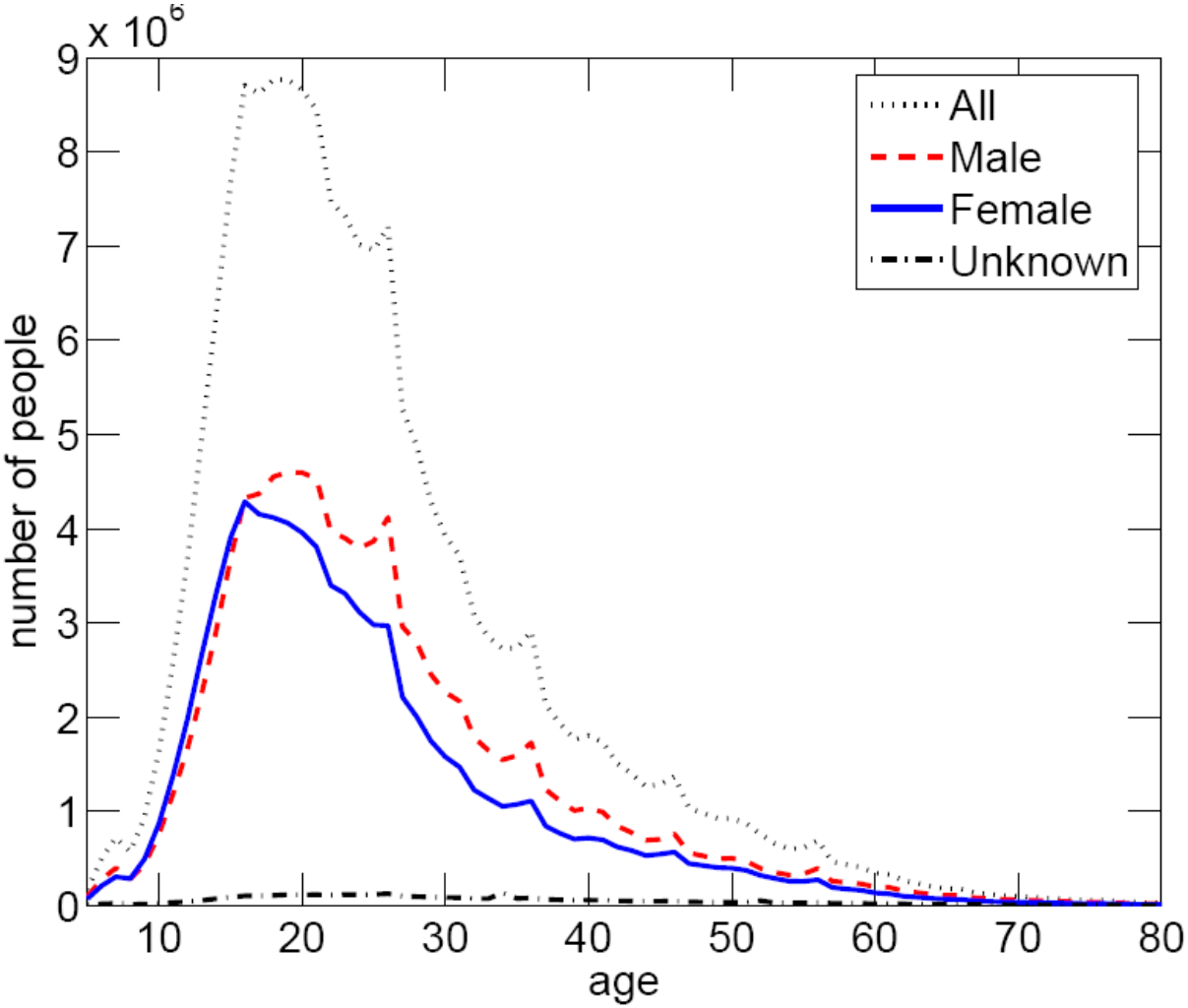
- 245 million users logged in
- 180 million users engaged in conversations
- 17,5 million new accounts activated
- More than 30 billion conversations

Data statistics per day

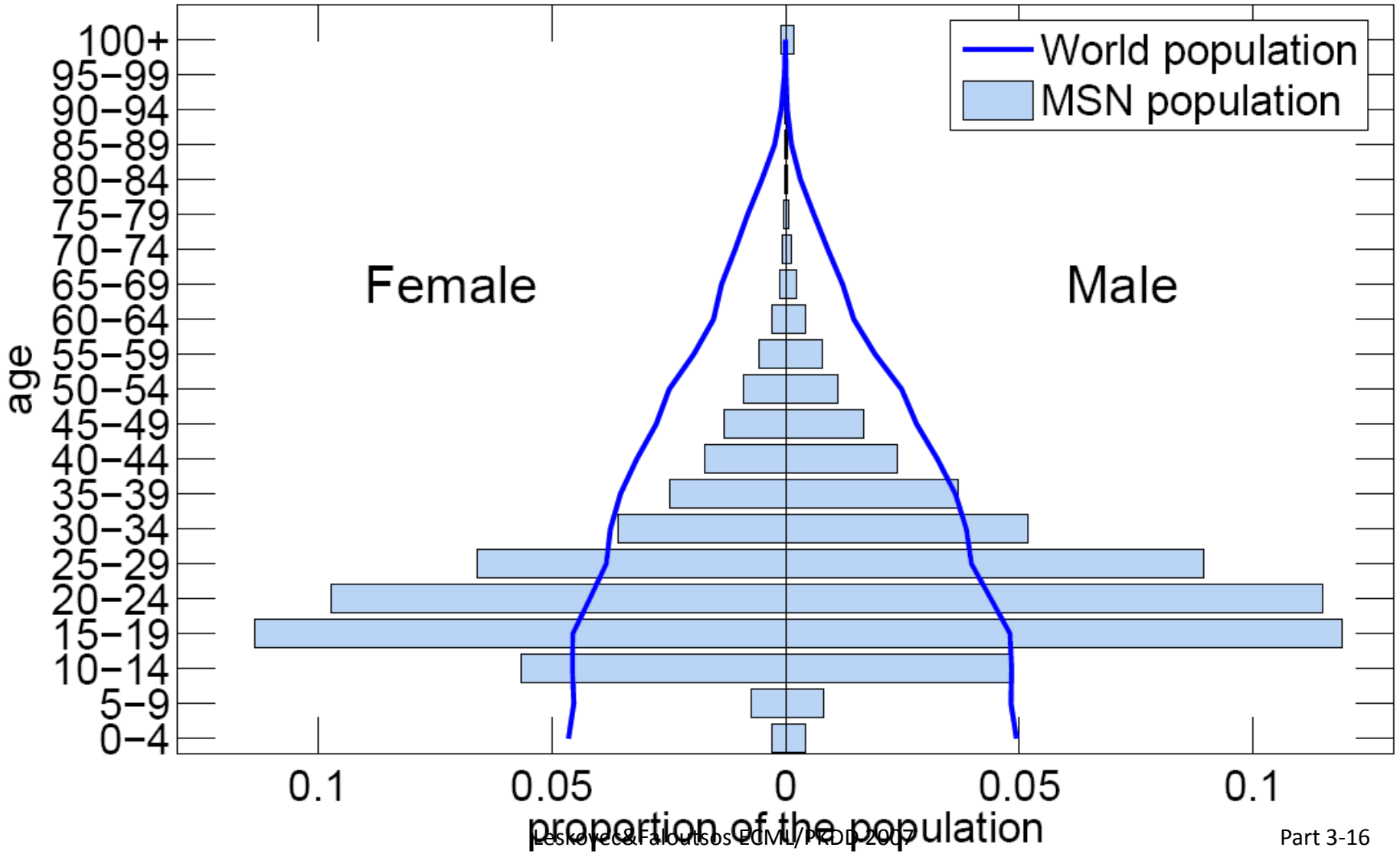
Activity on June 1 2006

- 1 billion conversations
- 93 million users login
- 65 million different users talk (exchange messages)
- 1.5 million invitations for new accounts sent

User characteristics: age



Age piramid: MSN vs. the world



Conversation: Who talks to whom?

- Cross gender edges:
  - 300 male-male and 235 female-female edges
  - 640 million female-male edges

	Unknown	Female	Male
Unknown	1.3	3.6	3.7
Female		21.3	49.9
Male			20.2

(a) Proportion of conversations

	Unknown	Female	Male
Unknown	277	301	277
Female		275	304
Male			252

(b) Conversation duration (seconds)

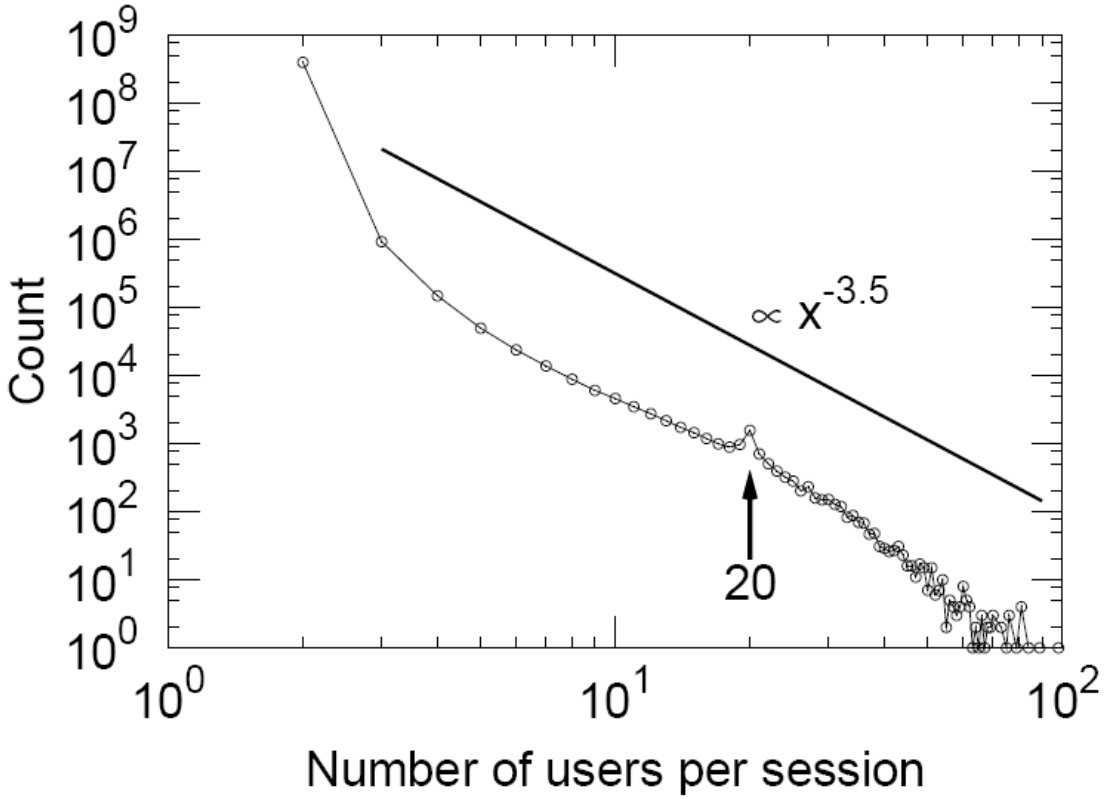
	Unknown	Female	Male
Unknown	5.7	7.1	6.7
Female		6.6	7.6
Male			5.9

(c) Exchanged messages per conversation

	Unknown	Female	Male
Unknown	1.25	1.42	1.38
Female		1.43	1.50
Male			1.42

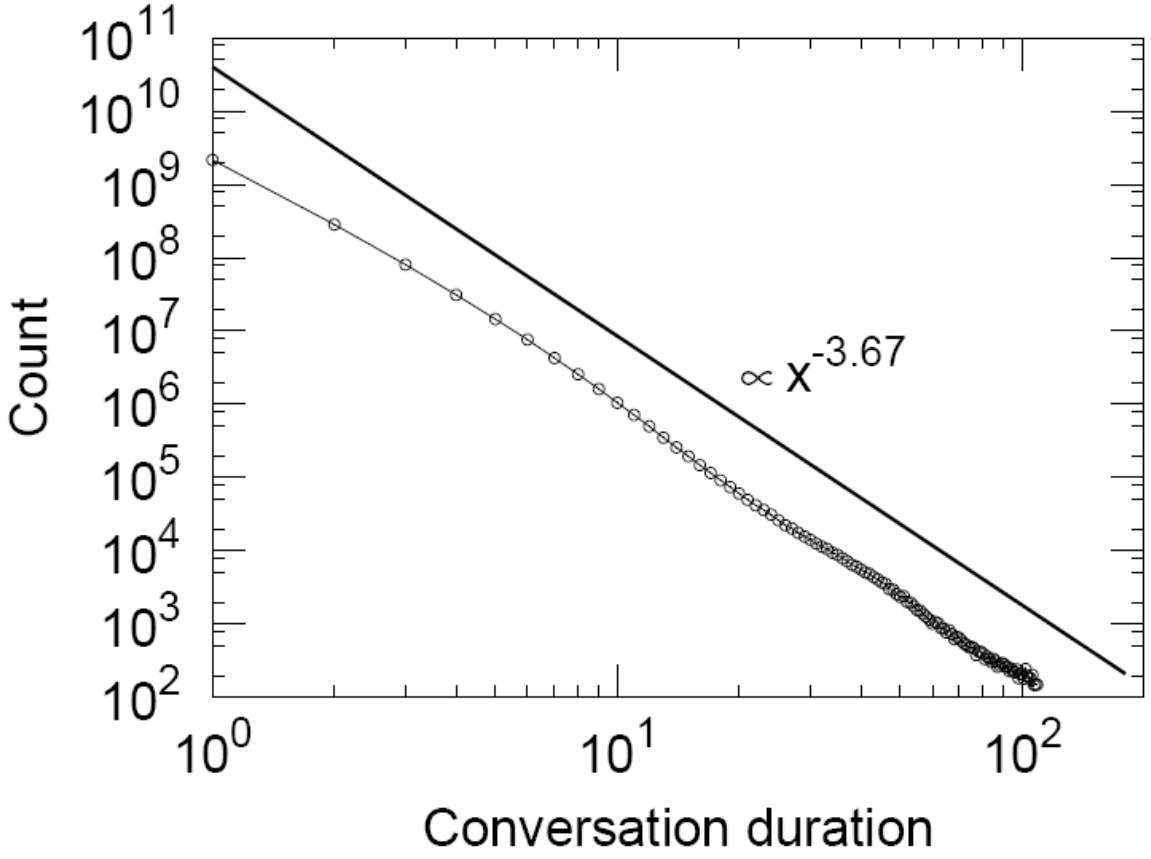
(d) Exchanged messages per minute

Number of people per conversation



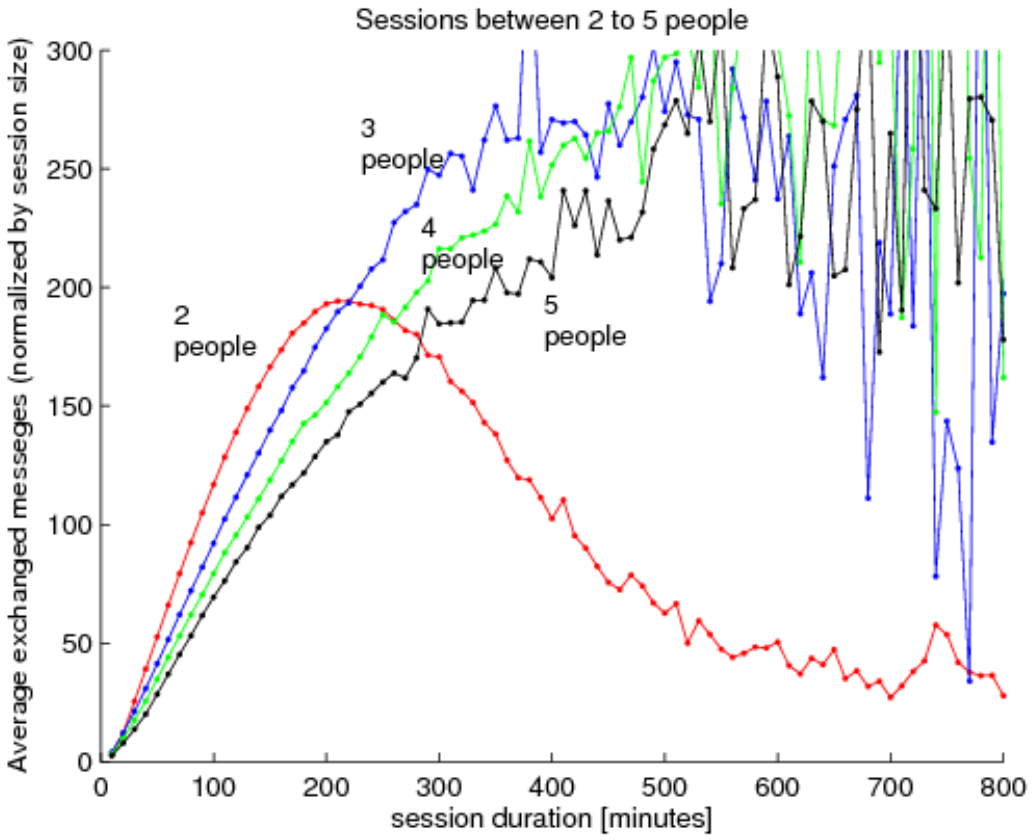
- Max number of people simultaneously talking is 20, but conversation can have more people

Conversation duration



- Most conversations are short

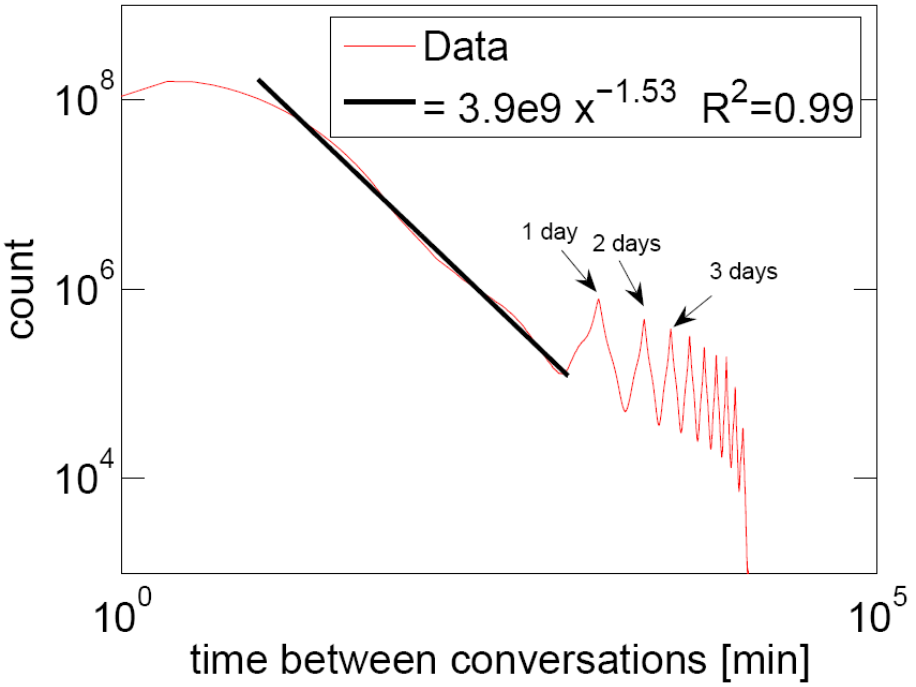
Conversations: number of messages



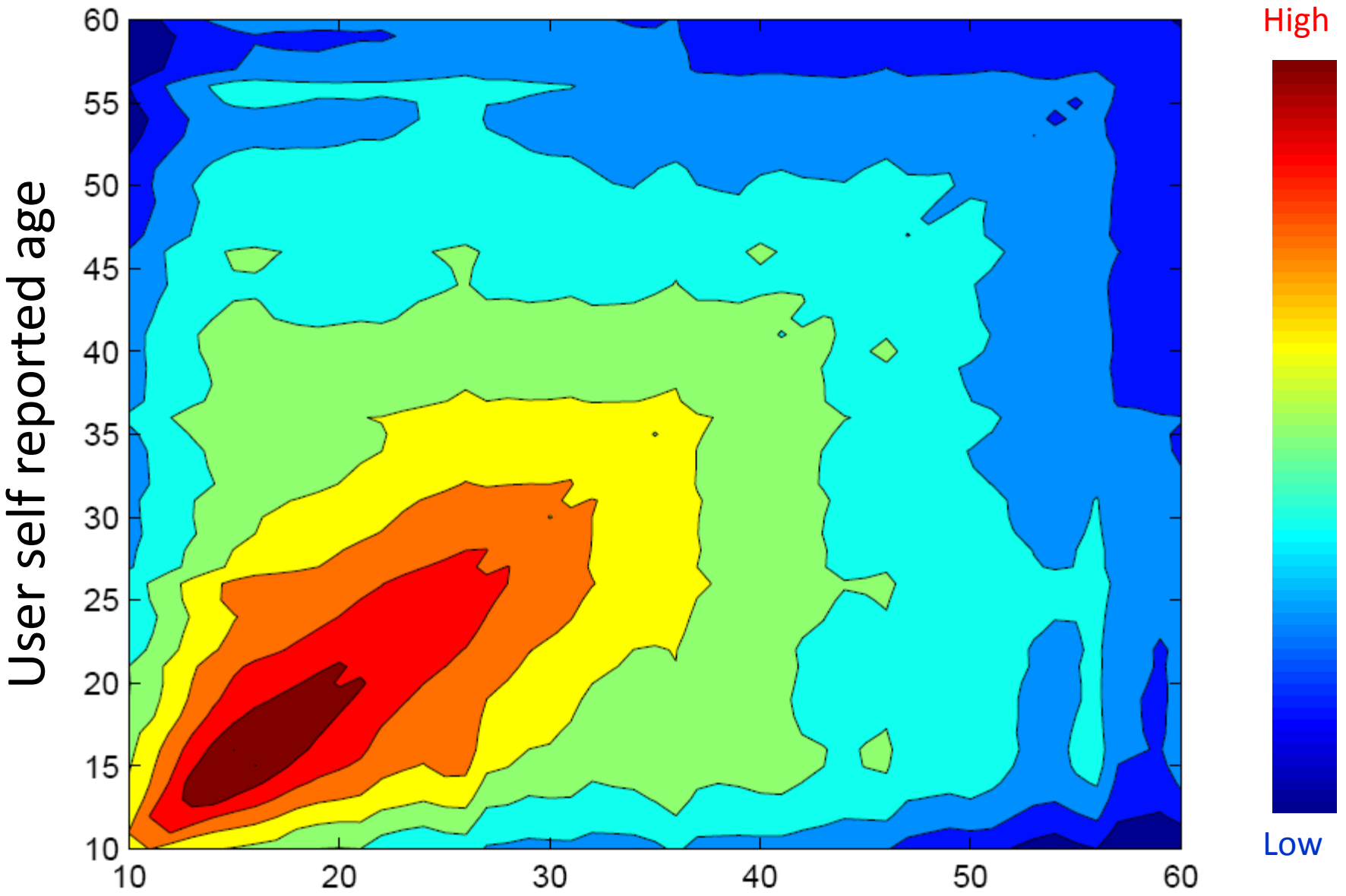
- Sessions between fewer people run out of steam

