Network Algorithms for Homeland Security

Mark Goldberg and Malik Magdon-Ismail
Rensselaer Polytechnic Institute

September 27, 2004.

Collaborators

J. **Baumes**, M. **Krishmamoorthy**, N. **Preston**, W. **Wallace**.

Partially supported by NSF grants: 0324947, 0346341.

Motivation

– Social communities communicate in structured ways.
(**spatial correlation**)

– A group planning an activity communicates in order to coordinate.
(**spatial and temporal correlation**)

We should be able to exploit the structure in such communications to discover social comunities and groups that may be planning some activity.

Communication Networks

Communication Data:

– Sequence of actor pairs, and time/content of communication.

Communication Cycle:

– A time period over which communications are aggregated.

Communication Graph for a Communication Cycle:

– Actors are nodes.

– Edge between two actors if a communication occured.

Many cycles $\Longrightarrow$ time series of communication graphs.

Non-Semantic Analysis

– Semantic information is informative but **computationally intensive** to analyse for large communication networks.

– Semantic information can be **misleading** if cryptographic protocols are being used.

– Semantic information may be **unavailable**, especially on account of privacy constraints.

What can we accomplish without semantic-analysis?

Outline

Part I: Hidden Groups

- Example
- Model
- Algorithms
- Experiments
- Extensions and Ongoing work.