Time between conversations

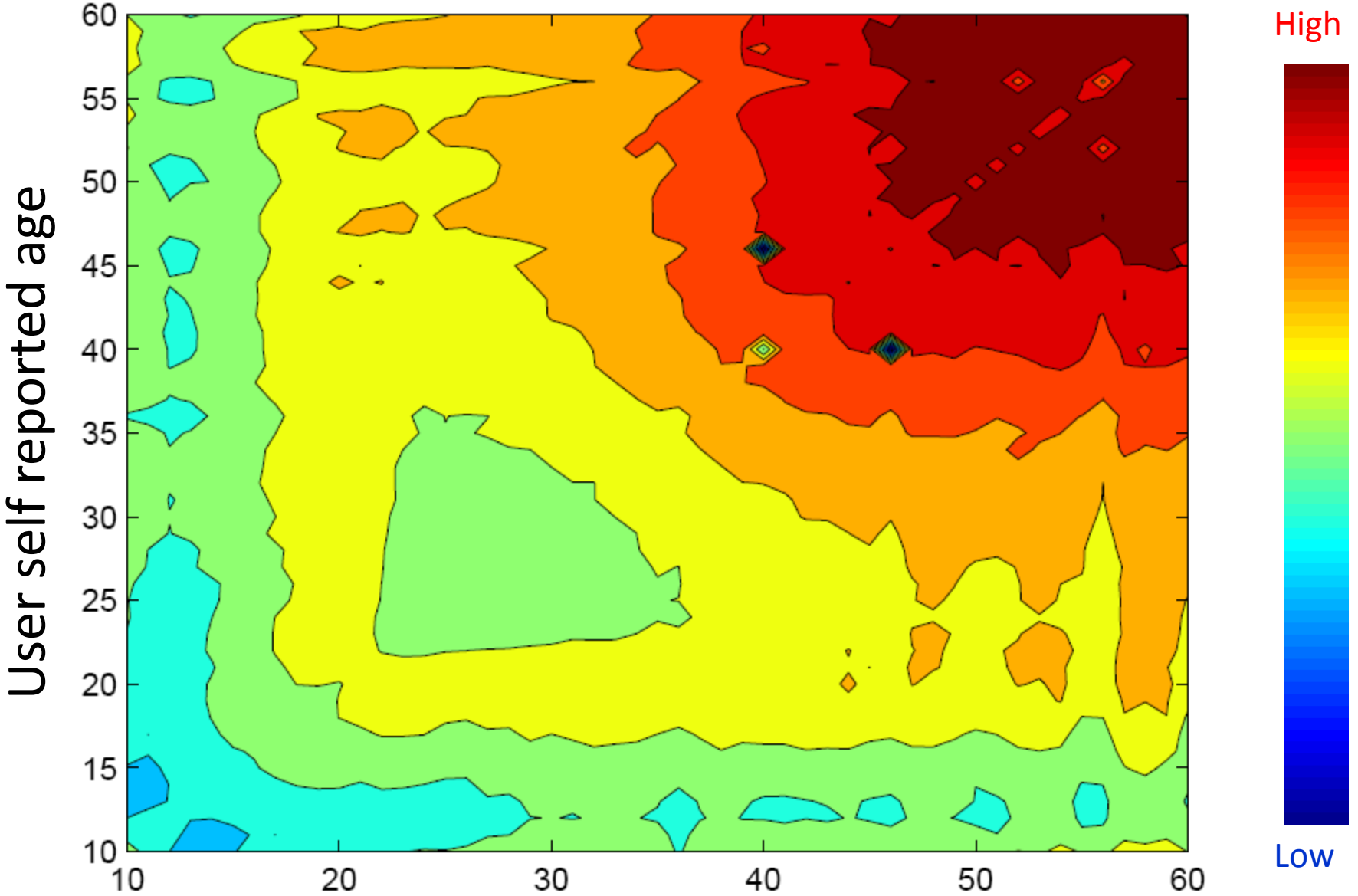
- Individuals are highly diverse
- What is probability to login into the system after *t* minutes?
- Power-law with exponent 1.5
- Task queuing model [Barabasi '05]



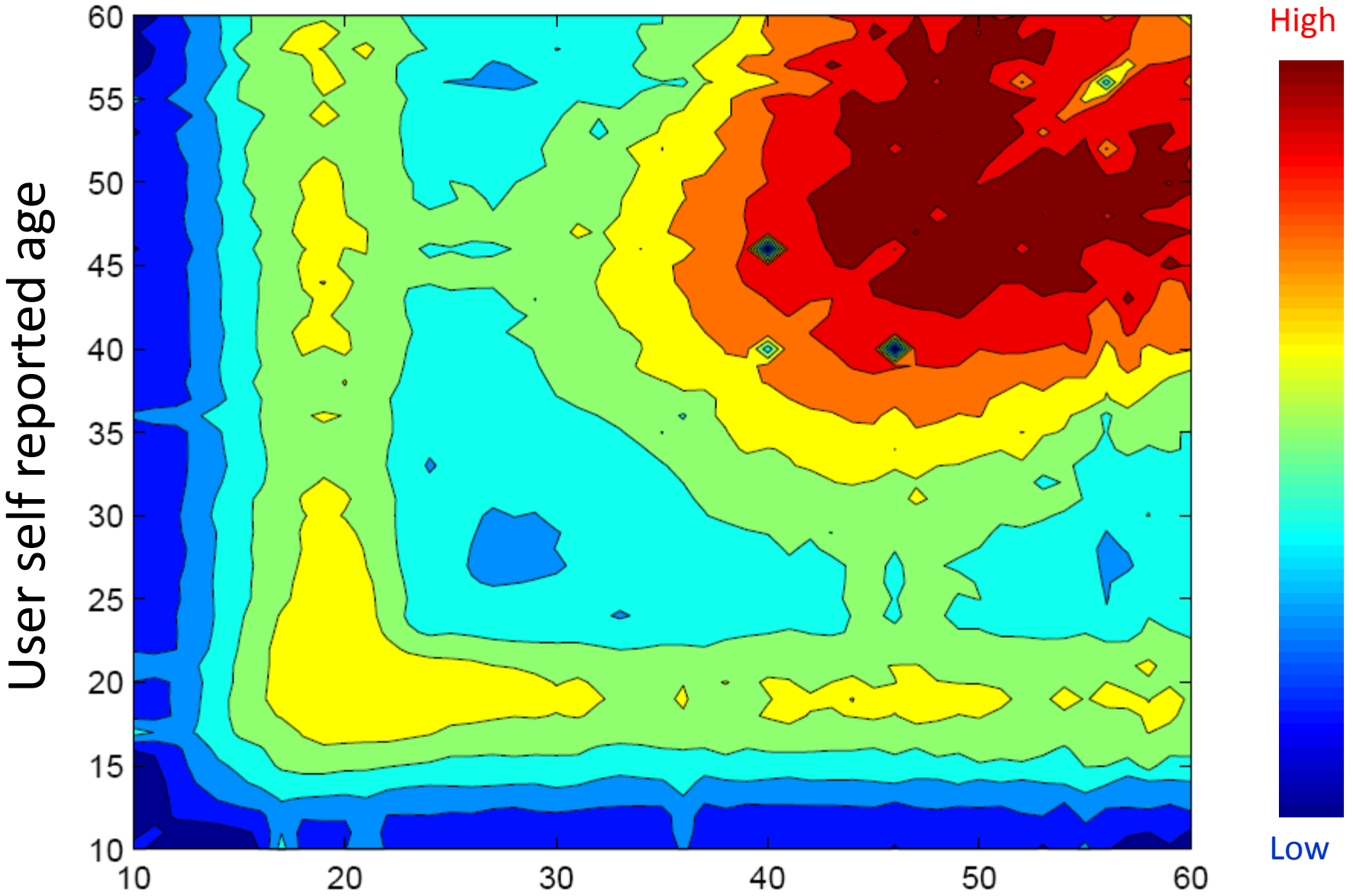
Age: Number of conversations



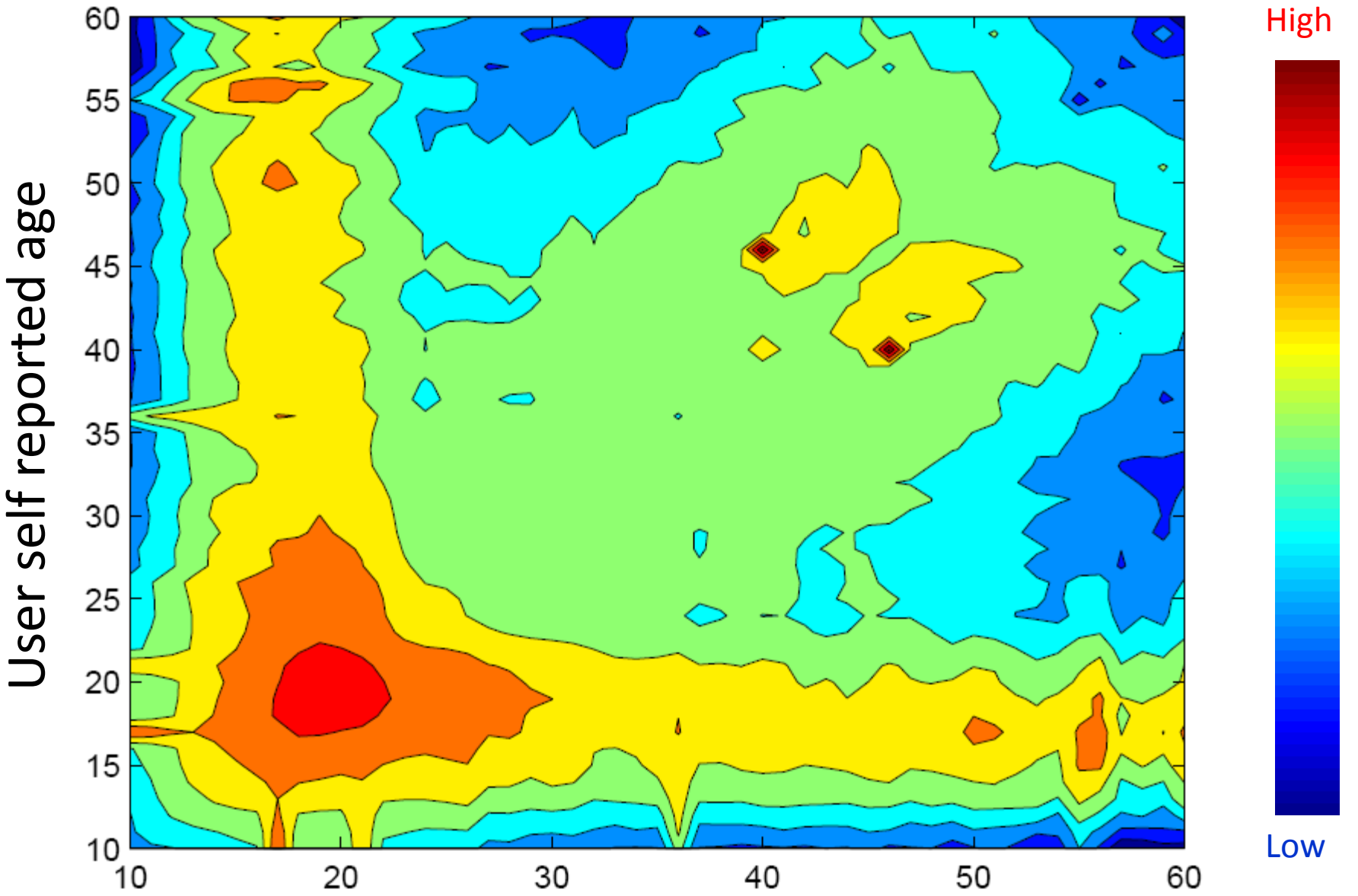
Age: Total conversation duration



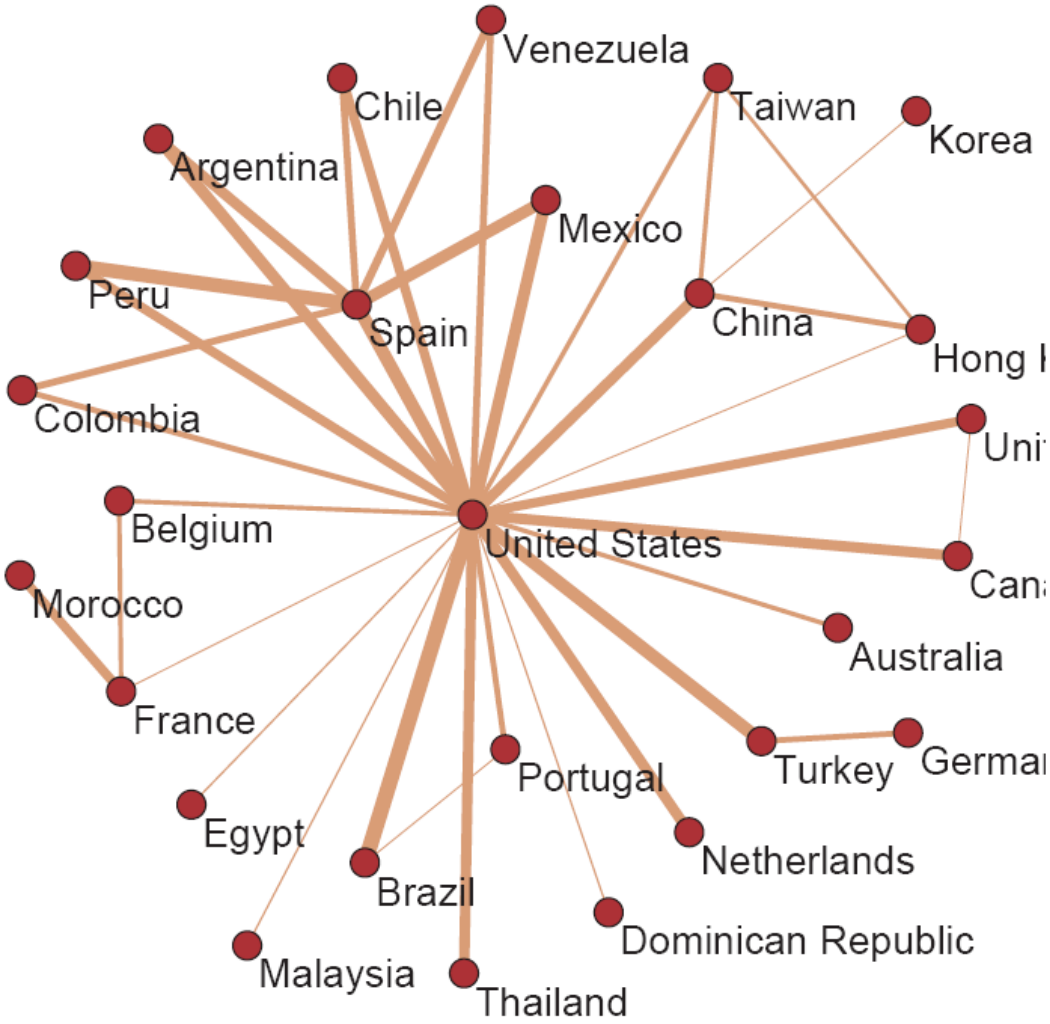
Age: Messages per conversation



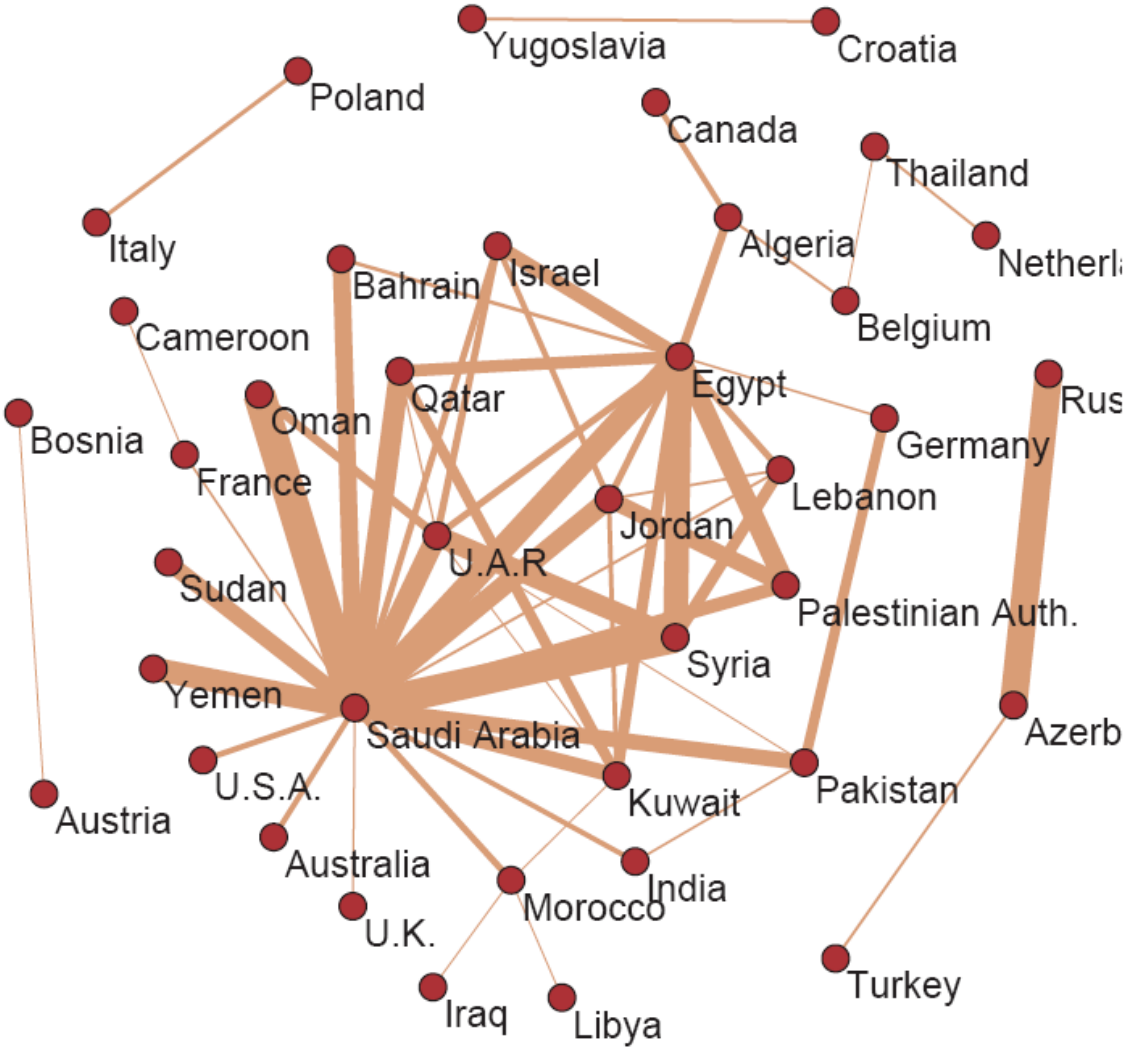
Age: Messages per unit time



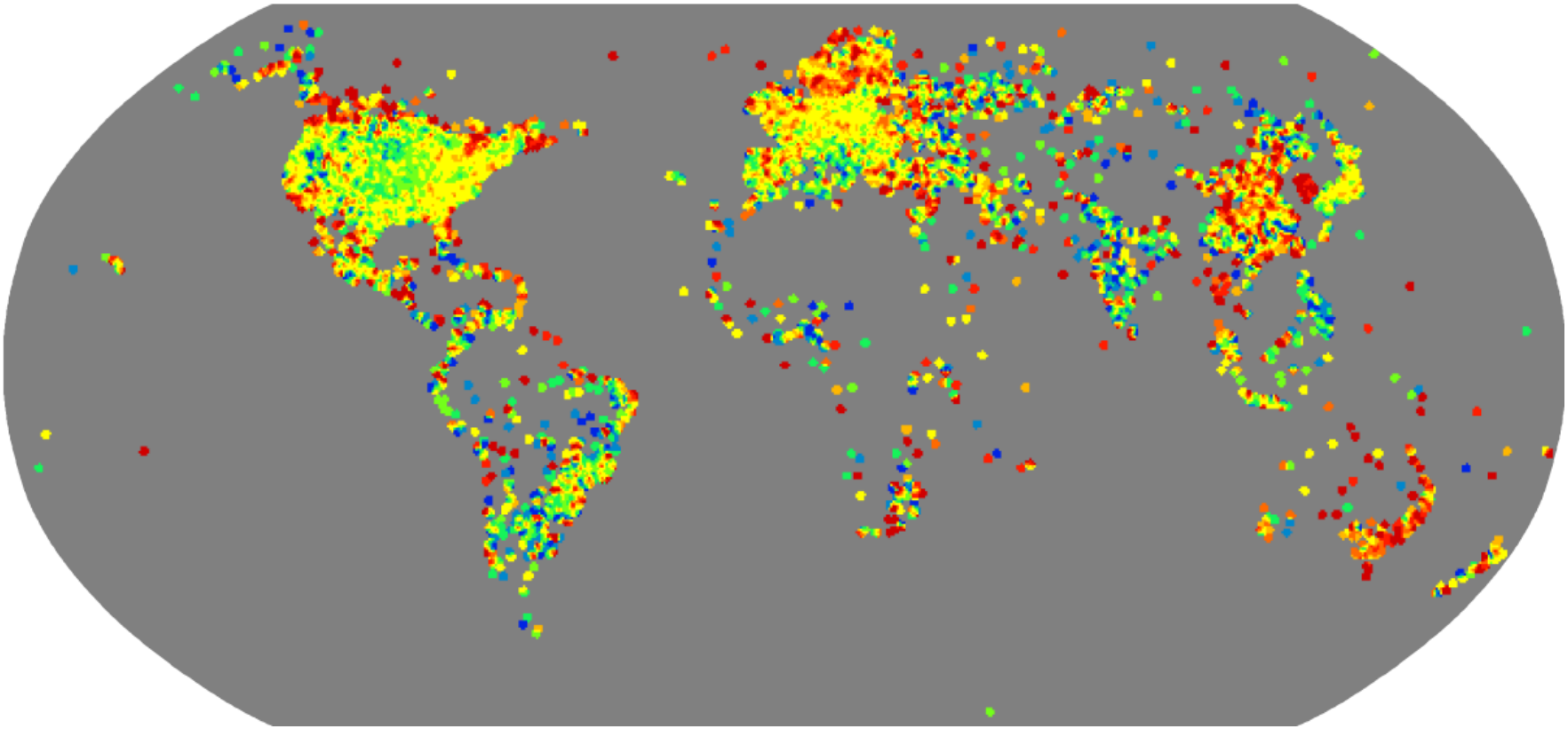
Who talks to whom: Number of conversations



Who talks to whom: Conversation duration

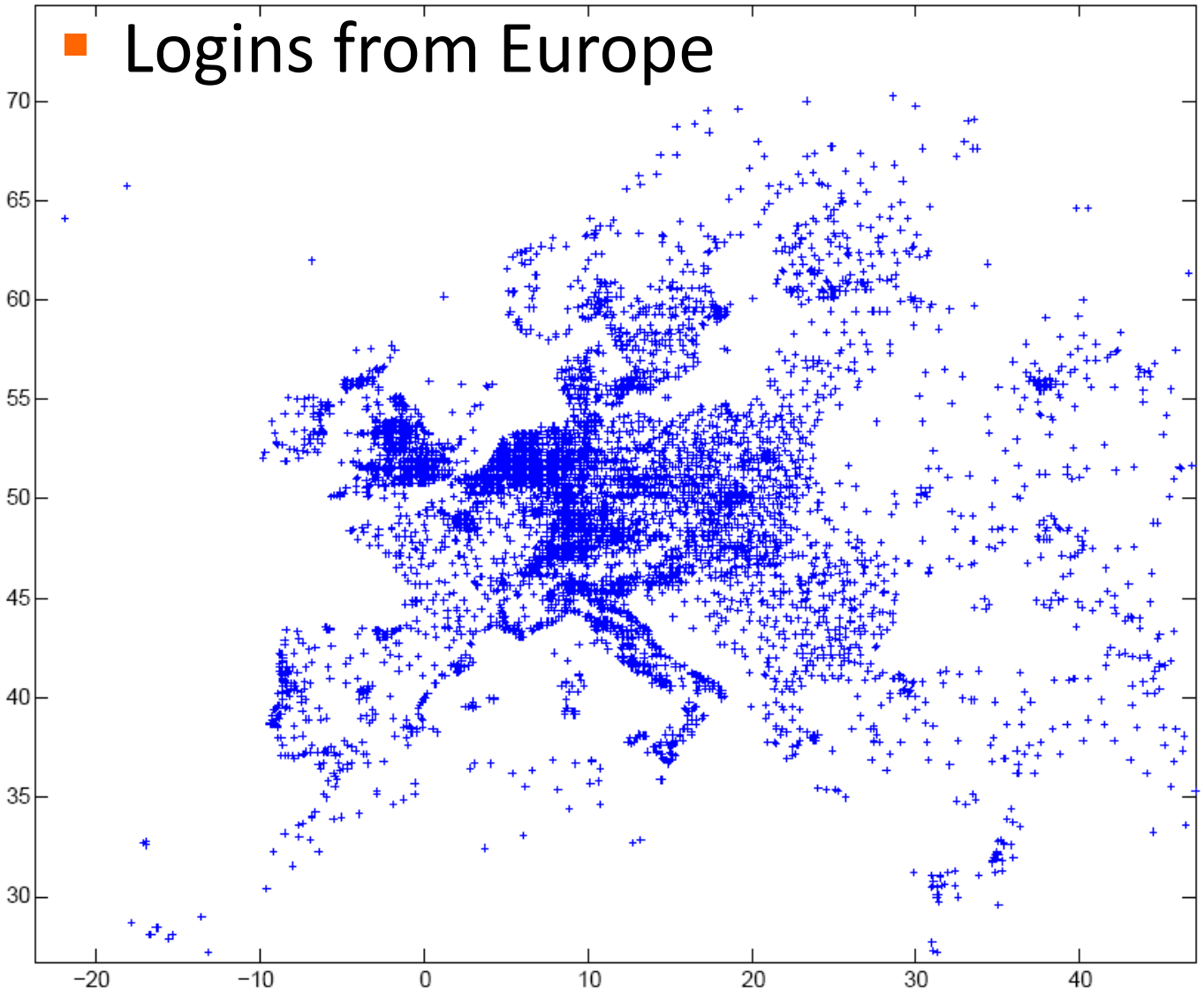


Geography and communication

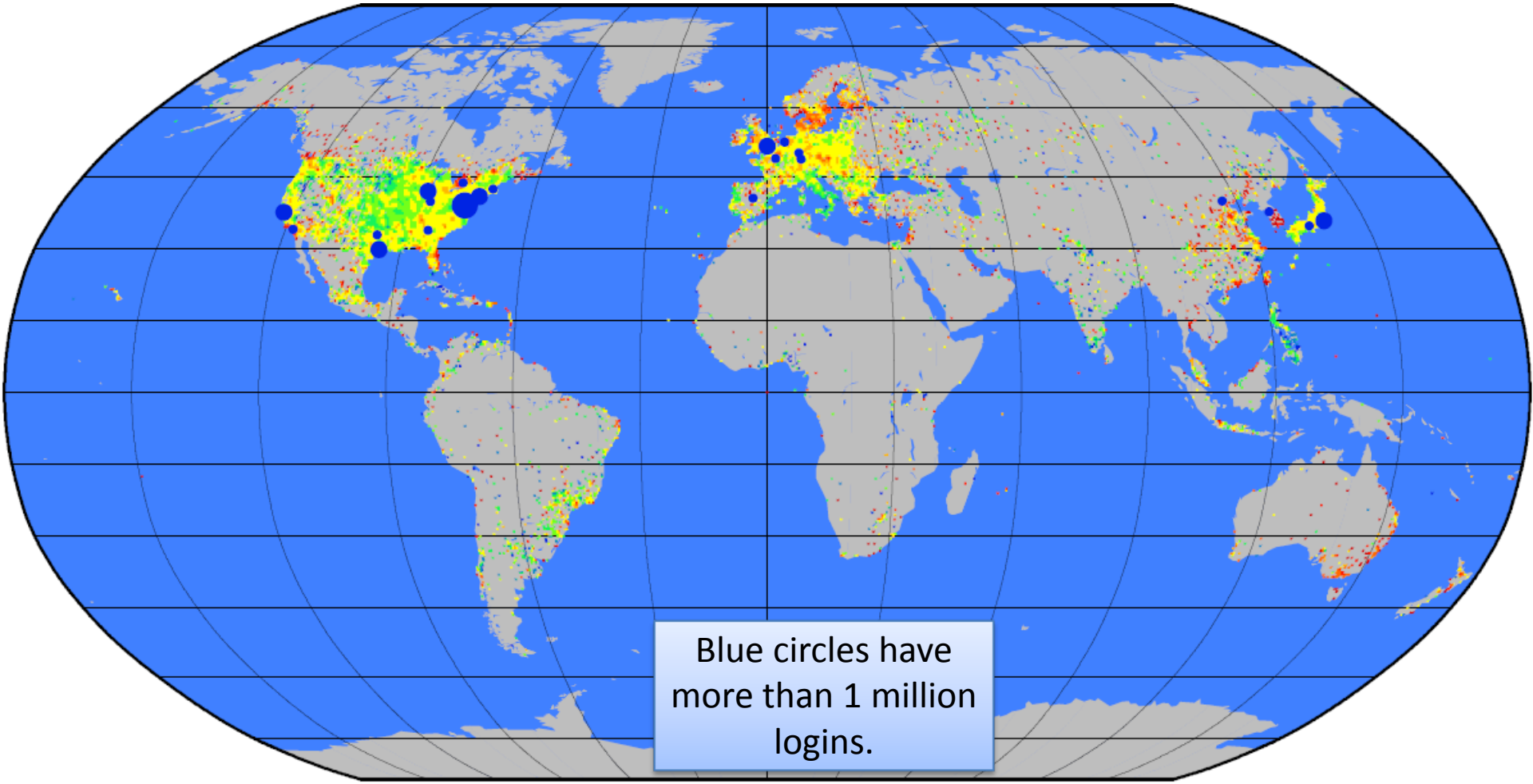


- Count the number of users logging in from particular location on the earth

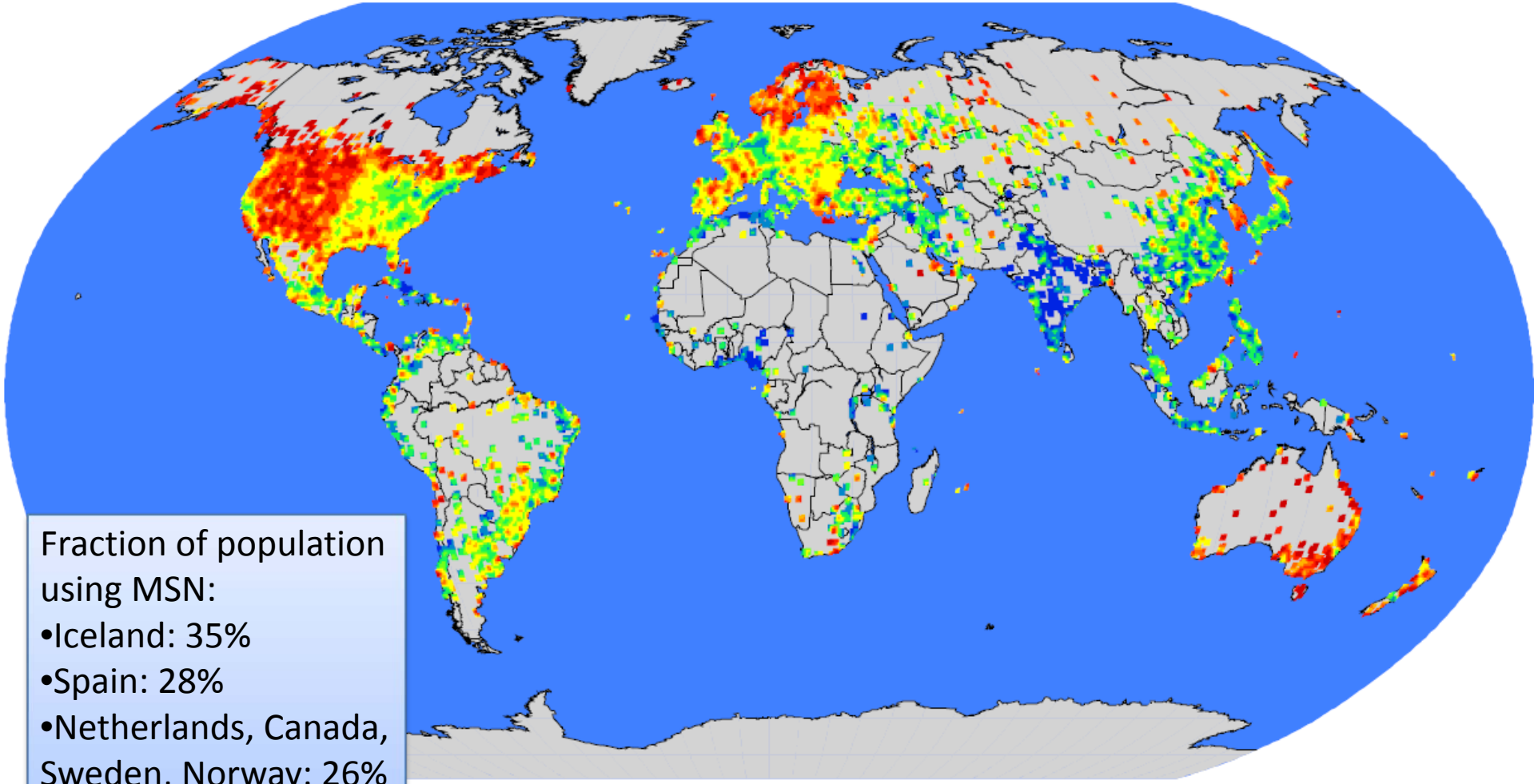
How is Europe talking



Users per geo location

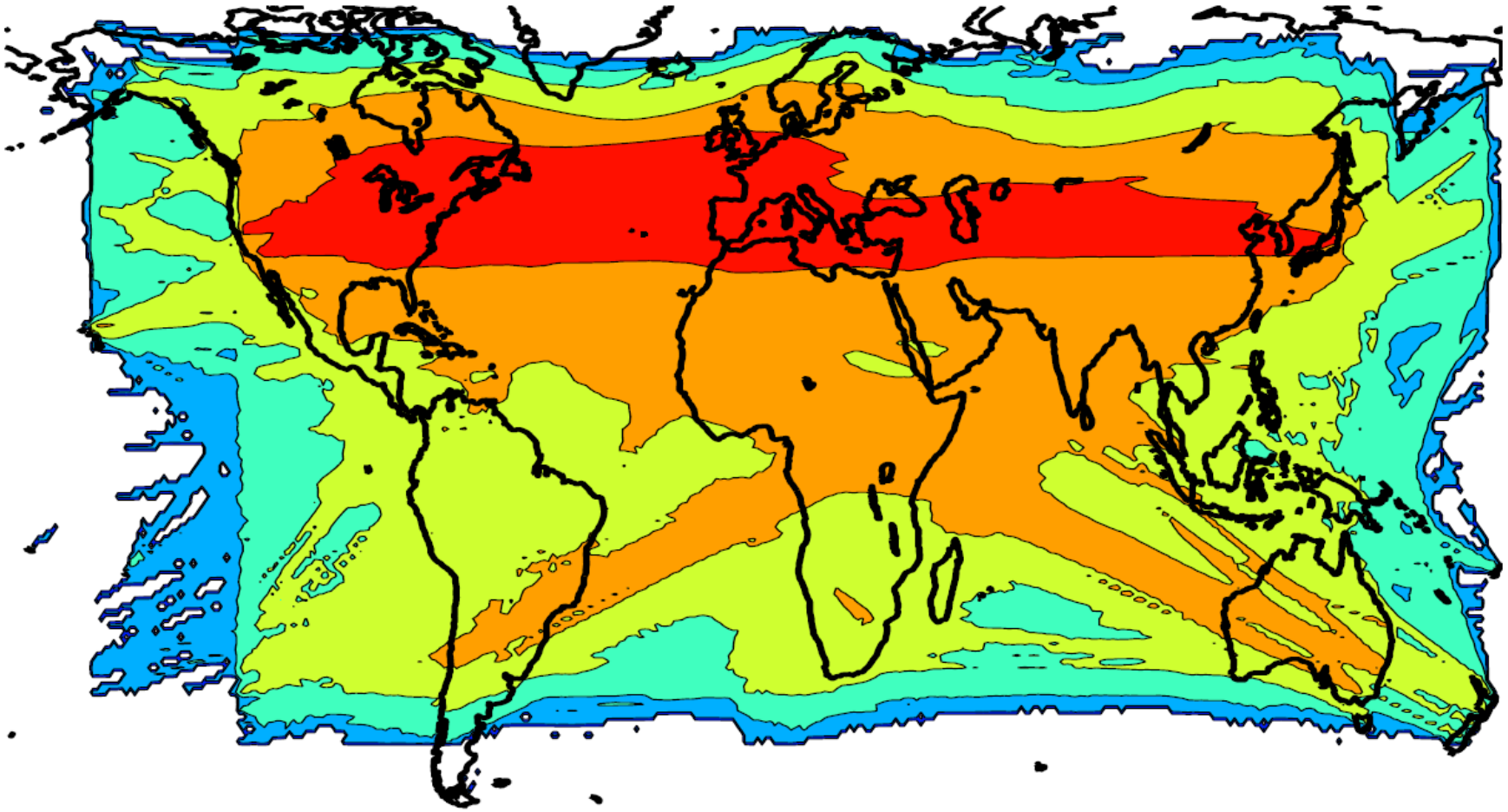


Users per capita



Fraction of population using MSN:
- Iceland: 35%
- Spain: 28%
- Netherlands, Canada, Sweden, Norway: 26%
- France, UK: 18%
- USA, Brazil: 8%

Communication heat map



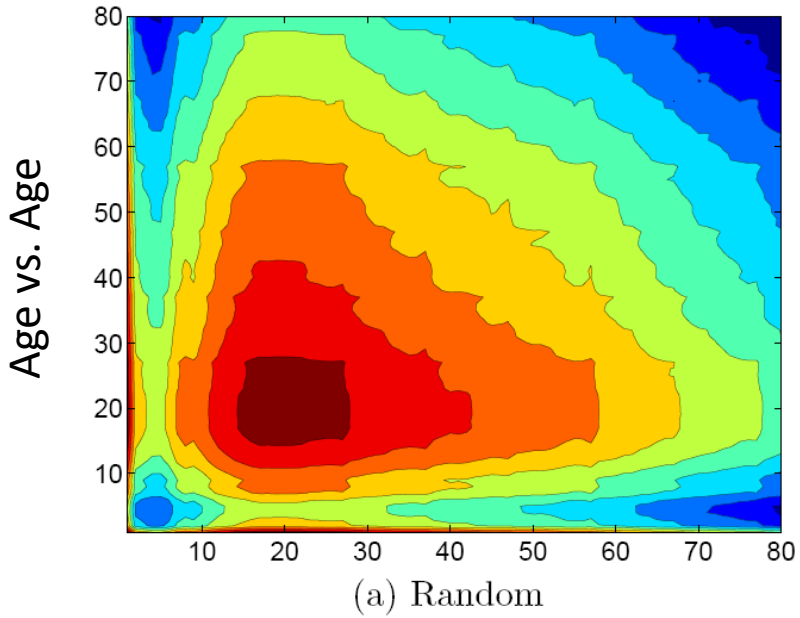
- For each conversation between geo points (A,B) we increase the intensity on the line between A and B

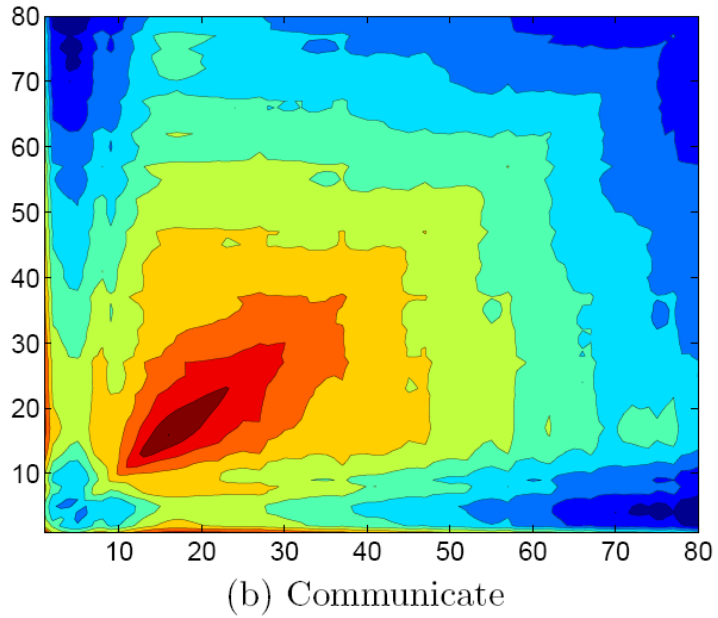
- Correlation:

Attribute	Random	Communicate
Age	-0.0001	0.297
Gender	0.0001	-0.032
ZIP	-0.0003	0.557
County	0.0005	0.704
Language	-0.0001	0.694

- Probability:

Attribute	Random	Communicate
Age	0.030	0.162
Gender	0.434	0.426
ZIP	0.001	0.23
County	0.046	0.734
Language	0.030	0.798

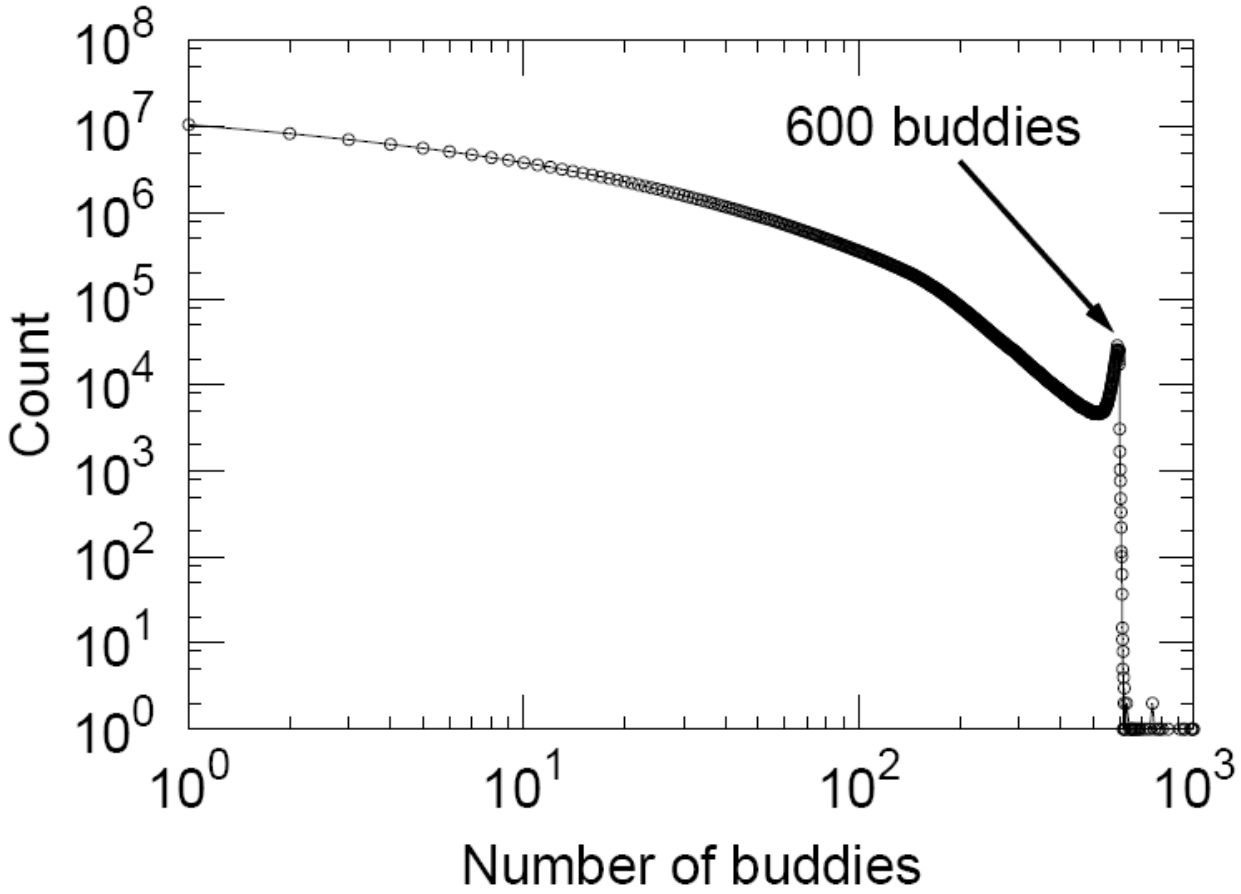




IM Communication Network

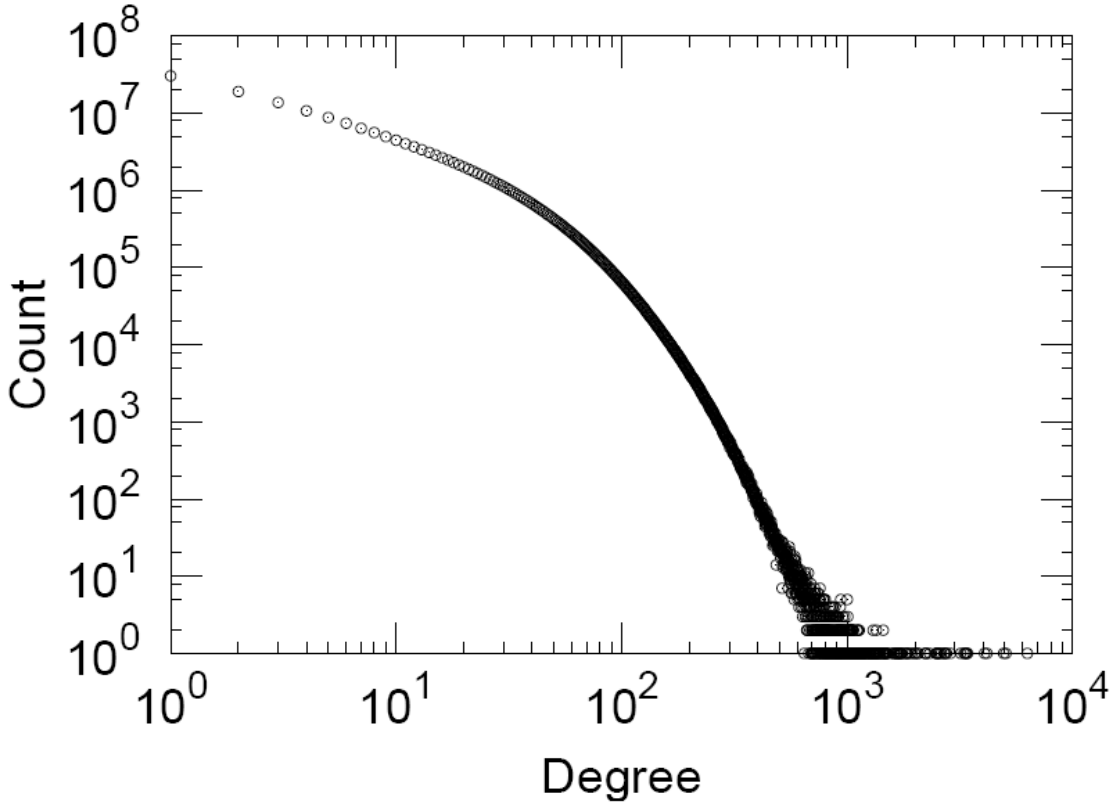
- Buddy graph:
  - 240 million people (people that login in June ’06)
  - 9.1 billion edges (friendship links)
- Communication graph:
  - There is an edge if the users exchanged at least one message in June 2006
  - 180 million people
  - 1.3 billion edges
  - 30 billion conversations

Buddy network: Number of buddies



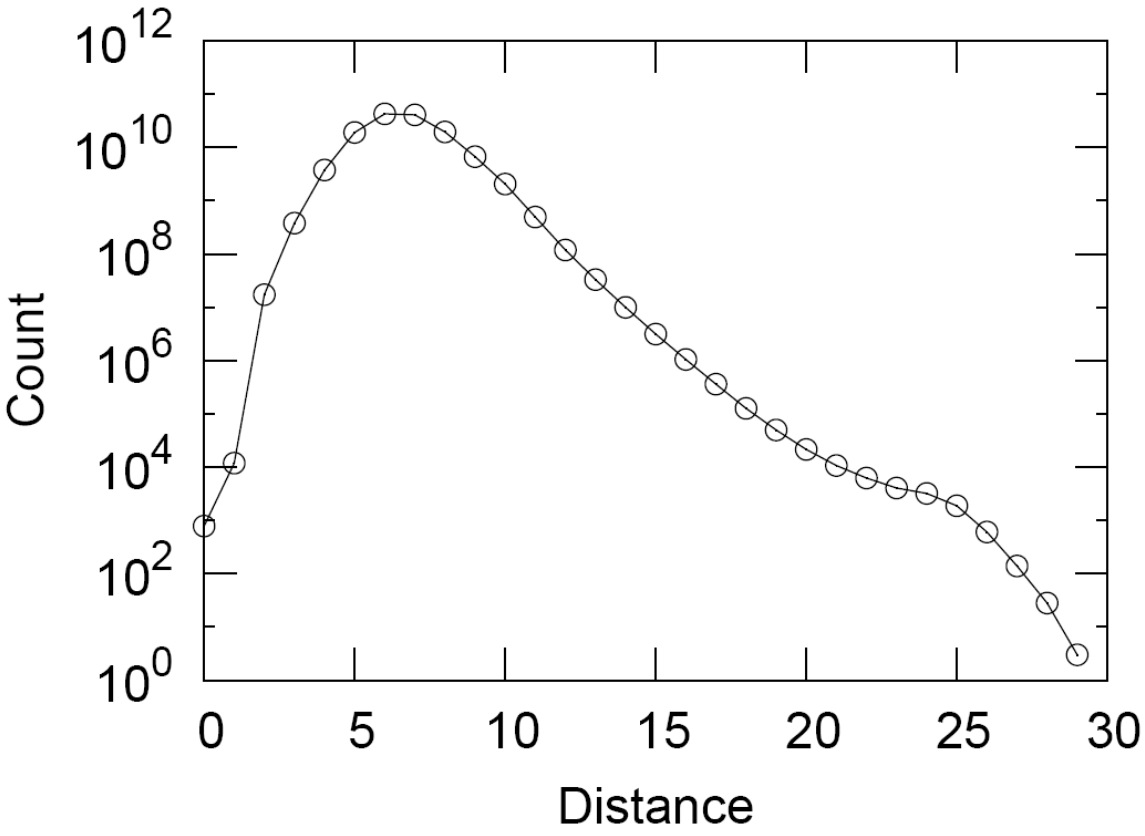
- Buddy graph: 240 million nodes, 9.1 billion edges (~40 buddies per user)

Communication Network: Degree



- Number of people a users talks to in a month

Communication Network: Small-world



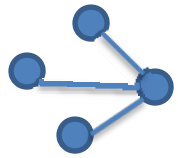
Hops	Nodes
1	10
2	78
3	396
4	8648
5	3299252
6	28395849
7	79059497
8	52995778
9	10321008
10	1955007
11	518410
12	149945
13	44616
14	13740
15	4476
16	1542
17	536
18	167
19	71
20	29
21	16
22	10
23	3
24	2
25	3

- 6 degrees of separation [Milgram ’60s]
- Average distance 5.5
- 90% of nodes can be reached in < 8 hops

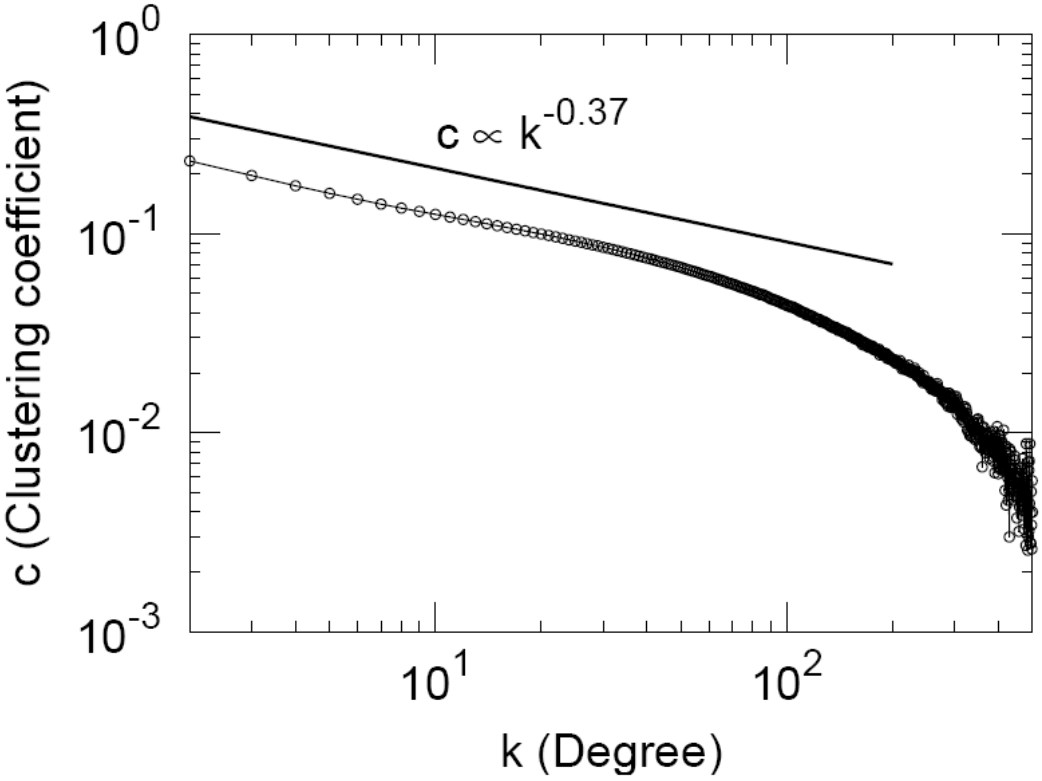
Communication network: Clustering

- How many triangles are closed?
- Clustering normally decays as k^{-1}
- Communication network is highly clustered: $k^{-0.37}$

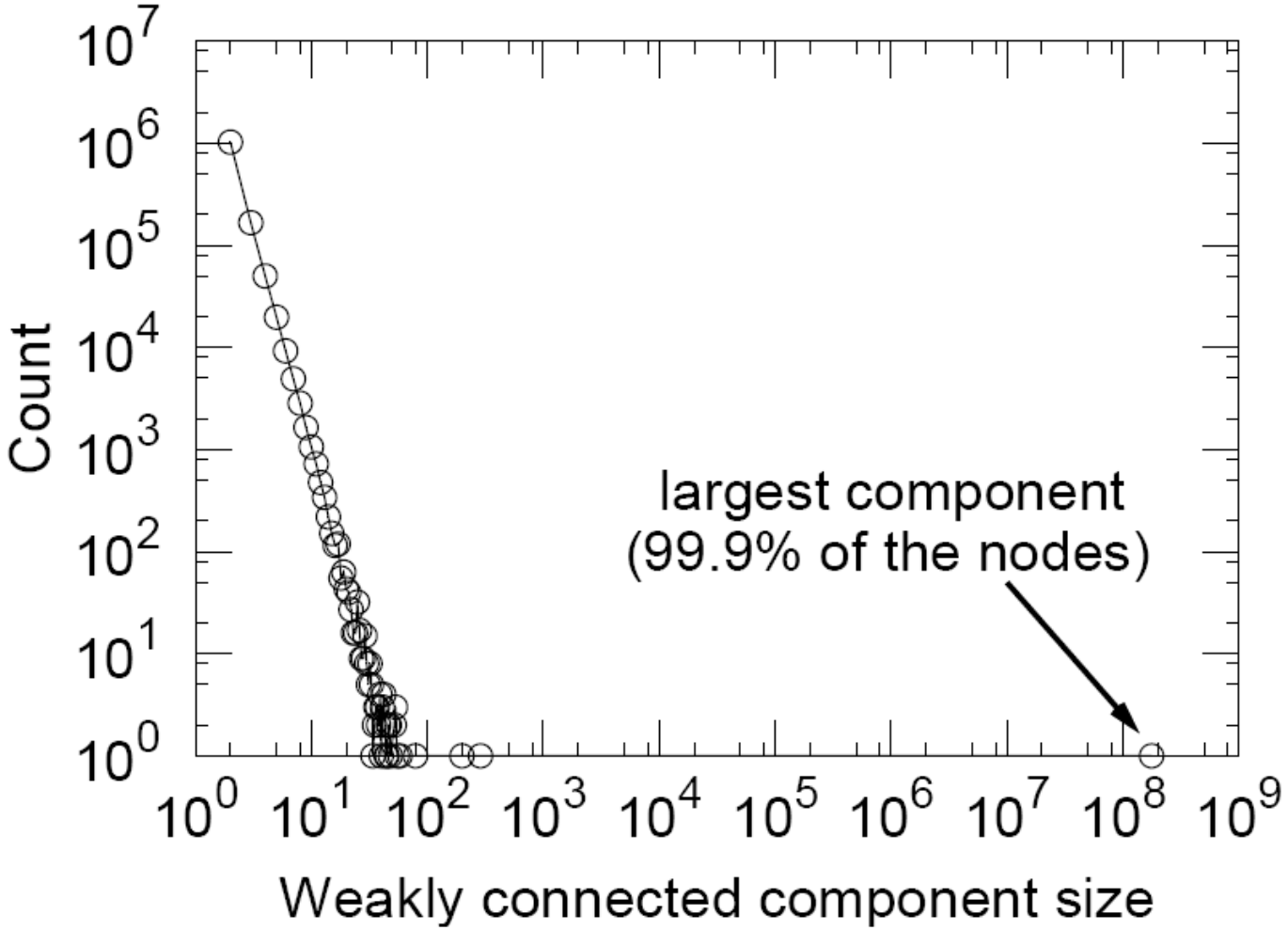
High clustering



Low clustering

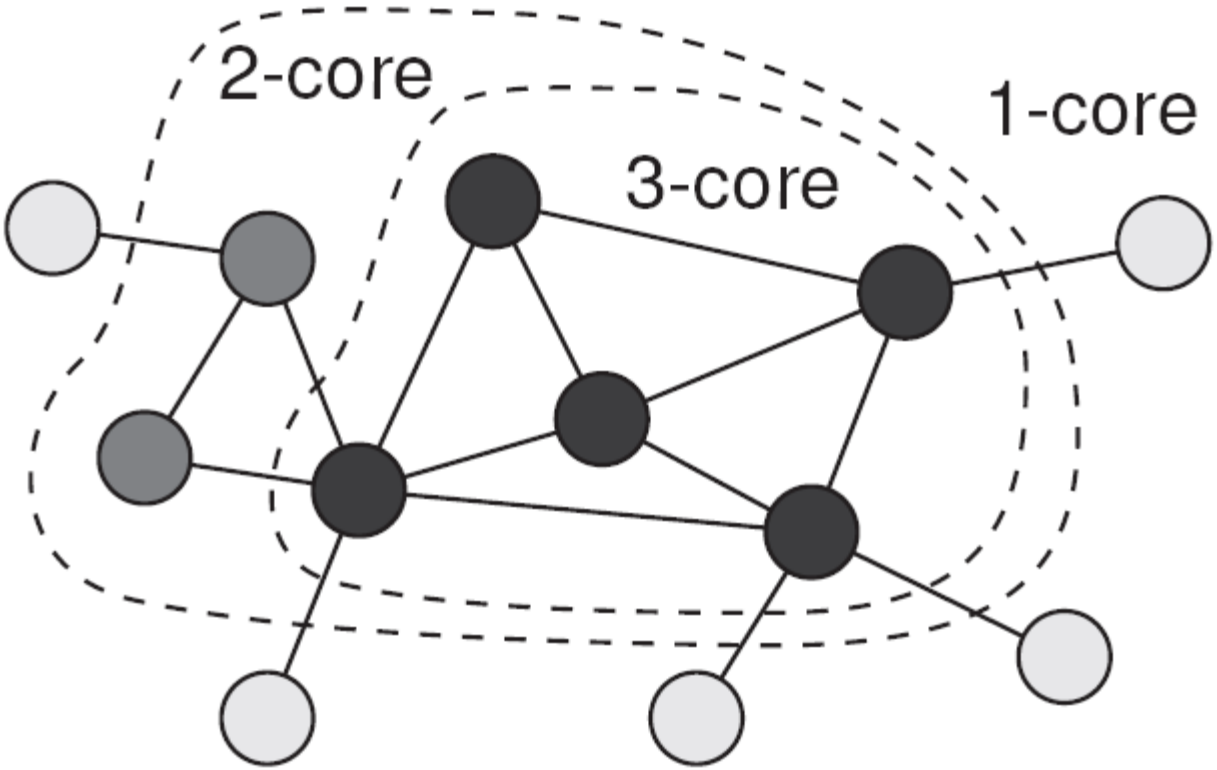


Communication Network Connectivity

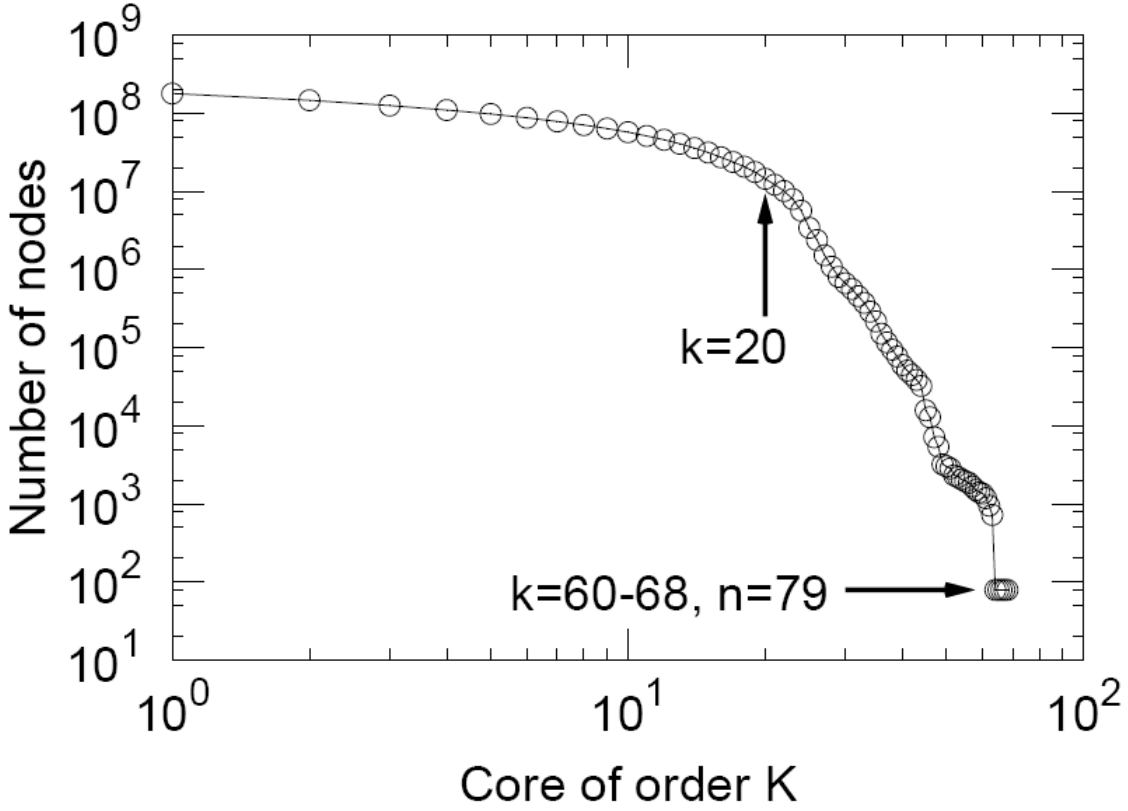


k-Cores decomposition

- What is the structure of the core of the network?

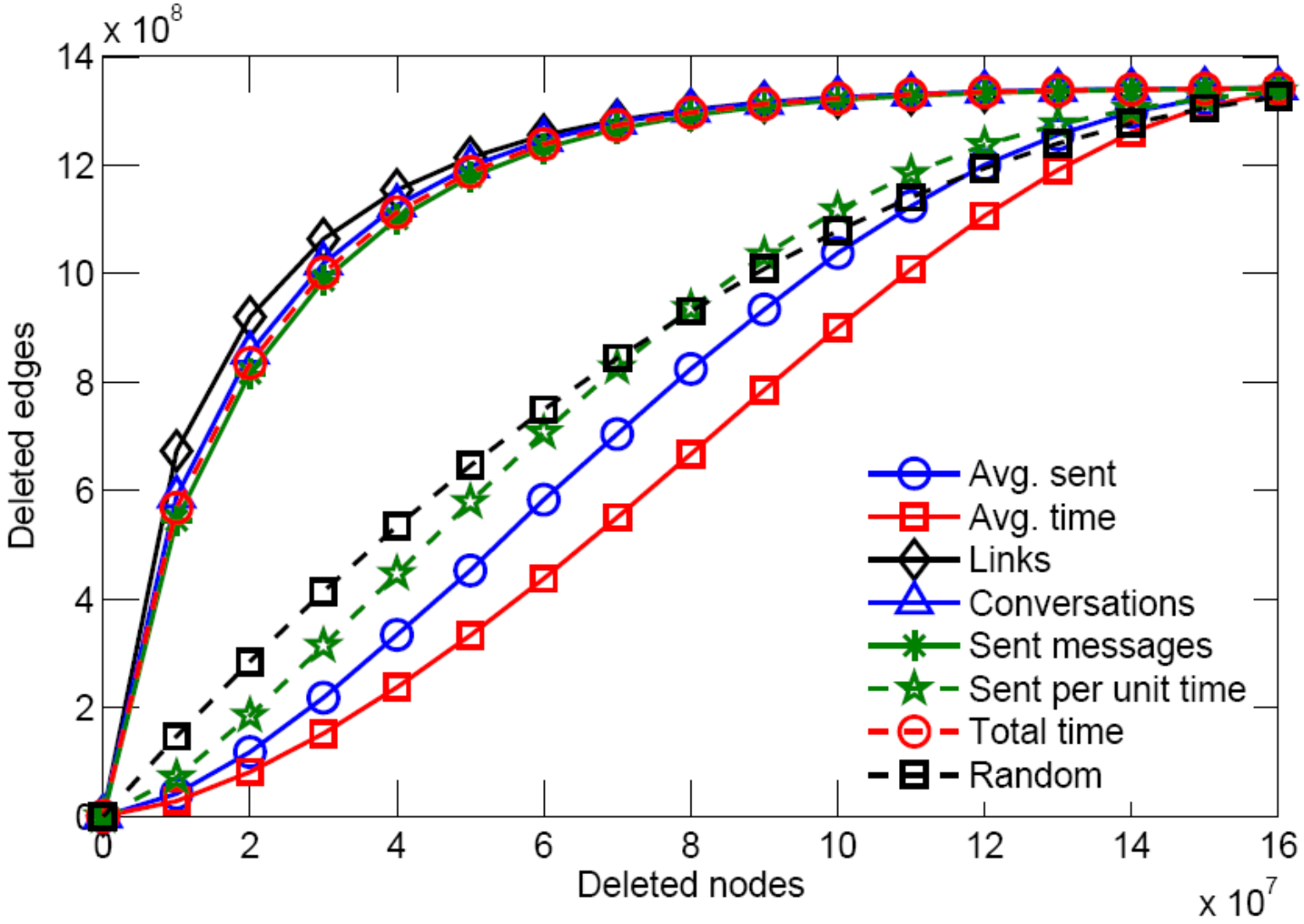


k-Cores: core of the network

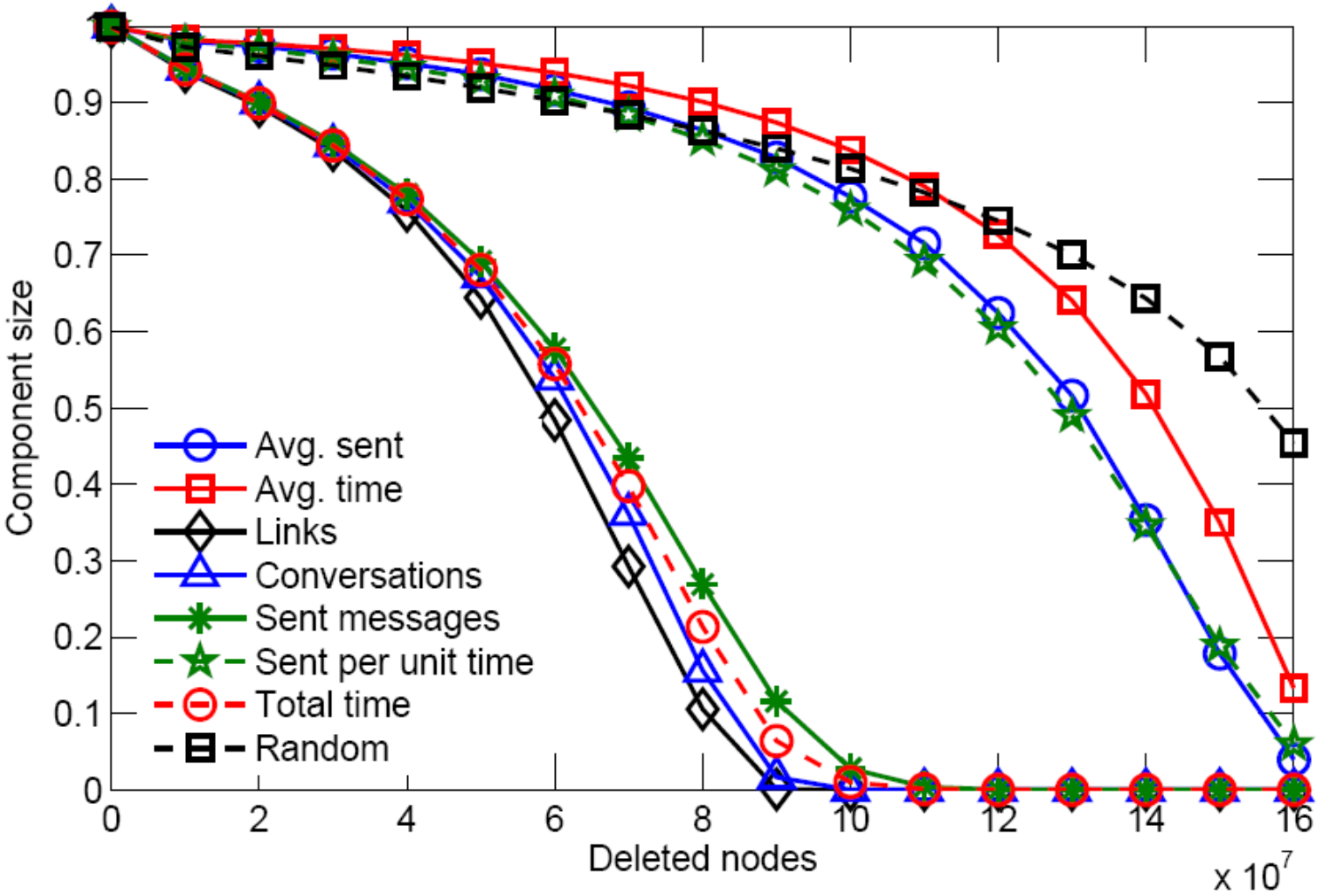


- People with k<20 are the periphery
- Core is composed of 79 people, each having 68 edges among them

Node deletion: Nodes vs. Edges



Node deletion: Connectivity



Web Projections

Learning from contextual graphs of the web

How to predict user intention from the web graph?

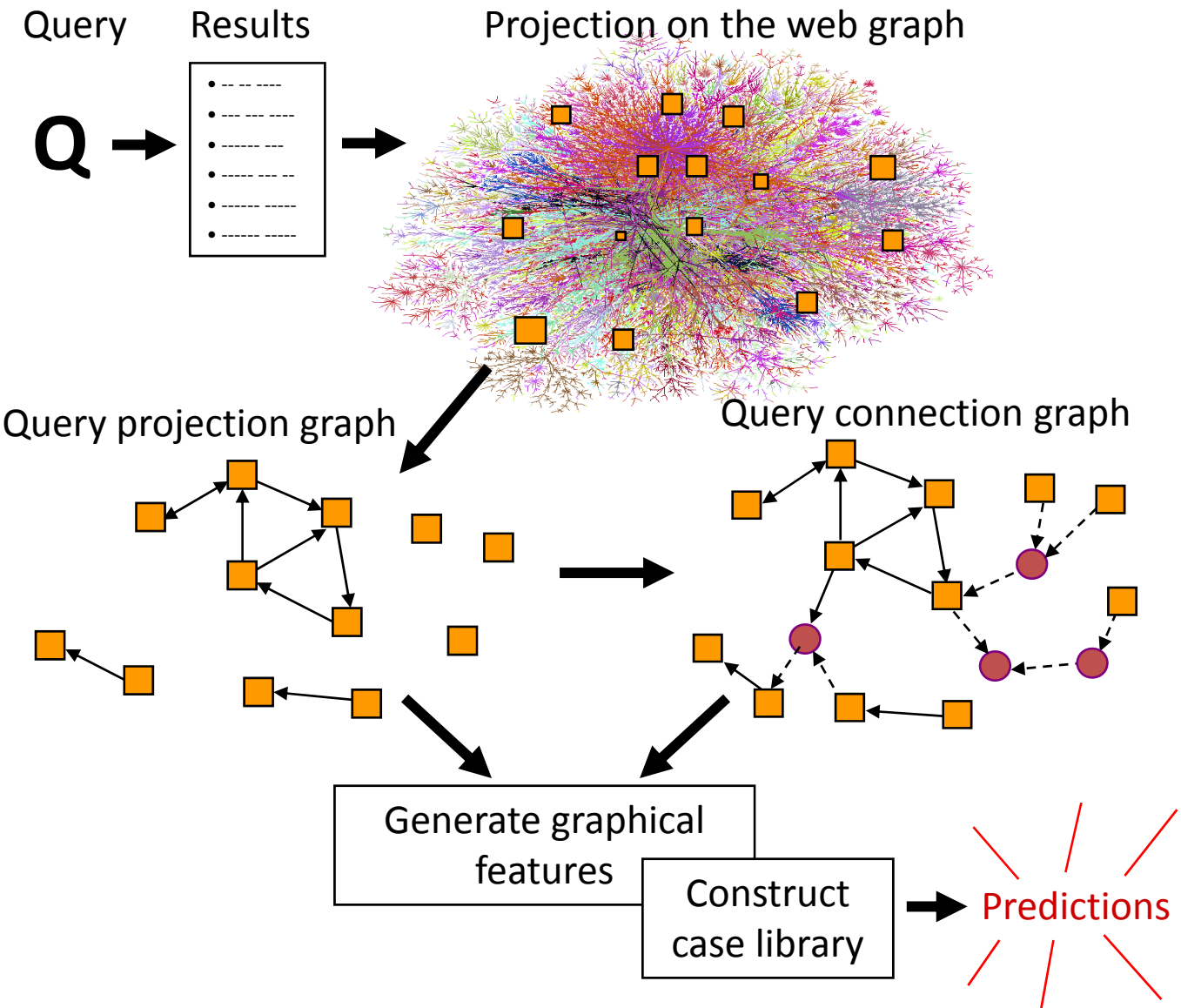
Motivation

- Information retrieval traditionally considered documents as independent
- Web retrieval incorporates global hyperlink relationships to enhance ranking (*e.g.*, PageRank, HITS)
  - Operates on the entire graph
  - Uses just one feature (principal eigenvector) of the graph
- Our work on Web projections focuses on
  - contextual subsets of the web graph; in-between the independent and global consideration of the documents
  - a rich set of graph theoretic properties

Web projections

- Web projections: How they work?
  - Project a set of web pages of interest onto the web graph
  - This creates a subgraph of the web called projection graph
  - Use the graph-theoretic properties of the subgraph for tasks of interest
- Query projections
  - Query results give the context (set of web pages)
  - Use characteristics of the resulting graphs for predictions about search quality and user behavior

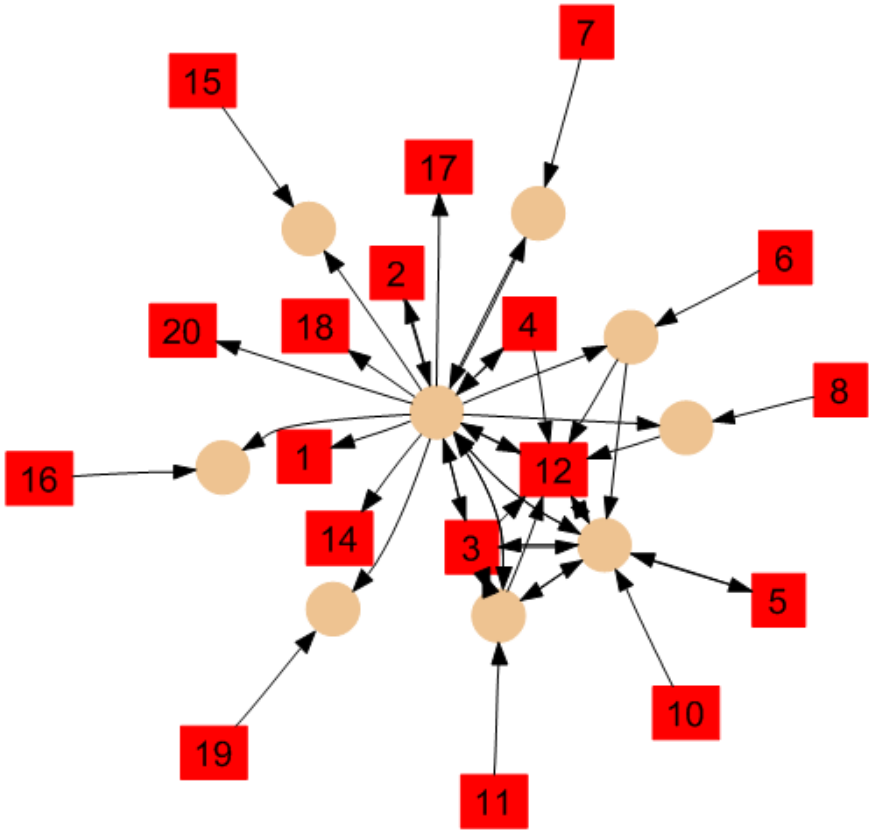
Query projections



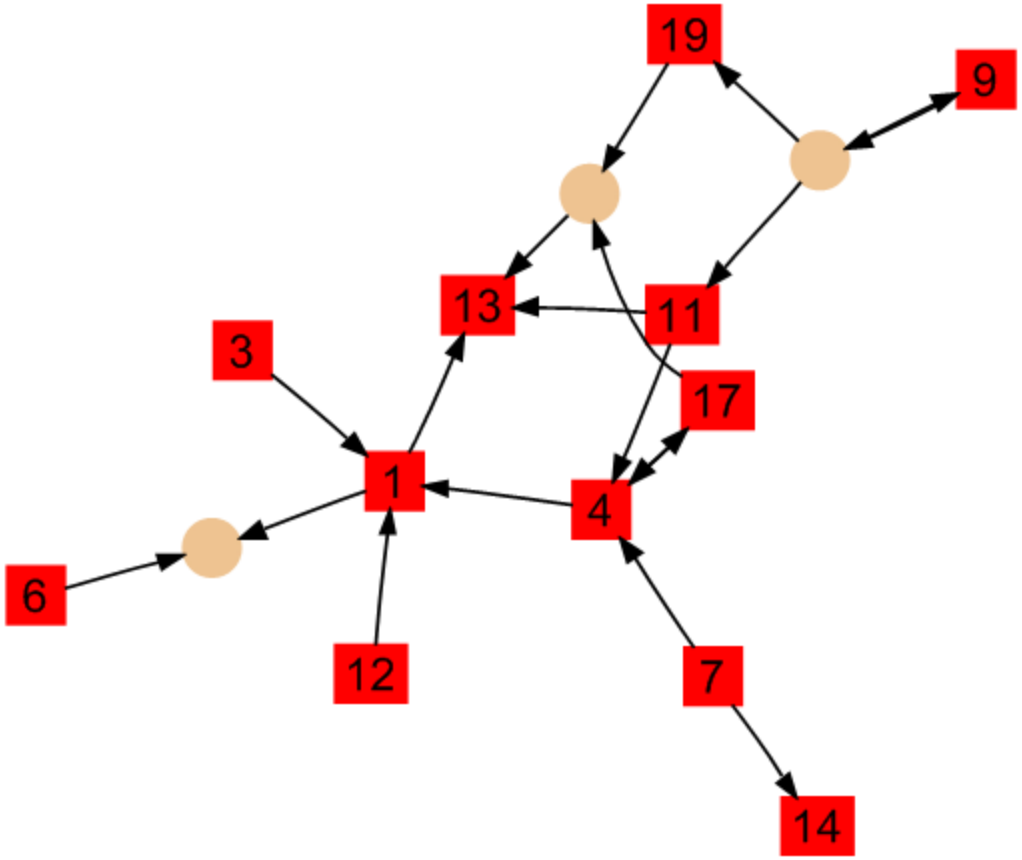
Questions we explore

- How do query search results project onto the underlying web graph?
- Can we predict the quality of search results from the projection on the web graph?
- Can we predict users' behaviors with issuing and reformulating queries?

Is this a good set of search results?



Will the user reformulate the query?

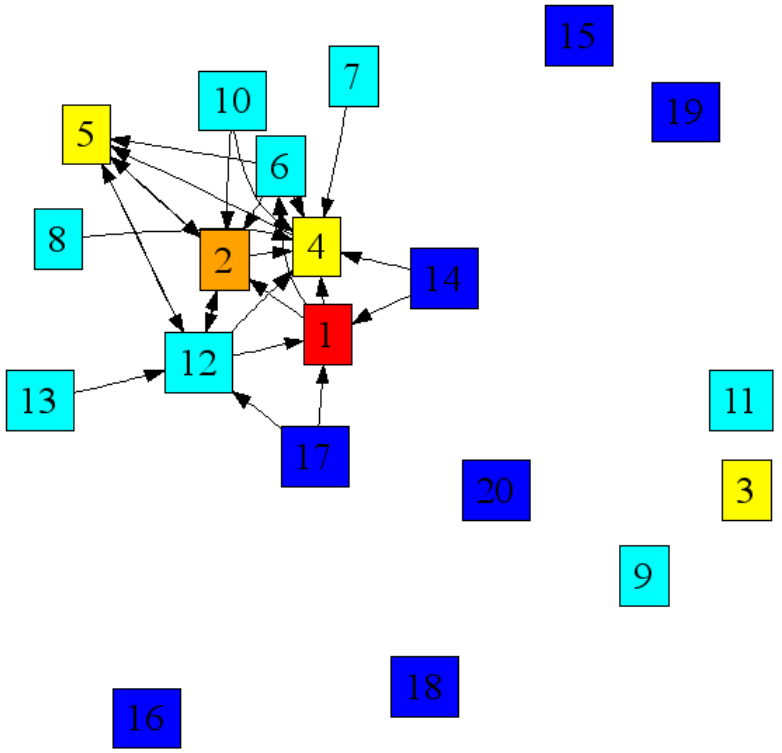


Resources and concepts

- Web as a graph
  - URL graph:
    - Nodes are web pages, edges are hyper-links
    - March 2006
    - Graph: 22 million nodes, 355 million edges
  - Domain graph:
    - Nodes are domains (cmu.edu, bbc.co.uk). Directed edge (u,v) if there exists a webpage at domain u pointing to v
    - February 2006
    - Graph: 40 million nodes, 720 million edges
- Contextual subgraphs for queries
  - Projection graph
  - Connection graph
- Compute graph-theoretic features

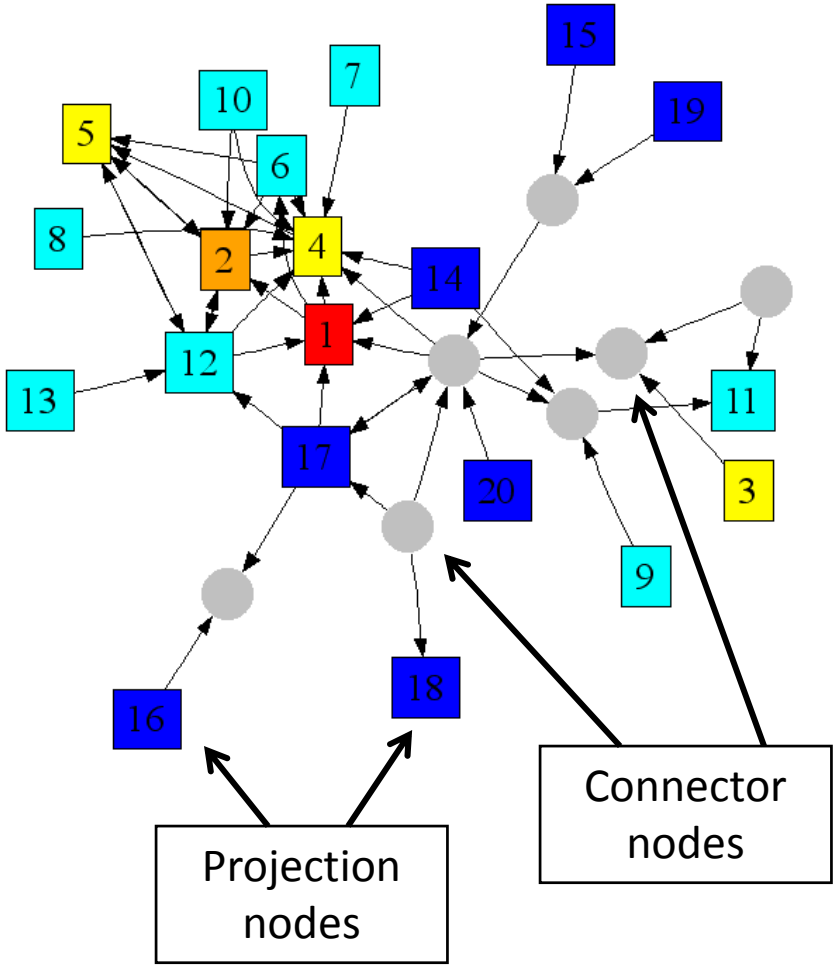
“Projection” graph

- Example query: *Subaru*
- Project top 20 results by the search engine
- Number in the node denotes the search engine rank
- Color indicates relevancy as assigned by human:
  - **Perfect**
  - **Excellent**
  - **Good**
  - **Fair**
  - **Poor**
  - **Irrelevant**



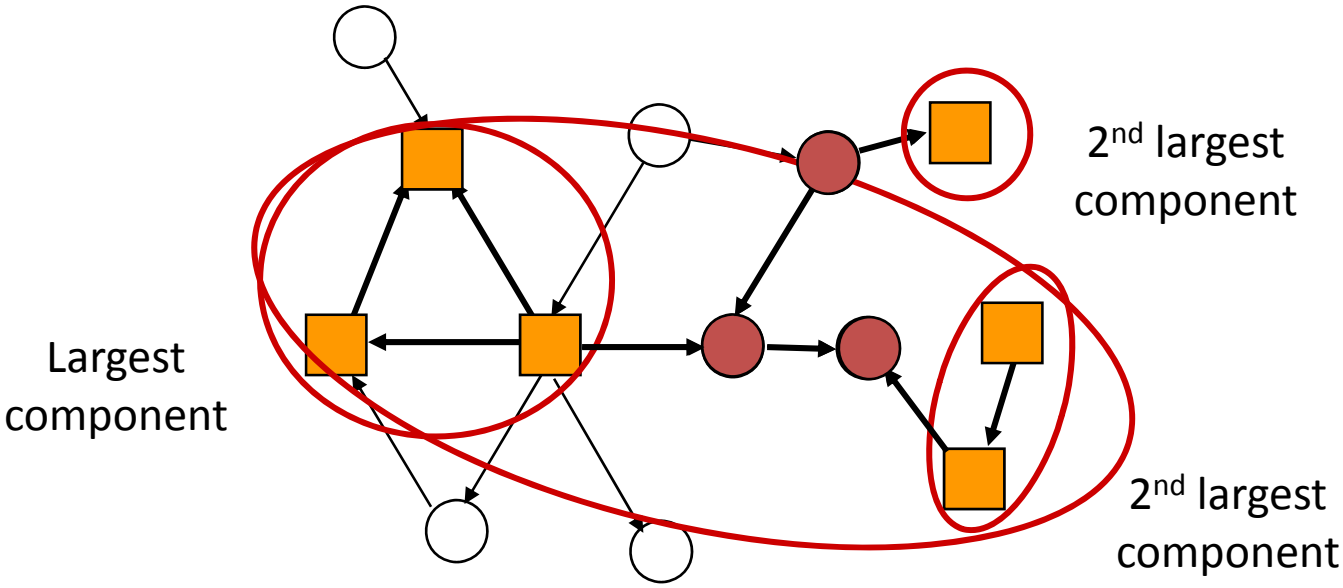
“Connection” graph

- Projection graph is generally disconnected
- Find connector nodes
- Connector nodes are existing nodes that are not part of the original result set
- Ideally, we would like to introduce fewest possible nodes to make projection graph connected



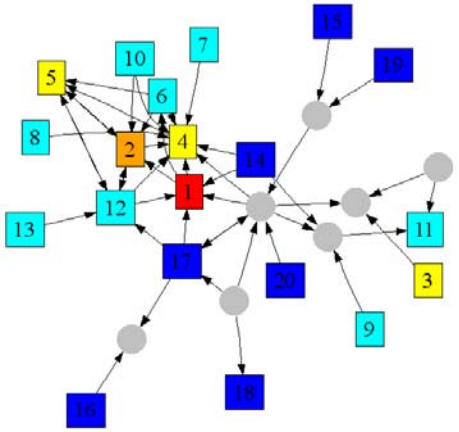
Finding connector nodes

- Find connector nodes is a Steiner tree problem which is NP hard
- Our heuristic:
  - Connect 2nd largest connected component via shortest path to the largest
  - This makes a new largest component
  - Repeat until the graph is connected

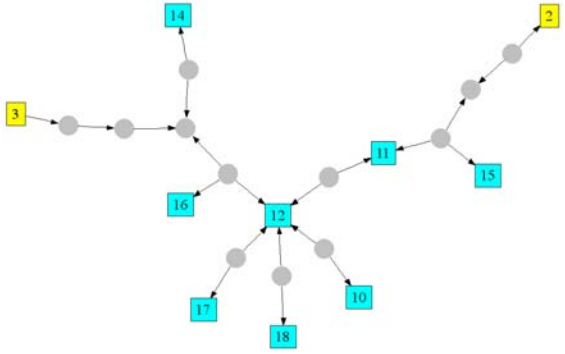


Extracting graph features

- The idea
  - Find features that describe the structure of the graph
  - Then use the features for machine learning
- Want features that describe
  - Connectivity of the graph
  - Centrality of projection and connector nodes
  - Clustering and density of the core of the graph

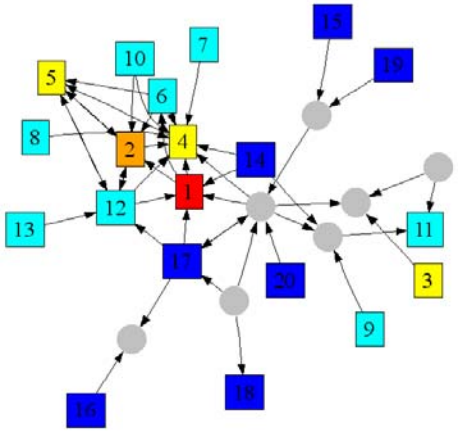


vs.

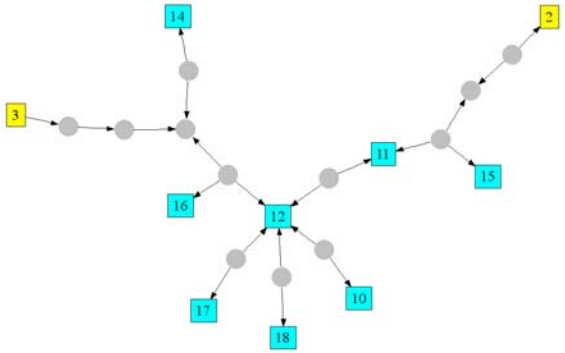


Examples of graph features

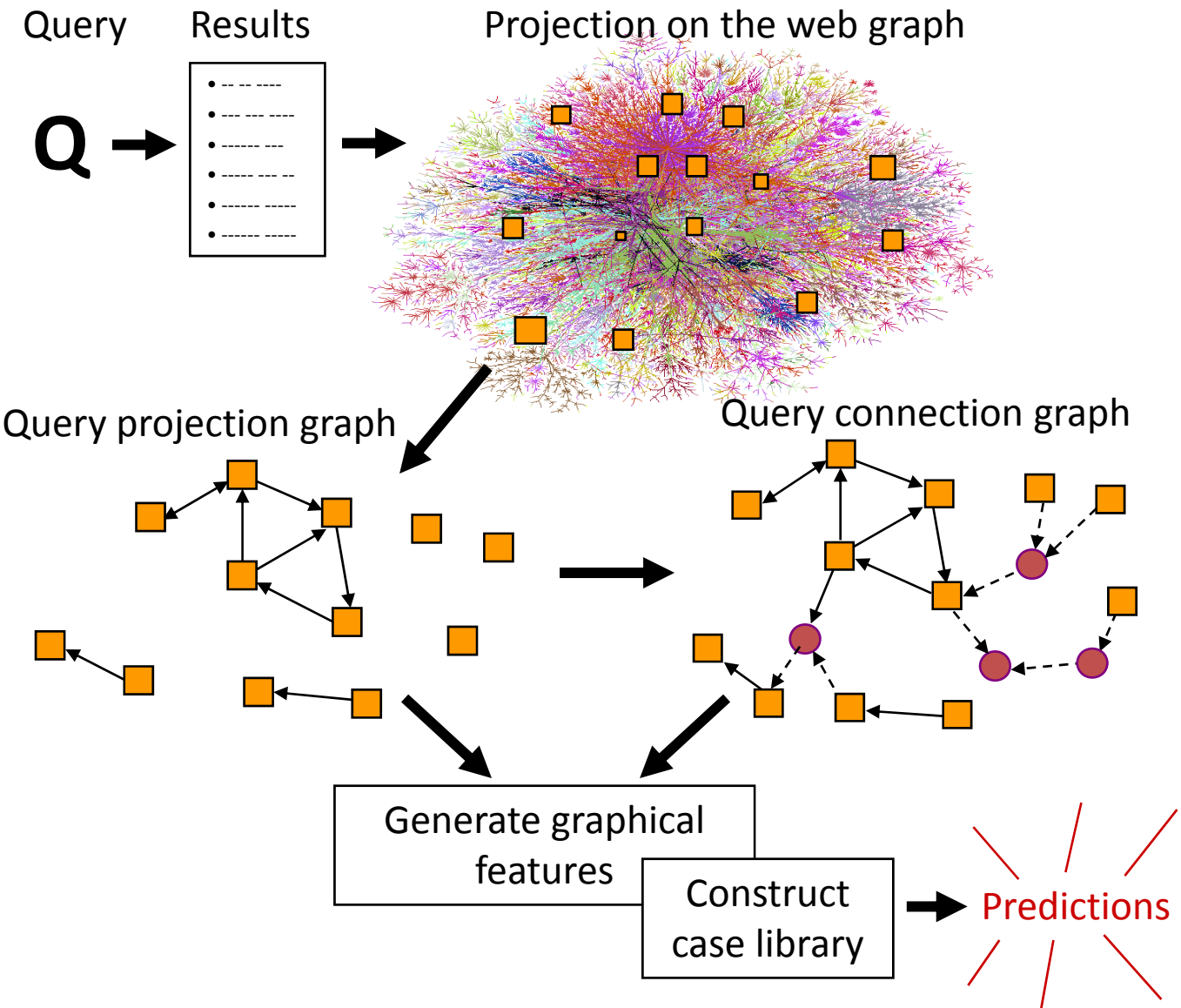
- Projection graph
  - Number of nodes/edges
  - Number of connected components
  - Size and density of the largest connected component
  - Number of triads in the graph
- Connection graph
  - Number of connector nodes
  - Maximal connector node degree
  - Mean path length between projection/connector nodes
  - Triads on connector nodes
- We consider 55 features total



vs.



Experimental setup



Constructing case library for machine learning

- Given a task of interest
- Generate contextual subgraph and extract features
- Each graph is labeled by target outcome
- Learn statistical model that relates the features with the outcome
- Make prediction on unseen graphs

Experiments overview

- Given a set of search results generate projection and connection graphs and their features
- Predict quality of a search result set
  - Discriminate top20 vs. top40to60 results
  - *Predict rating of highest rated document in the set* 
- Predict user behavior
  - *Predict queries with high vs. low reformulation probability* 
  - Predict query transition (generalization vs. specialization)
  - Predict direction of the transition

Experimental details

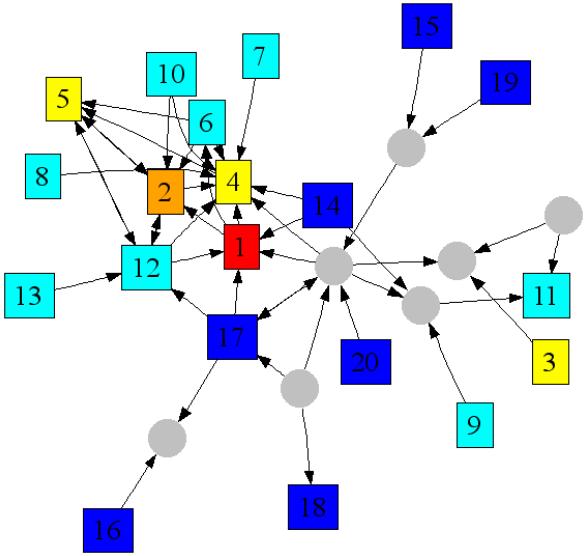
- Features
  - 55 graphical features
  - Note we use only graph features, no content
- Learning
  - We use probabilistic decision trees ("DNet")
- Report classification accuracy using 10-fold cross validation
- Compare against 2 baselines
  - Marginals: Predict most common class
  - RankNet: use 350 traditional features (document, anchor text, and basic hyperlink features)

Search results quality

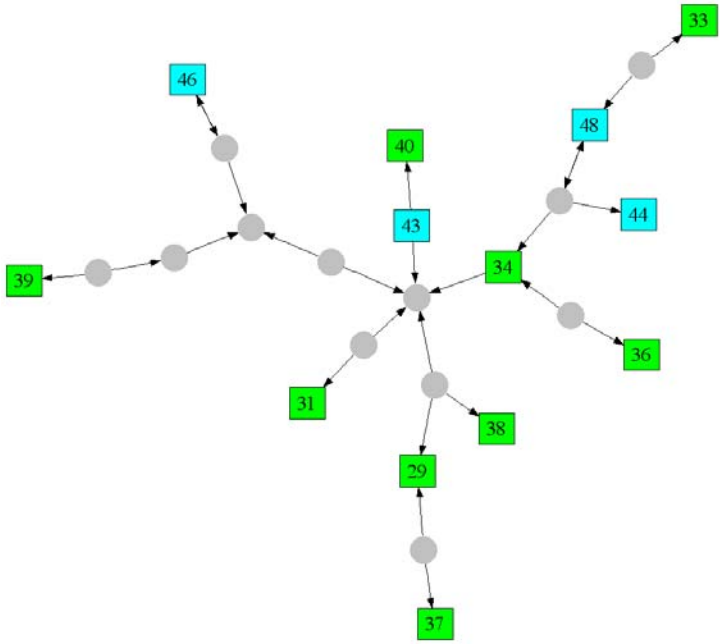
- Dataset:
  - 30,000 queries
  - Top 20 results for each
    - Each result is labeled by a human judge using a 6-point scale from "Perfect" to "Bad"
- Task:
  - Predict the highest rating in the set of results
    - 6-class problem
    - 2-class problem: “Good” (top 3 ratings) vs. “Poor” (bottom 3 ratings)

Search quality: the task

- Predict the rating of the top result in the set



Predict "Good"



Predict "Poor"

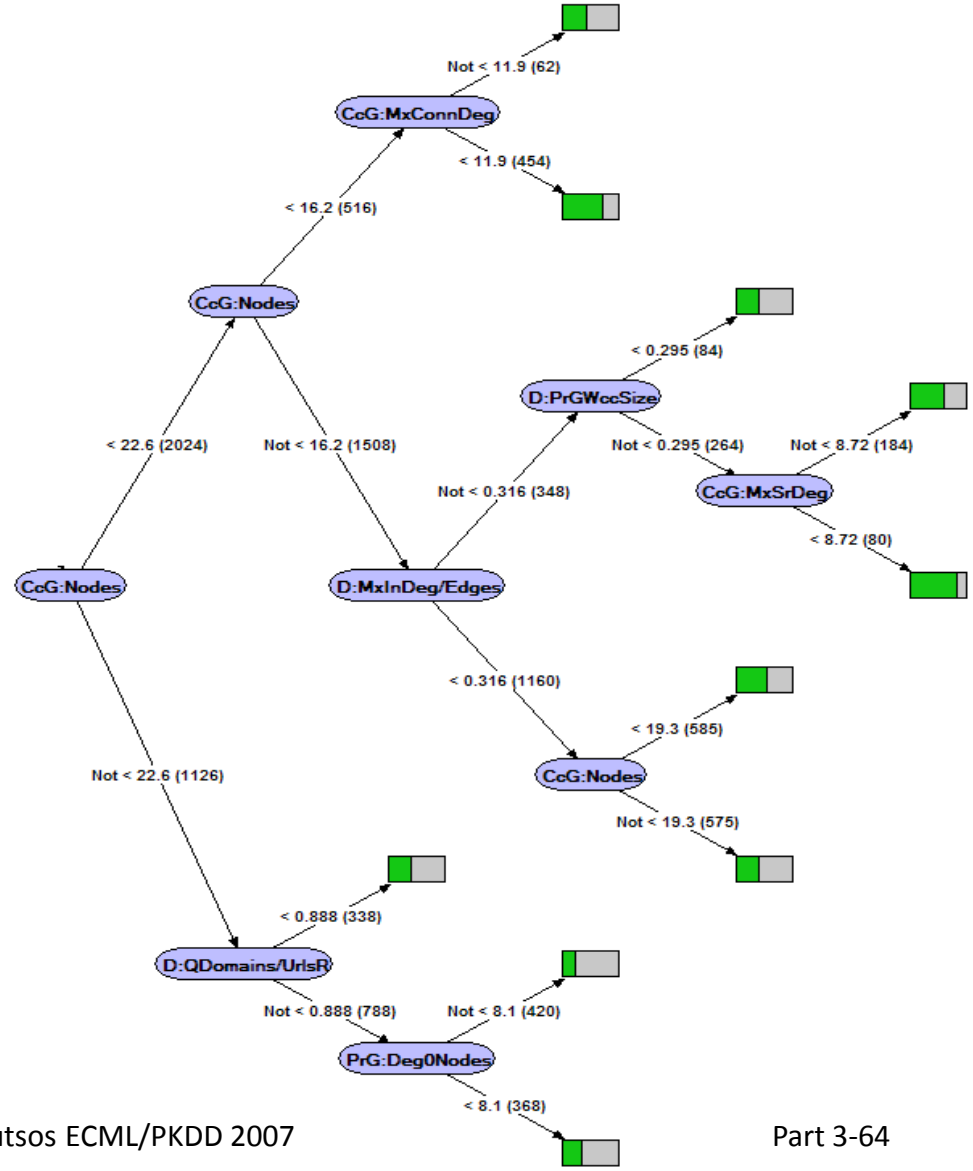
Search quality: results

- Predict top human rating in the set
  - Binary classification: Good vs. Poor
- 10-fold cross validation classification accuracy
- Observations:
  - Web Projections outperform both baseline methods
  - Just projection graph already performs quite well
  - Projections on the URL graph perform better

Attributes	URL Graph	Domain Graph
Marginals	0.55	0.55
RankNet	0.63	0.60
Projection	0.80	0.64
Connection	0.79	0.66
Projection + Connection	0.82	0.69
All	**0.83**	**0.71**

Search quality: the model

- The learned model shows graph properties of good result sets
- Good result sets have:
  - Search result nodes are hub nodes in the graph (have large degrees)
  - Small connector node degrees
  - Big connected component
  - Few isolated nodes in projection graph
  - Few connector nodes

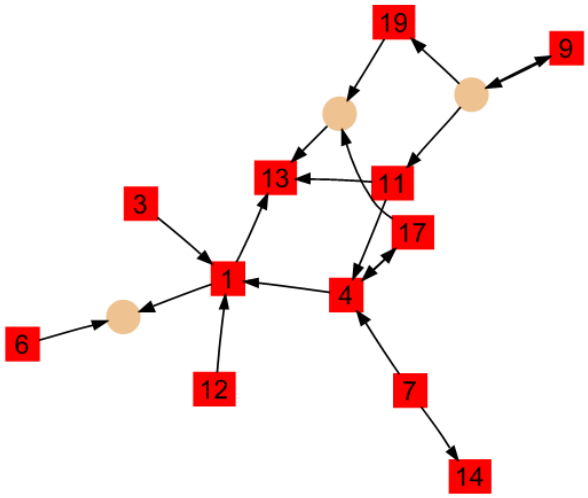


Predict user behavior

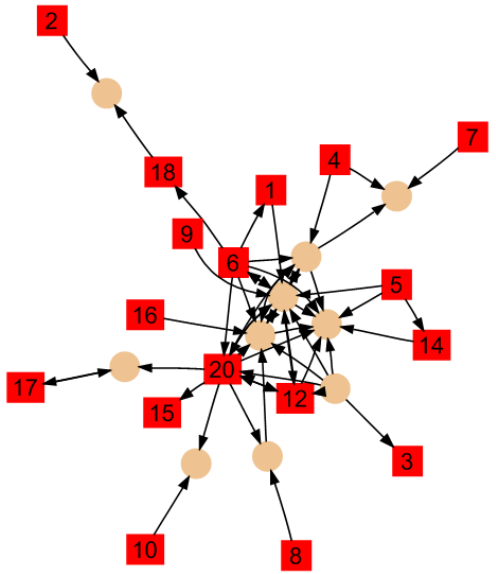
- Dataset
  - Query logs for 6 weeks
  - 35 million unique queries, 80 million total query reformulations
  - We only take queries that occur at least 10 times
  - This gives us 50,000 queries and 120,000 query reformulations
- Task
  - Predict whether the query is going to be reformulated

Query reformulation: the task

- Given a query and corresponding projection and connection graphs
- Predict whether query is likely to be reformulated



Query not likely to be reformulated



Query likely to be reformulated

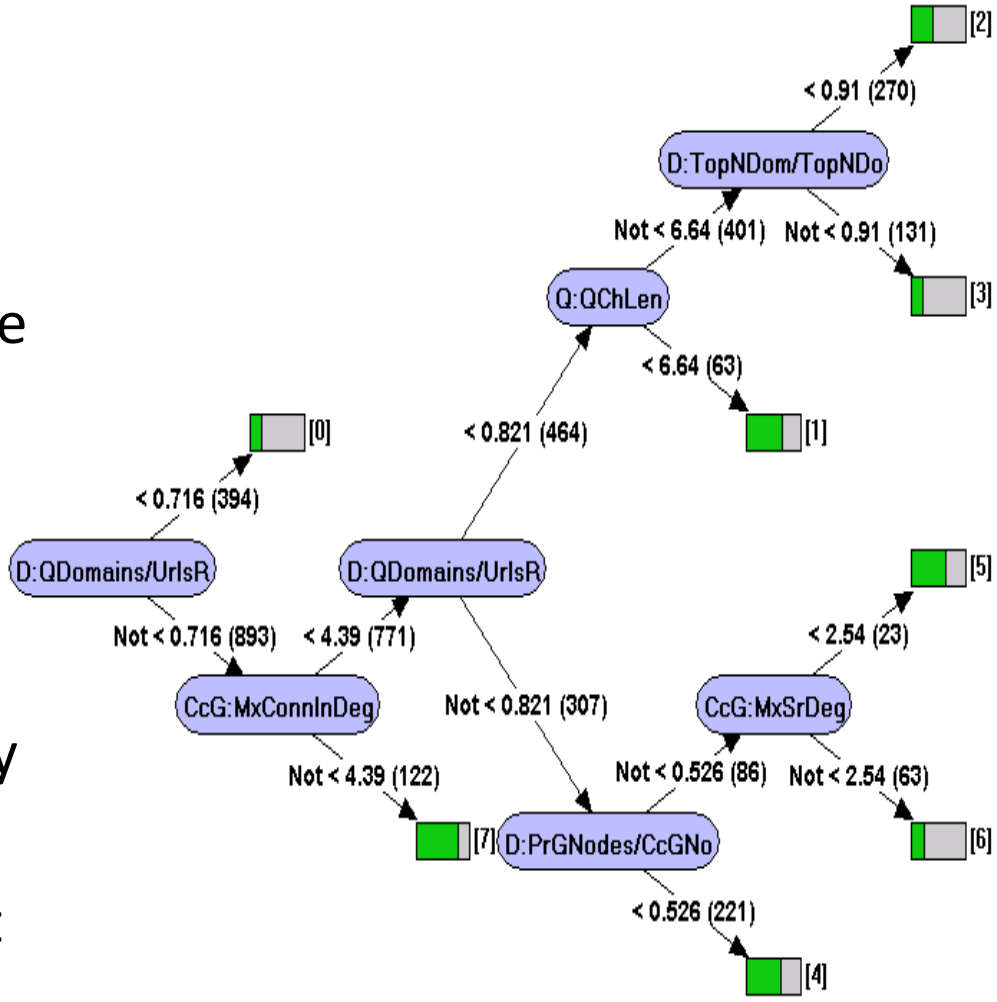
Query reformulation: results

- Observations:
  - Gradual improvement as using more features
  - Using Connection graph features helps
  - URL graph gives better performance
- We can also predict type of reformulation (specialization vs. generalization) with 0.80 accuracy

Attributes	URL Graph	Domain Graph
Marginals	0.54	0.54
Projection	0.59	0.58
Connection	0.63	0.59
Projection + Connection	0.63	0.60
All	**0.71**	**0.67**

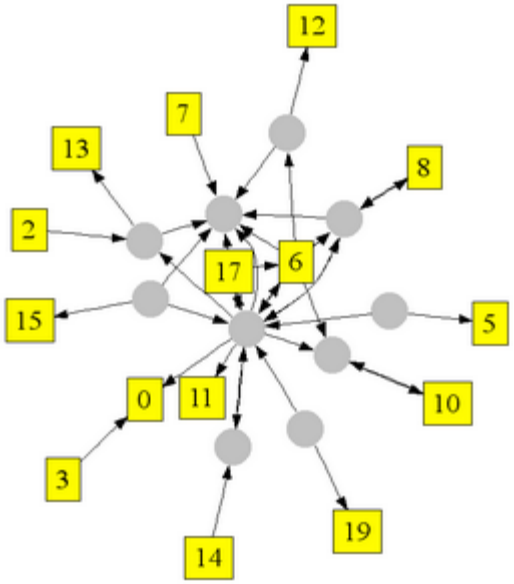
Query reformulation: the model

- Queries likely to be reformulated have:
  - Search result nodes have low degree
  - Connector nodes are hubs
  - Many connector nodes
  - Results came from many different domains
  - Results are sparsely knit



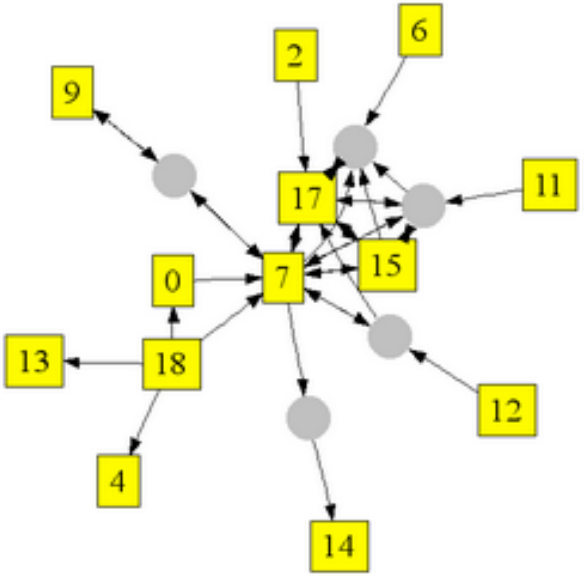
Query transitions

- Predict if and how will user transform the query



Q: Strawberry shortcake

transition



Q: Strawberry shortcake pictures

Query transition

- With 75% accuracy we can say whether a query is likely to be reformulated:
  - Def: Likely reformulated $p(\text{reformulated}) > 0.6$
- With 87% accuracy we can predict whether observed transition is specialization or generalization
- With 76% we can predict whether the user will specialize or generalize

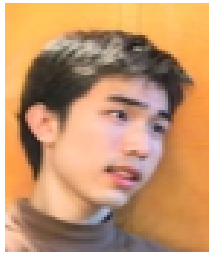
Conclusion

- We introduced Web projections
  - A general approach of using context-sensitive sets of web pages to focus attention on relevant subset of the web graph
  - And then using rich graph-theoretic features of the subgraph as input to statistical models to learn predictive models
- We demonstrated Web projections using search result graphs for
  - Predicting result set quality
  - Predicting user behavior when reformulating queries

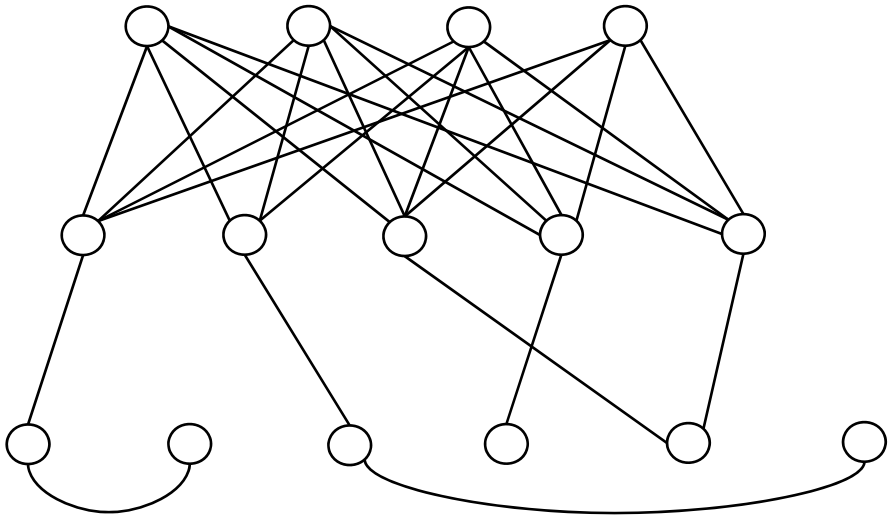
Fraud detection on e-bay

How to find fraudsters on e-bay?

E-bay Fraud detection



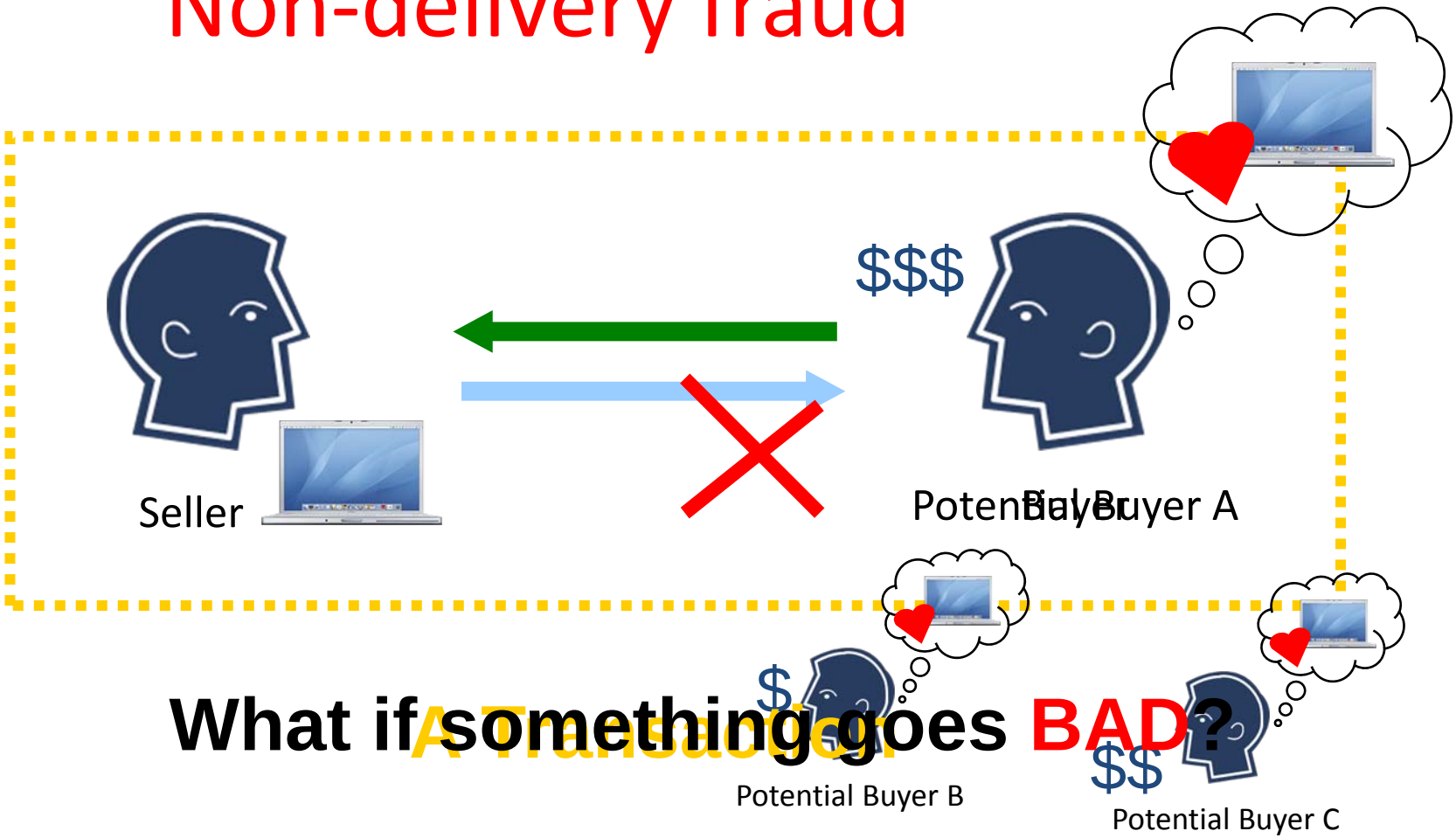
Polo Chau & Shashank Pandit, CMU



•“non-delivery” fraud: seller takes $$ and disappears

Online Auctions: How They Work

Non-delivery fraud



What if something goes BAD?

Modeling Fraudulent Behavior (contd.)

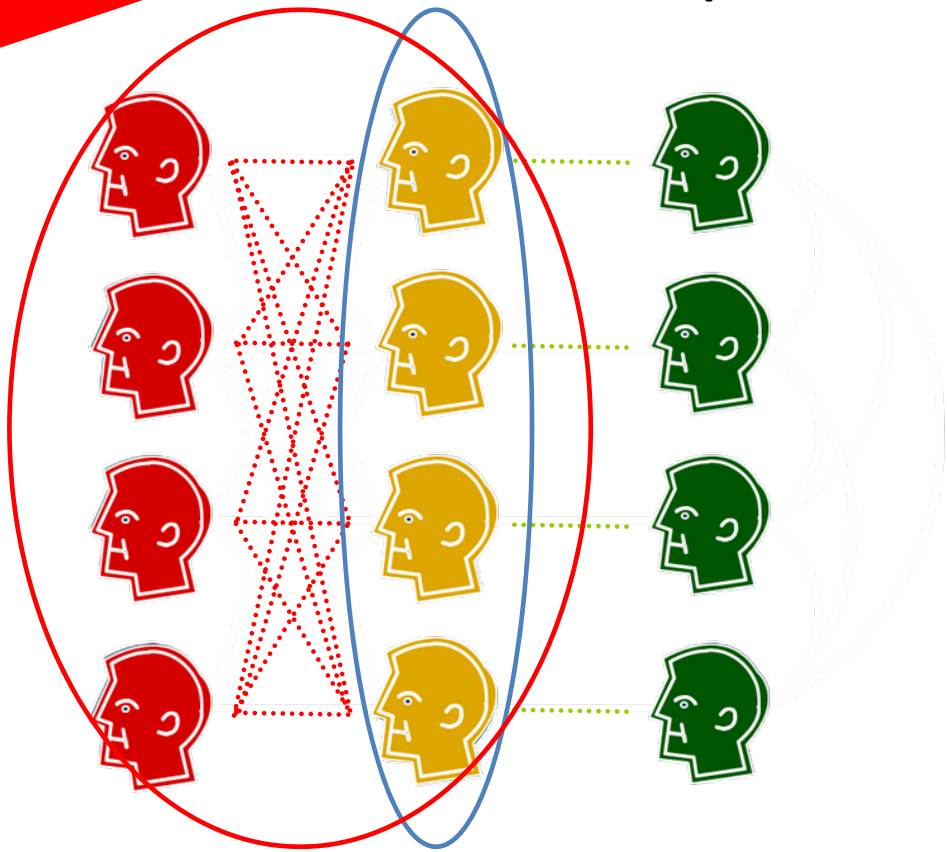
- How would fraudsters behave in this graph?
  - interact closely with other fraudsters
  - fool reputation-based systems

- Wow! ould lead to nice an
  detec ques of fraudsters ..
- Not qui
  - experiments with a real eBay dataset showed they rarely form cliques

Modeling Fraudulent Behavior

Bipartite Core

...do fraudsters operate?



= fraudster

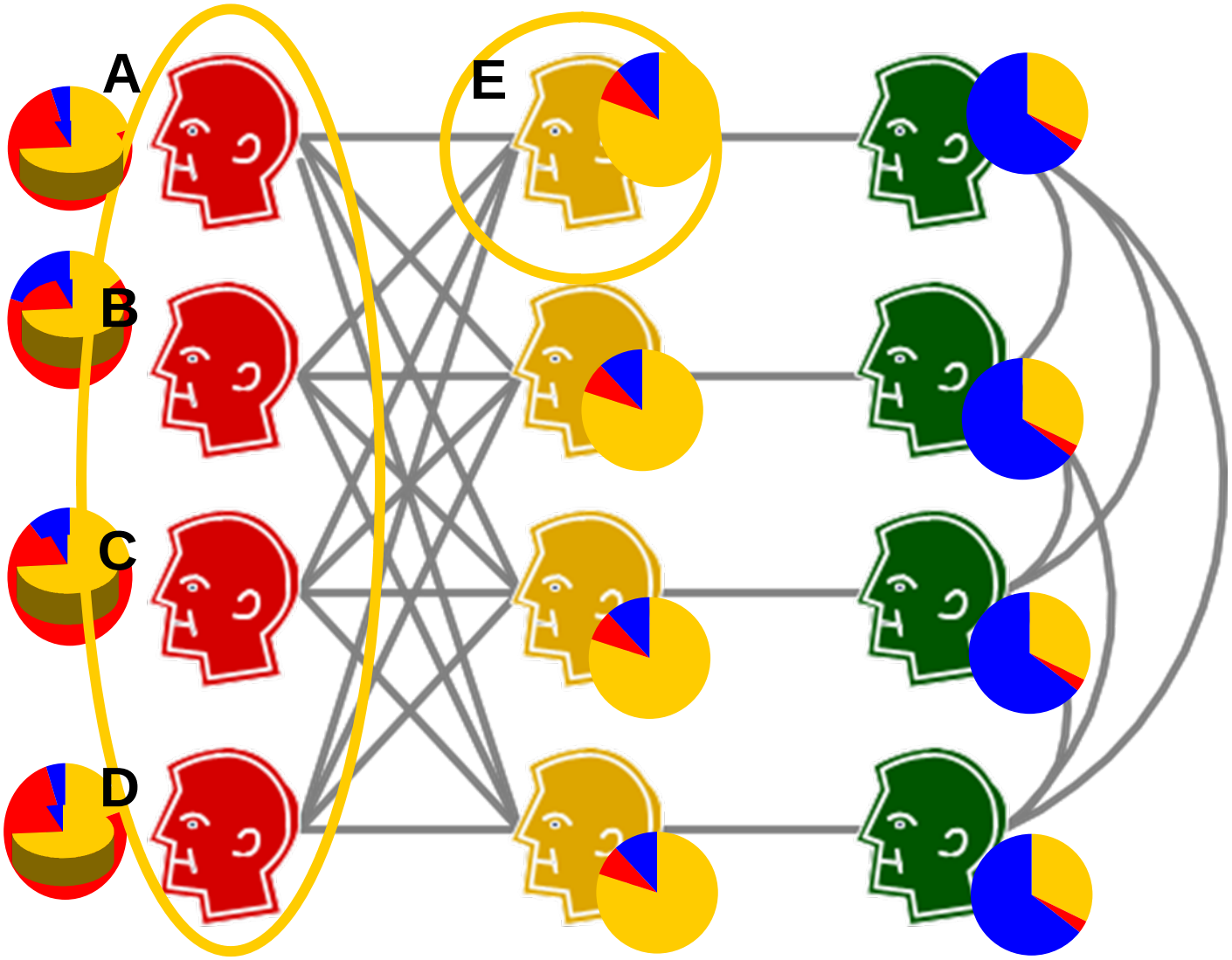
= accomplice

= honest

Modeling Fraudulent Behavior

- The 3 roles
  - Honest
    - people like you and me
  - Fraudsters
    - those who actually commit fraud
  - Accomplices
    - erstwhile behave like honest users
    - accumulate feedback via low-cost transactions
    - secretly boost reputation of fraudsters (e.g., occasionally trading expensive items)

Belief Propagation

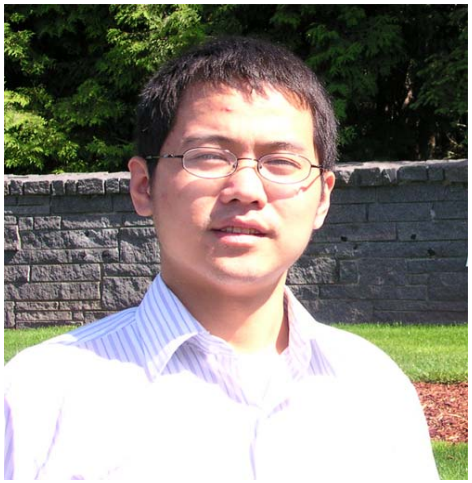


Center piece subgraphs

What is the best explanatory path between the nodes in a graph?

MasterMind – 'CePS'

- Hanghang Tong, KDD 2006
- htong <at> cs.cmu.edu

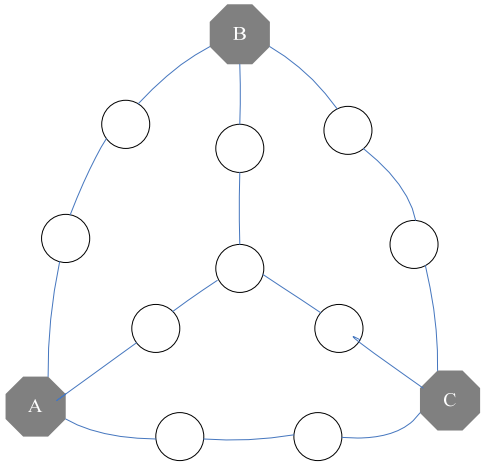


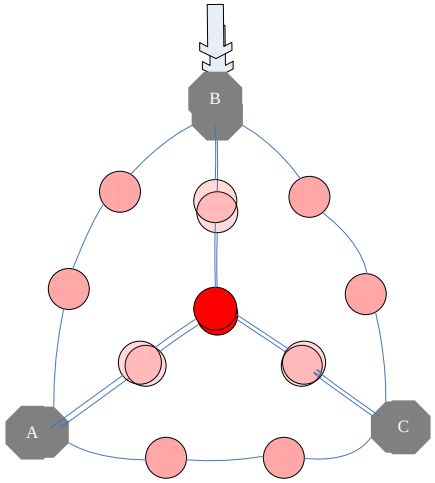
Center-Piece Subgraph(Ceps)

- **Given** Q query nodes
- **Find** Center-piece ($\leq b$

- App.
  - Social Networks
  - Law Inforcement, …

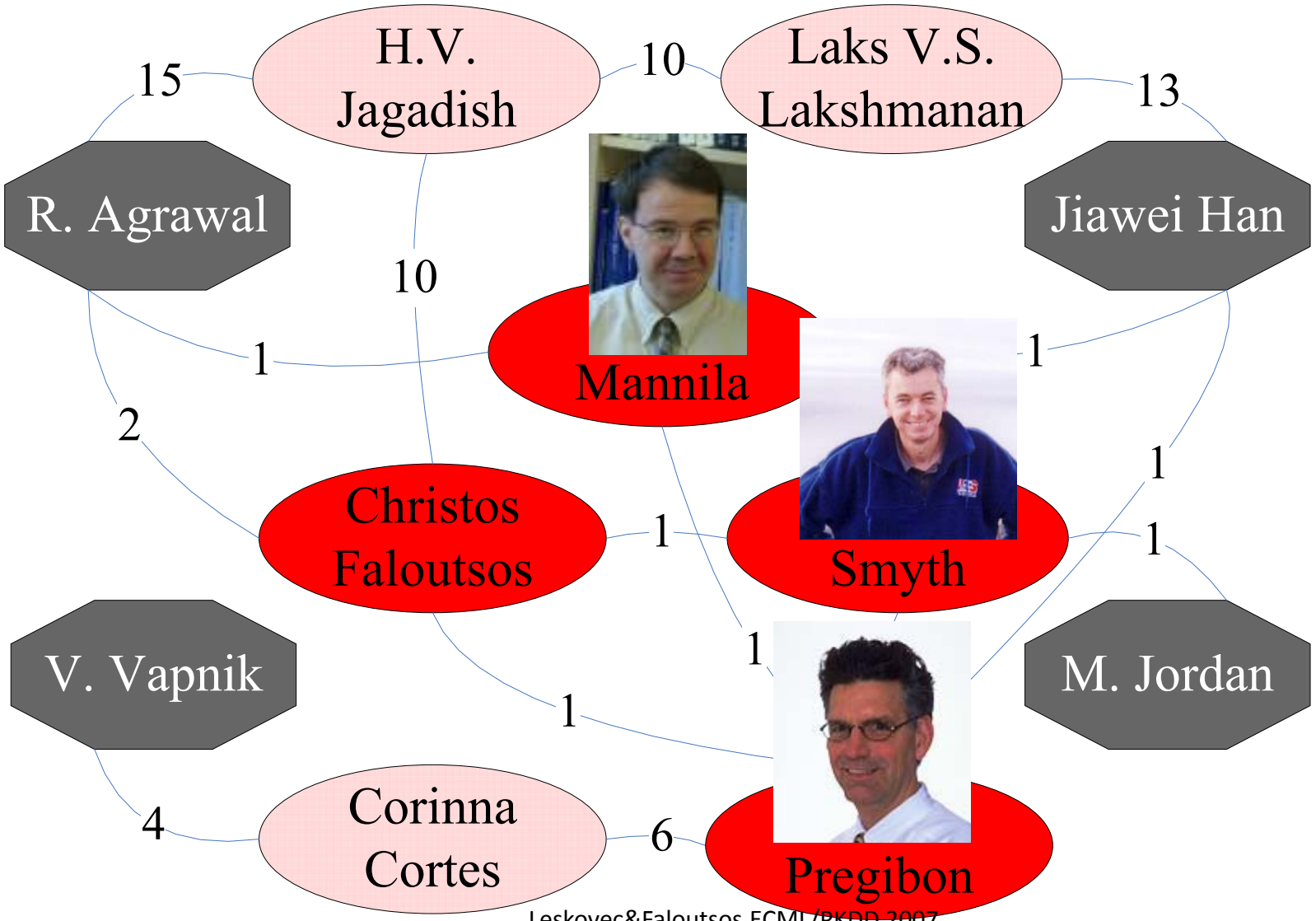
- Idea:
  - Proximity -> random walk with restarts

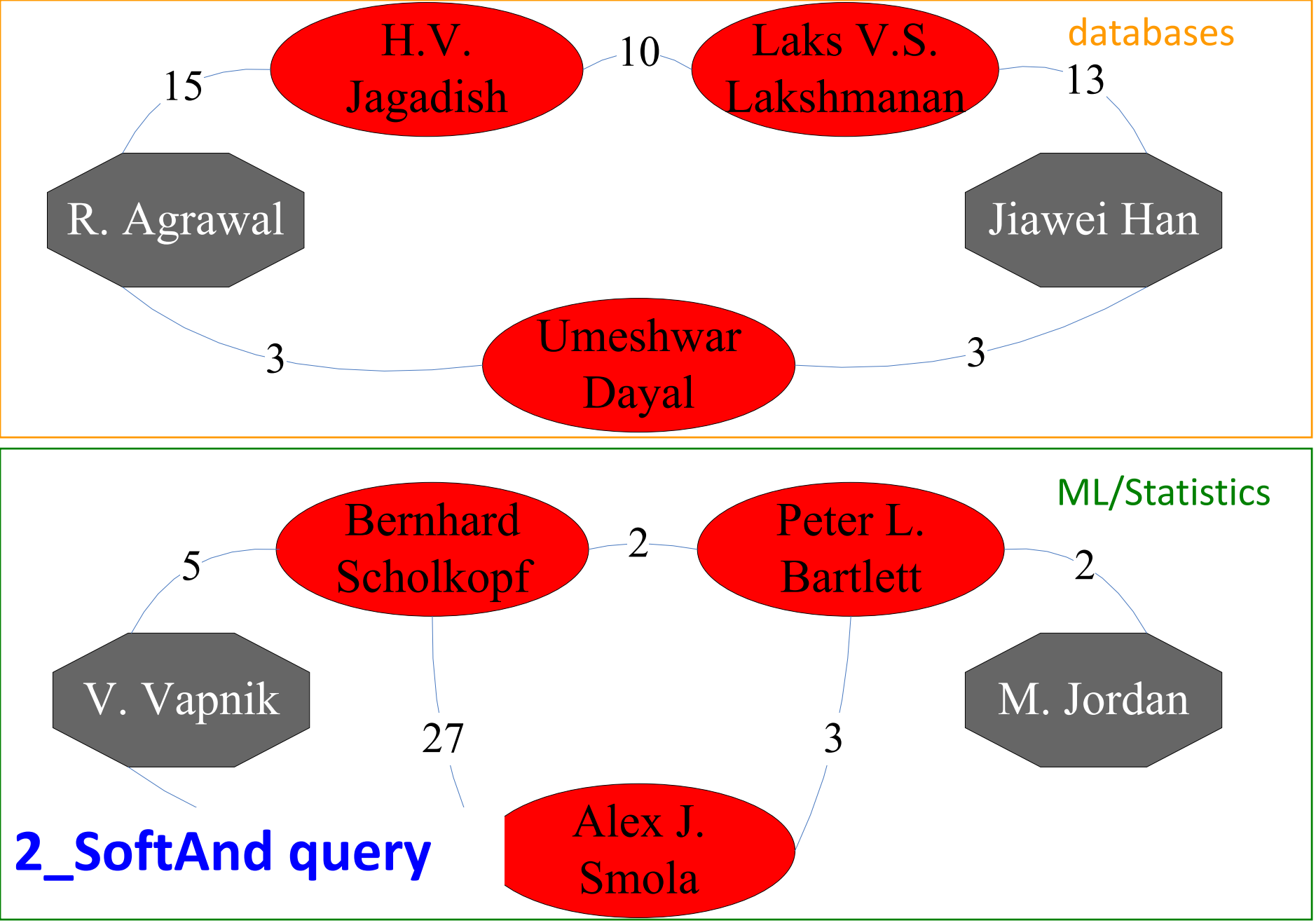




Case Study: AND query

Case Study: AND query

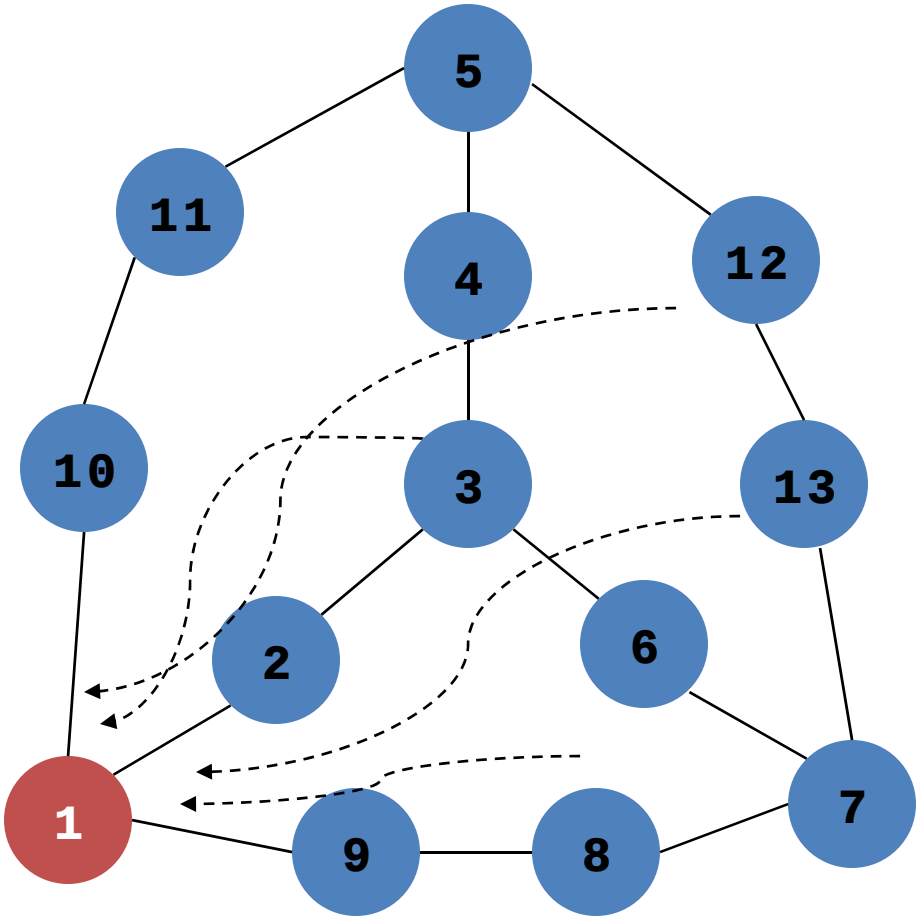



databases
H.V. Jagadish
10
Laks V.S. Lakshmanan
15
13
R. Agrawal
Jiawei Han
3
Umeshwar Dayal
3
ML/Statistics
Bernhard Scholkopf
2
Peter L. Bartlett
5
2
V. Vapnik
M. Jordan
27
3
Alex J. Smola
2_SoftAnd query

Details

- Main idea: use random walk with restarts, to measure 'proximity' *p(i,j)* of node *j* to node *i*

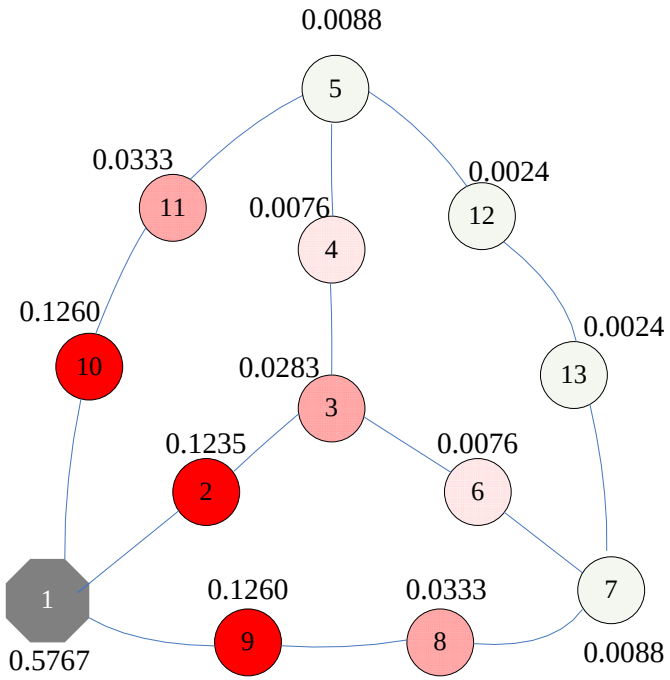
Example



Prob (RW will finally stay at j)

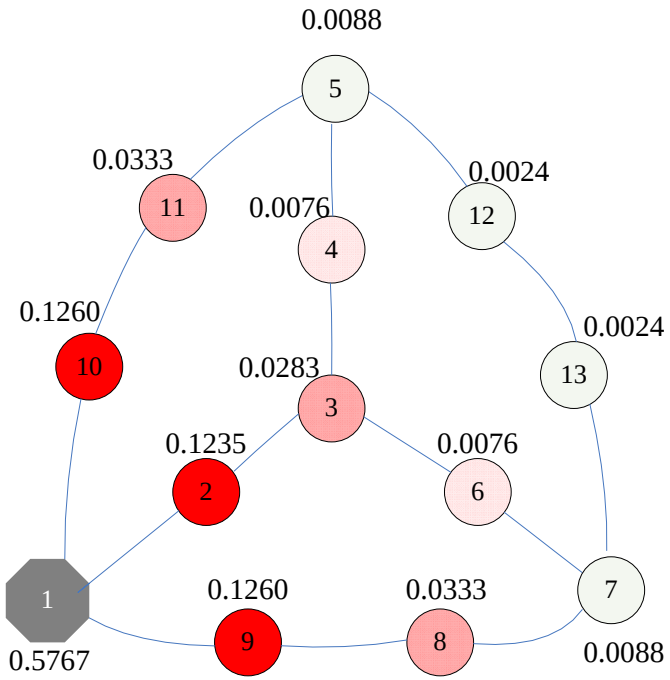
- Starting from 1
- Randomly to neighbor
- Some p to return to 1

Individual Score Calculation



	Q1	Q2	Q3
Node 1	0.5767	0.0088	0.0088
Node 2	**0.1235**	0.0076	0.0076
Node 3	0.0283	0.0283	0.0283
Node 4	0.0076	**0.1235**	0.0076
Node 5	0.0088	0.5767	0.0088
Node 6	0.0076	0.0076	**0.1235**
Node 7	0.0088	0.0088	0.5767
Node 8	0.0333	0.0024	**0.1260**
Node 9	**0.1260**	0.0024	0.0333
Node 10	**0.1260**	0.0333	0.0024
Node 11	0.0333	**0.1260**	0.0024
Node 12	0.0024	**0.1260**	0.0333
Node 13	0.0024	0.0333	**0.1260**

Individual Score Calculation

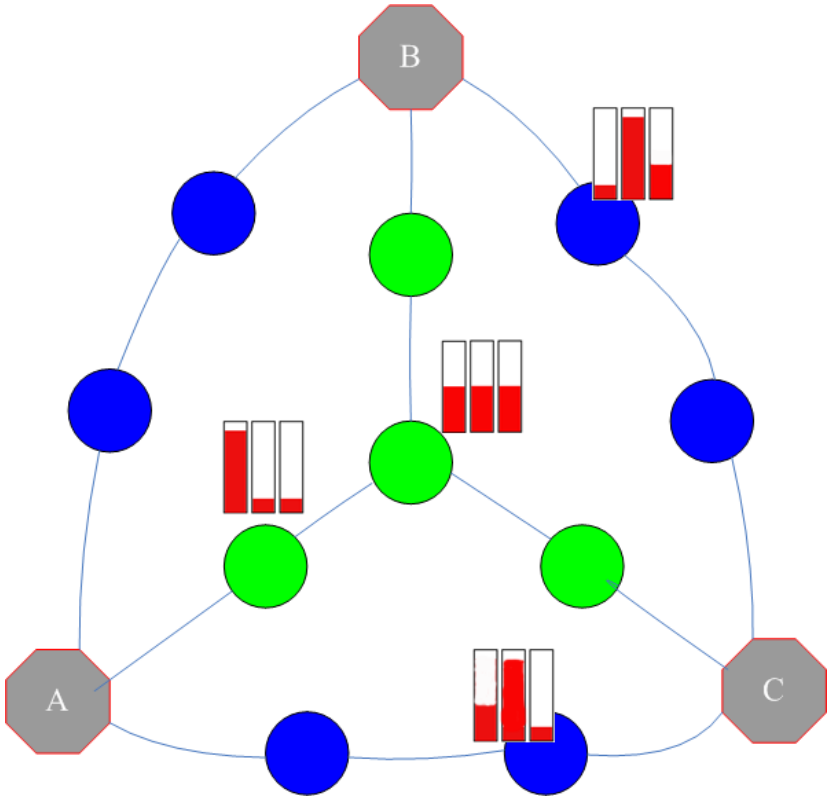


	Q1	Q2	Q3
Node 1	0.5767	0.0088	0.0088
Node 2	**0.1235**	0.0076	0.0076
Node 3	0.0283	0.0283	0.0283
Node 4	0.0076	**0.1235**	0.0076
Node 5	0.0088	0.5767	0.0088
Node 6	0.0076	0.0076	**0.1235**
Node 7	0.0088	0.0088	0.5767
Node 8	0.0333	0.0024	**0.1260**
Node 9	**0.1260**	0.0024	0.0333
Node 10	**0.1260**	0.0333	0.0024
Node 11	0.0333	**0.1260**	0.0024
Node 12	0.0024	**0.1260**	0.0333
Node 13	0.0024	0.0333	**0.1260**

Individual Score matrix

AND: Combining Scores

- Q: How to combine scores?
- A: Multiply
- …= prob. 3 random particles coincide on node *j*



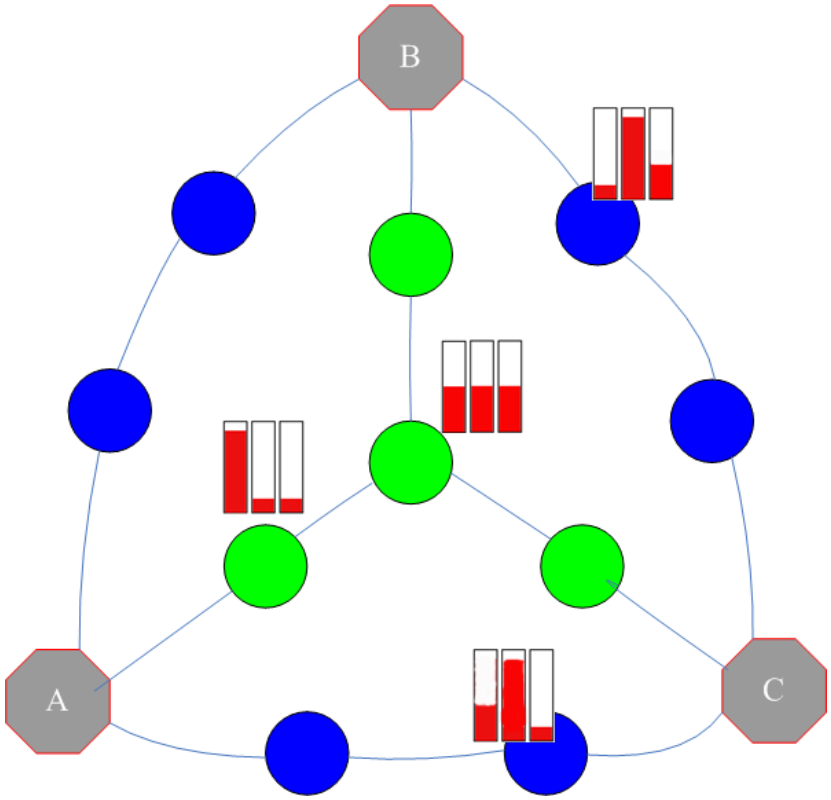
K_SoftAnd: Combining Scores

details

Generalization – SoftAND:

We want nodes close to *k* of *Q* (*k*<*Q*) query nodes.

Q: How to do that?



K_SoftAnd: Combining Scores

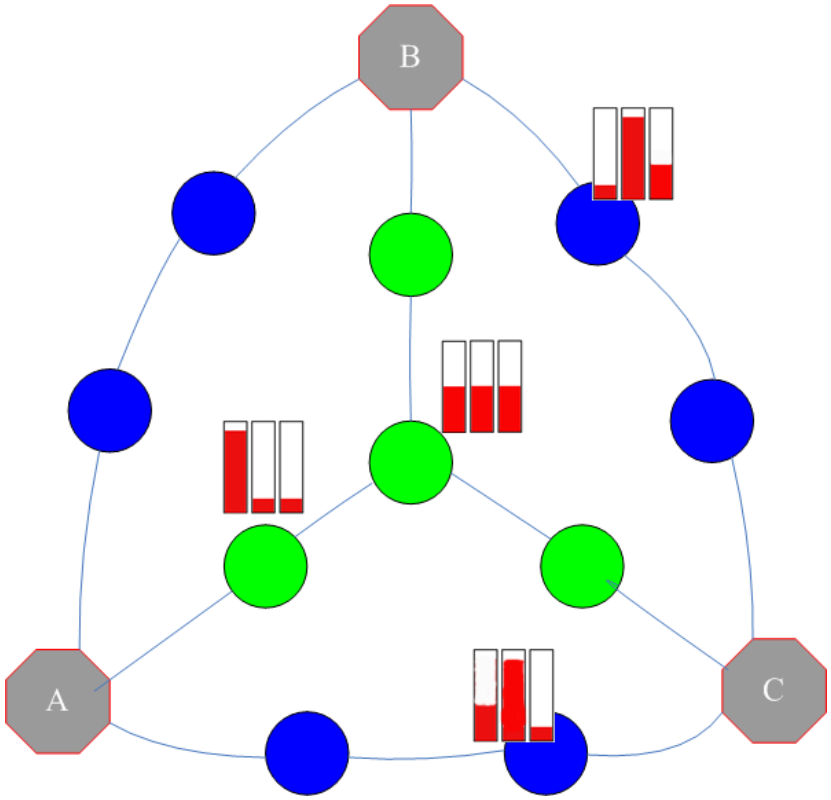
details

Generalization – softAND:

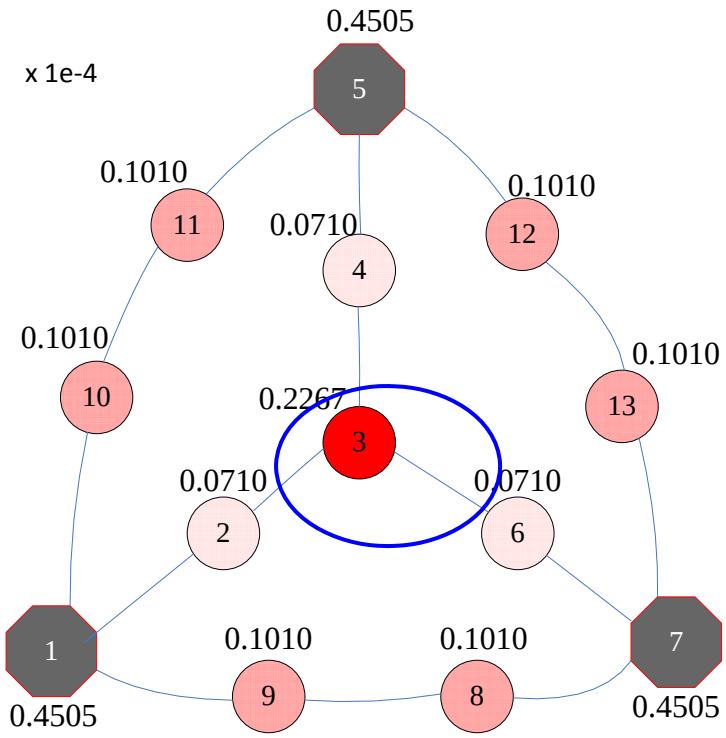
We want nodes close to *k* of *Q* (*k*<*Q*) query nodes.

Q: How to do that?

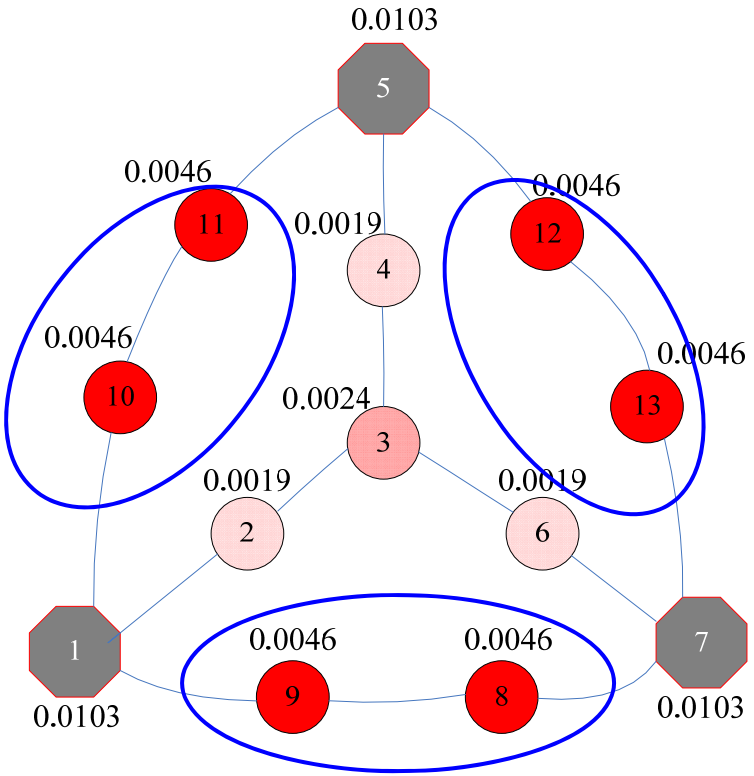
A: Prob(at least *k*-out-of-*Q* will meet each other at j)



AND query vs. K_SoftAnd query

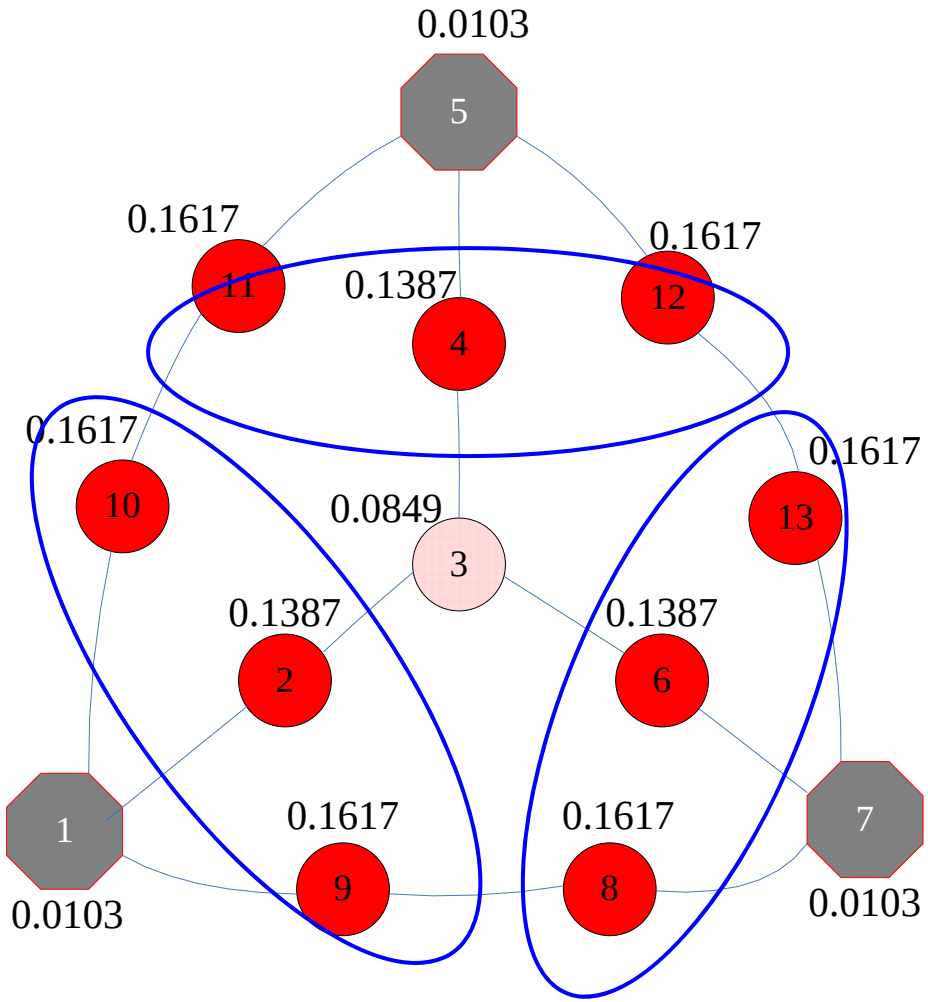


And Query



2_SoftAnd Query

1_SoftAnd query = OR query

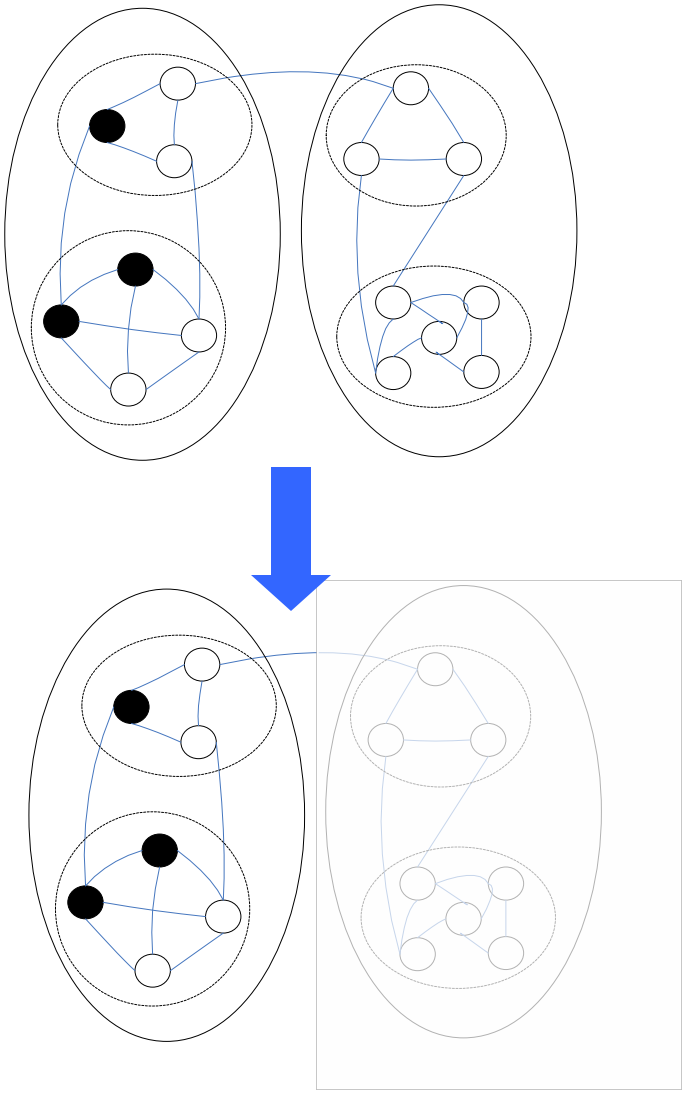


Challenges in Ceps

- Q1: How to measure the importance?
  - A: RWR
- Q2: How to do it efficiently?

Graph Partition: Efficiency Issue

- Straightforward way
  - solve a linear system:
  - time: linear to # of edges
- Observation
  - Skewed dist.
  - communities
- How to exploit them?
  - Graph partition



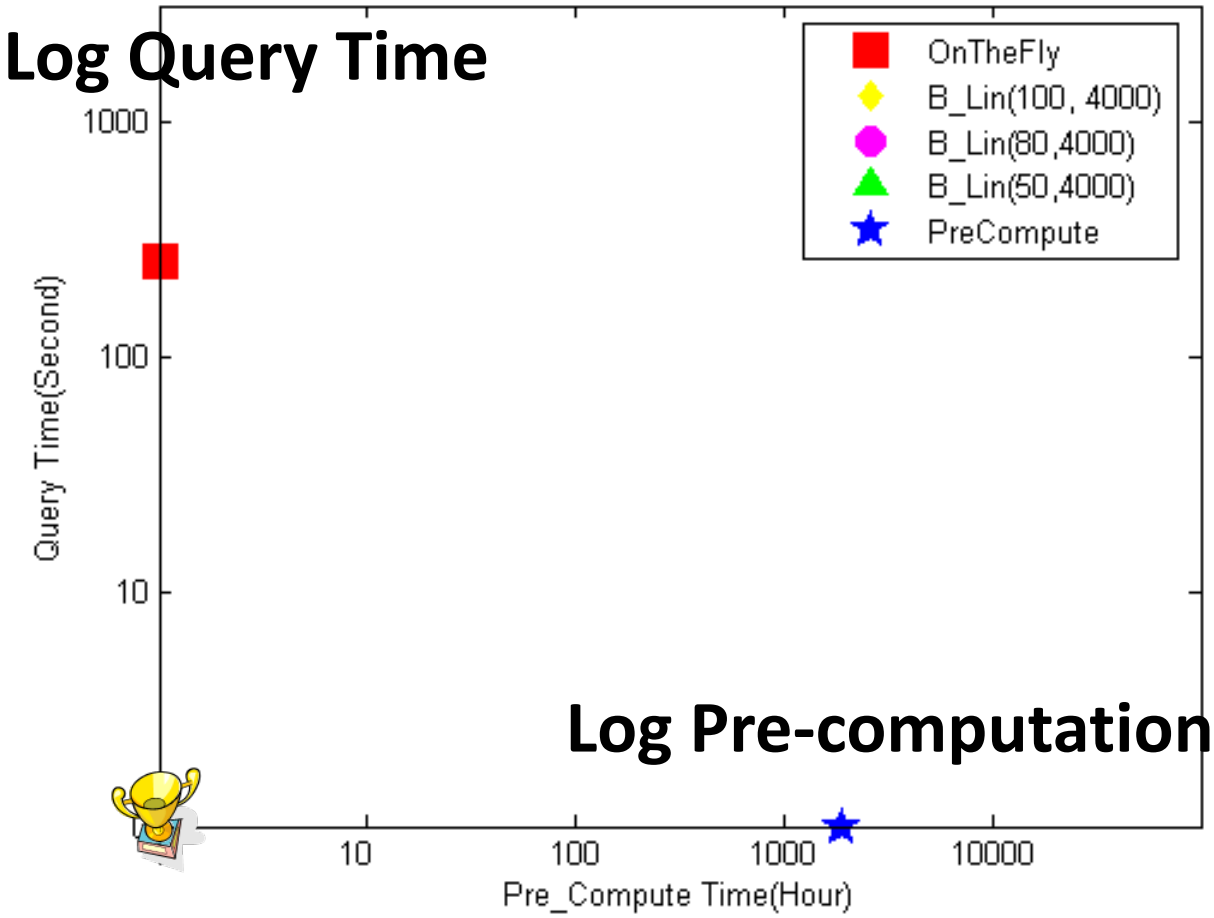
Even better:

- We can correct for the deleted edges (Tong+, ICDM'06, best paper award)

Experimental Setup

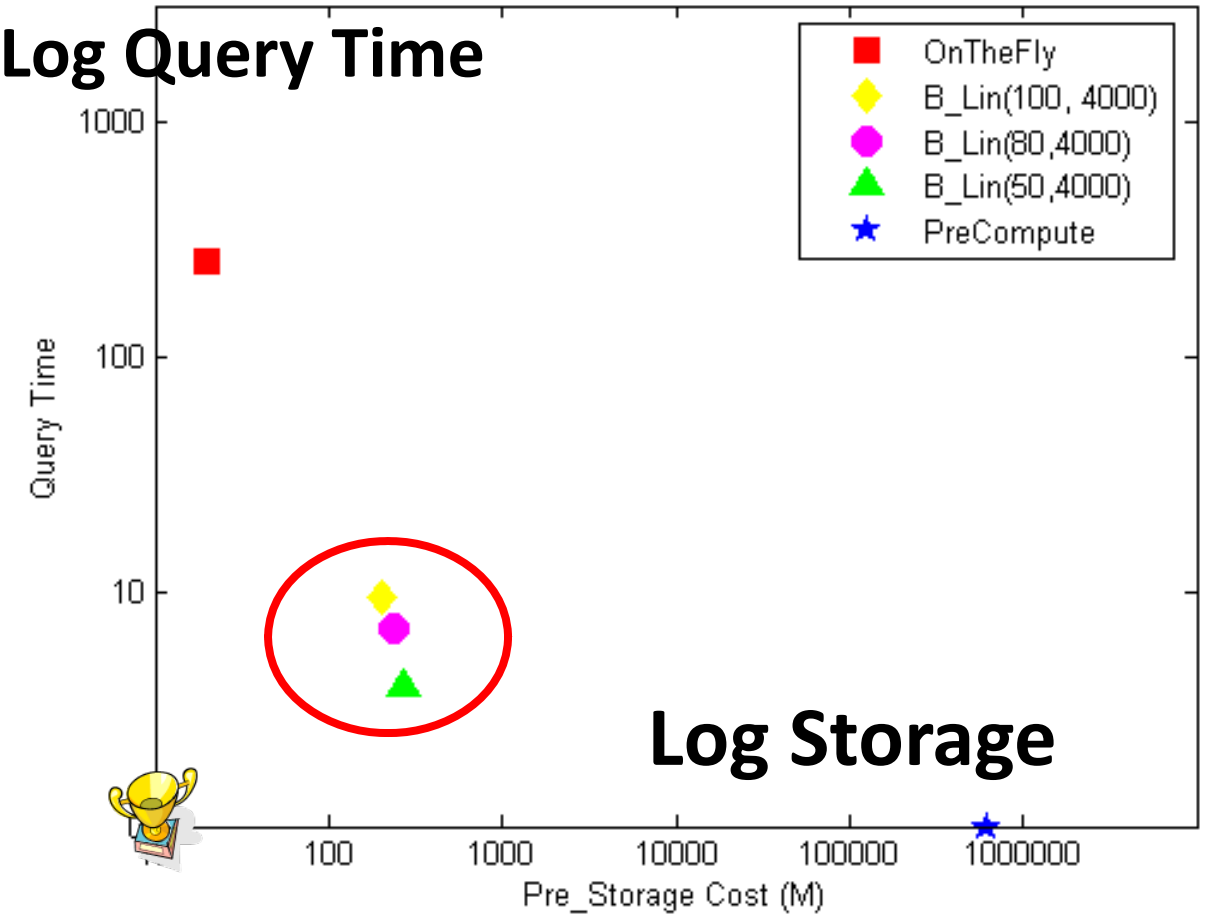
- Dataset
  - DBLP/authorship
  - Author-Paper
  - 315k nodes
  - 1.8M edges

Query Time vs. Pre-Computation Time



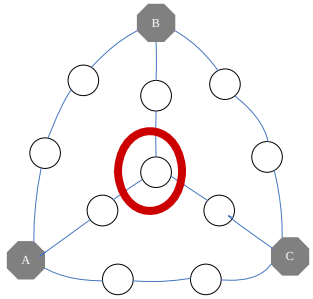
- Quality: 90%+
- On-line:
  - Up to 150x speedup
- Pre-computation:
  - Two orders saving

Query Time vs. Storage



- Quality: 90%+
- On-line:
  - Up to 150x speedup
- Pre-storage:
  - Three orders saving

Conclusions



- Q1:How to measure the importance?
- A1: RWR+K_SoftAnd
- Q2: How to find connection subgraph?
- A2:"Extract" Alg.)
- Q3:How to do it efficiently?
- A3:Graph Partition and Sherman-Morrison
  - ~90% quality
  - 6:1 speedup; 150x speedup (ICDM'06, b.p. award)

References

- Leskovec and Horvitz: Worldwide Buzz: Planetary-Scale Views on an Instant-Messaging Network, 2007
- Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan *Fast Random Walk with Restart and Its Applications* ICDM 2006.
- Hanghang Tong, Christos Faloutsos *Center-Piece Subgraphs: Problem Definition and Fast Solutions,* KDD 2006
- Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos: *NetProbe: A Fast and Scalable System for Fraud Detection in Online Auction Networks,* WWW 2007.