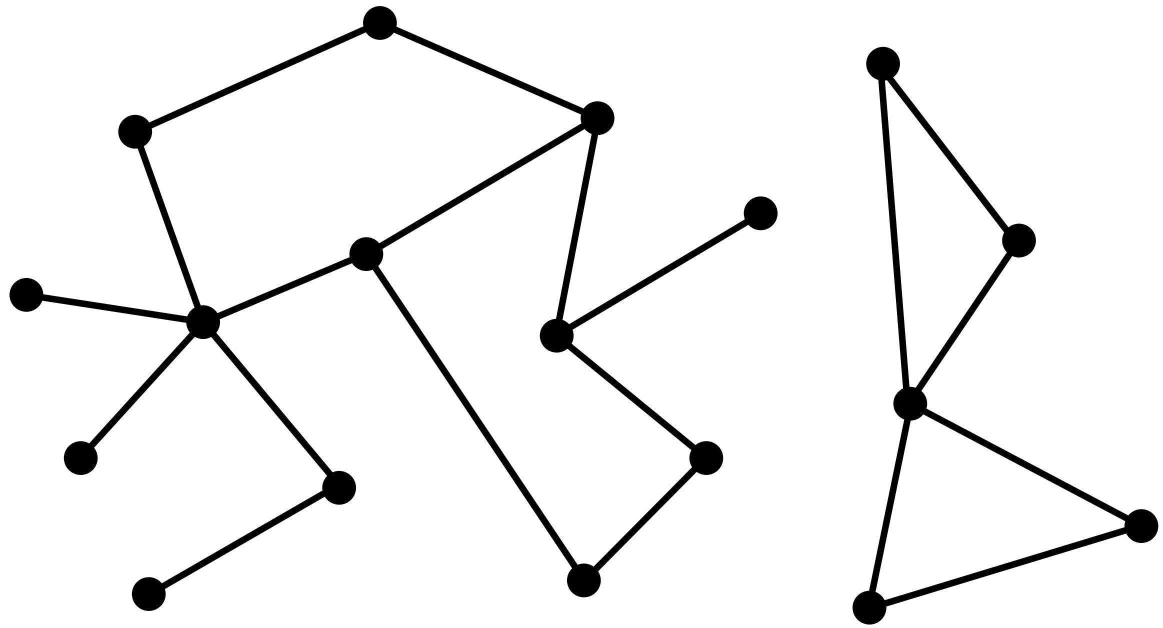
Part II: Overlapping Communities

- Example
- Model
- Algorithms
- Experiments

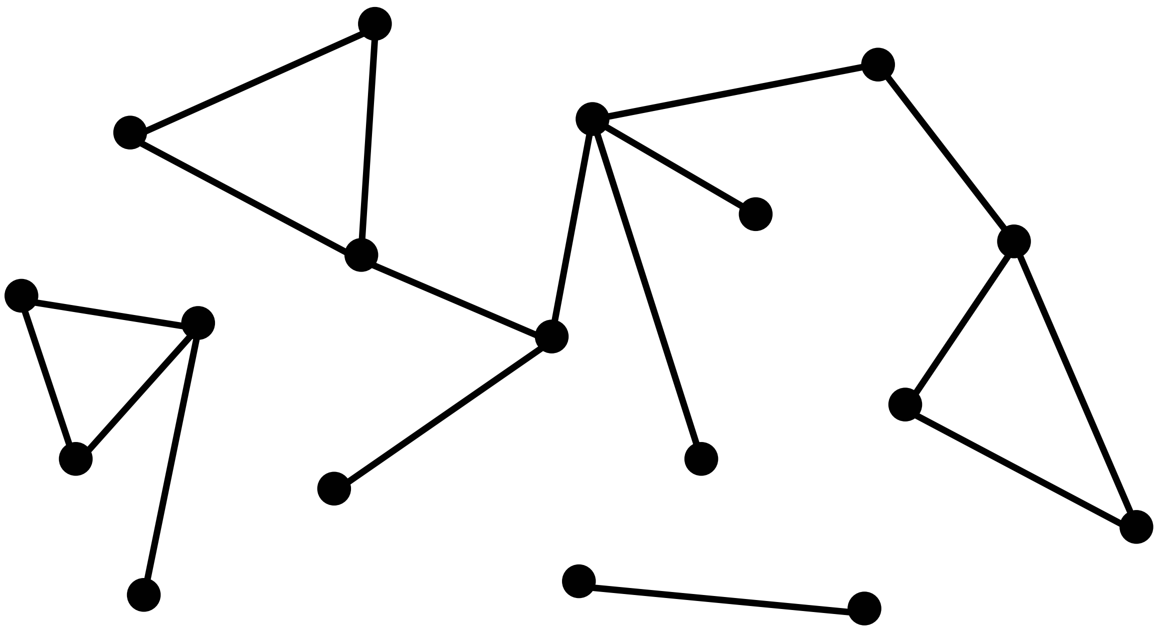
PART I

Hidden Groups

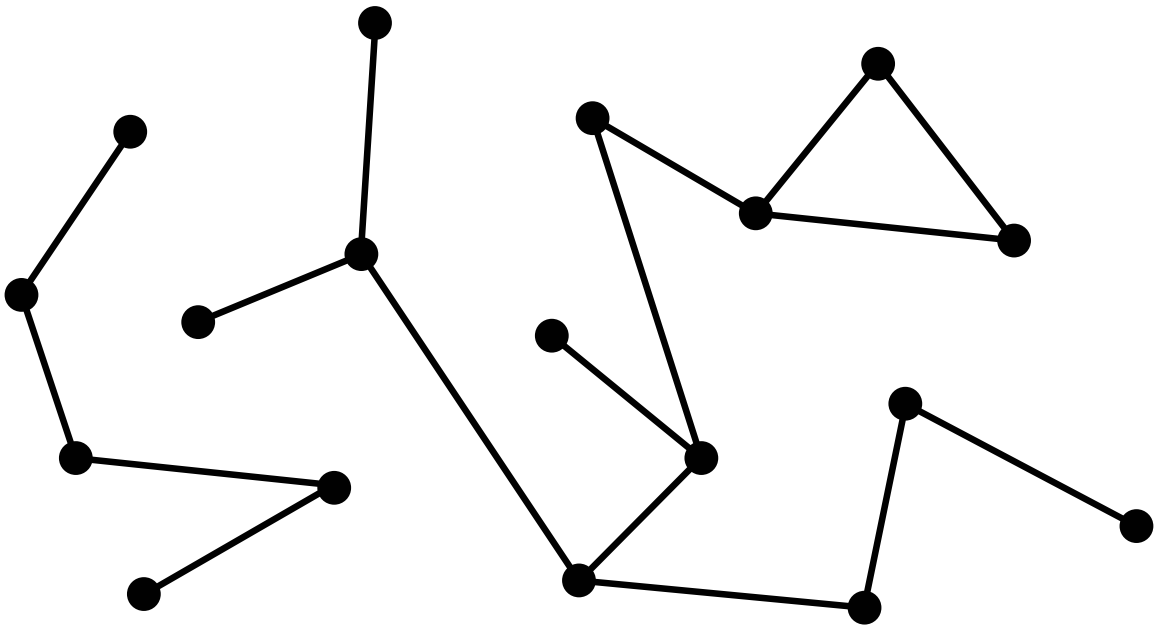
Example



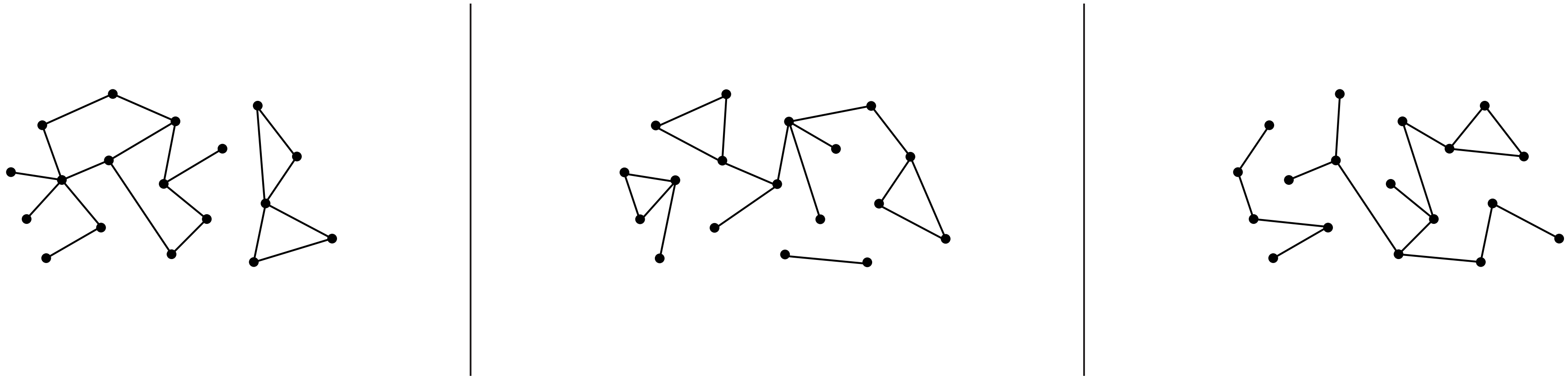
Example



Example

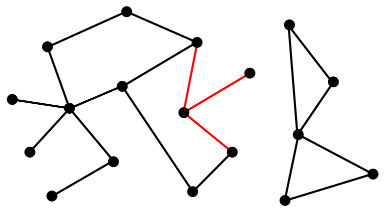


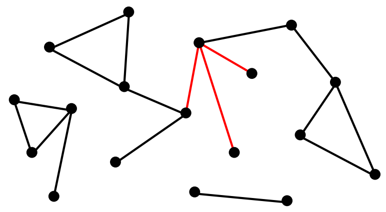
Example

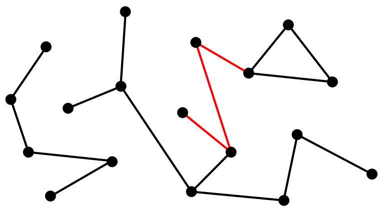


Where is the hidden group?

Example







Where is the hidden group?

What is a Hidden Group?

A **Hidden Group** is a group of actors that is **planning** some activity. The group may or may not actively try to conceal their planning, eg:

- group planning a Sunday afternoon picnic, or PTA meeting.
- group planning a malicious terrorist act.

Their planning related communications are embedded in the (random) **background communications** and hence obscured.

Planning related communications must occur for the activity to succeed; background communications tend to be random.

What is "Planning"?

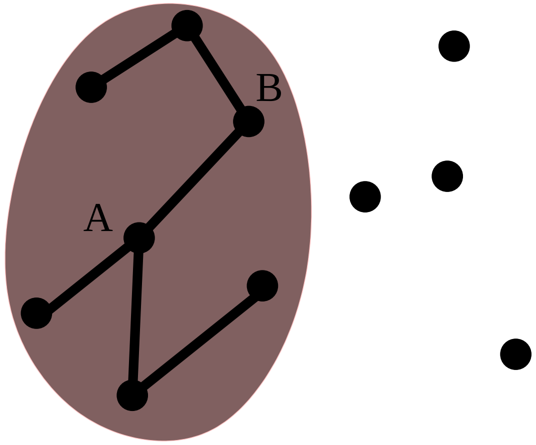
All members of the group need to exchange information during each communication cycle.

Implication on the communication graph:

Connectivity

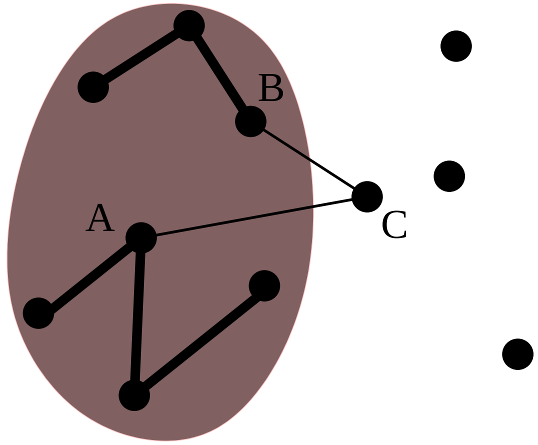
Types of Connectivity

Internal



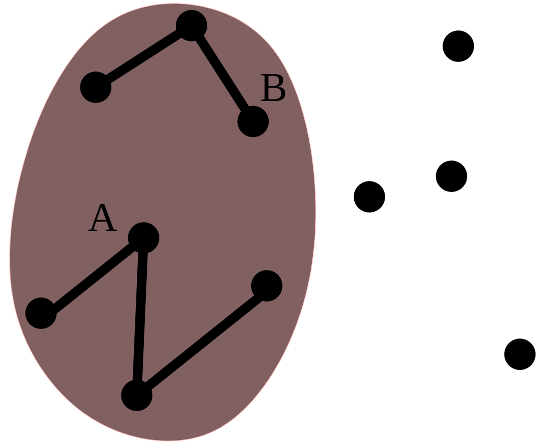
Secretive
Paranoid

External



Non-Secretive
Trusting

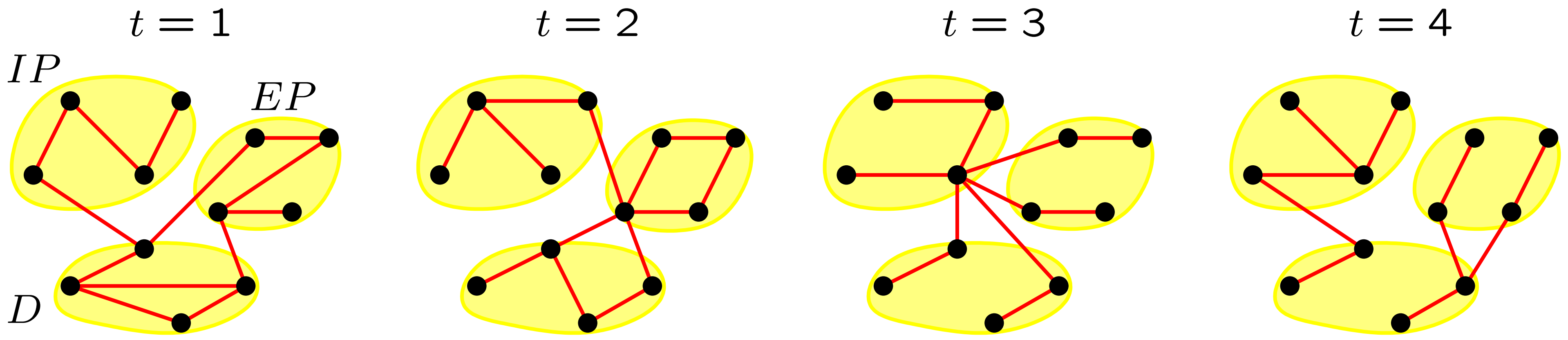
Disconnected



Not a Hidden Group

A malicious group is more likely to be secretive – **internally connected**.

Persistence



IP is **internally persistent (IP)** – internally connected in every graph.

EP is **externally persistent (EP)**.

D is not persistent.

$\boxed{\text{Hidden Group Model}}$

Internal Persistence. A (paranoid) hidden group is internally persistent

Static. The hidden group members do not change.

Problem Statement

Input: Communication graphs, $G_1, G_2, \ldots, G_T$; integer K.
Task: Is there a hidden group of size $\geq K$; find all such hidden groups.

Let E_t be the number of edges in G_t; Let $E = \sum_t E_t$ (total # edges).

Maximal Persistent Components

A persistent component C is **maximal** if any other persistent component that overlaps C is contained in C.

Observation: Maximal persistent components are disjoint.

Implication: The vertex set can be (uniquely) partitioned into maximally persistent components.

Let $\mathcal{P} = \{A_1, \ldots, A_K\}$ be such a partition.

Algorithm Ext_Persistent

Observation: A^{ext} is **EP** if it is in the same connected component in every G_t.

Implication: Maximal **EP** component $\Longrightarrow$ find maximal intersections of the connected components of $\{G_t\}$.

Computational complexity: $O(E + V \cdot T)$.

Algorithm Int_Persistent

Suppose A^{ext} is maximally **EP**.

Observation: A^{ext} can be partitioned into maximal **IP** components.

Consider the induced subgraphs $\{G_1(A^{ext}), G_2(A^{ext}), \ldots, G_T(A^{ext})\}$
Let $\mathcal{P}^{ext}$ be a partition of $\{G_t(A^{ext})\}$ into maximal **EP** components.

Lemma. A^{ext} is maximally **IP** *iff* $|\mathcal{P}^{ext}| = 1$

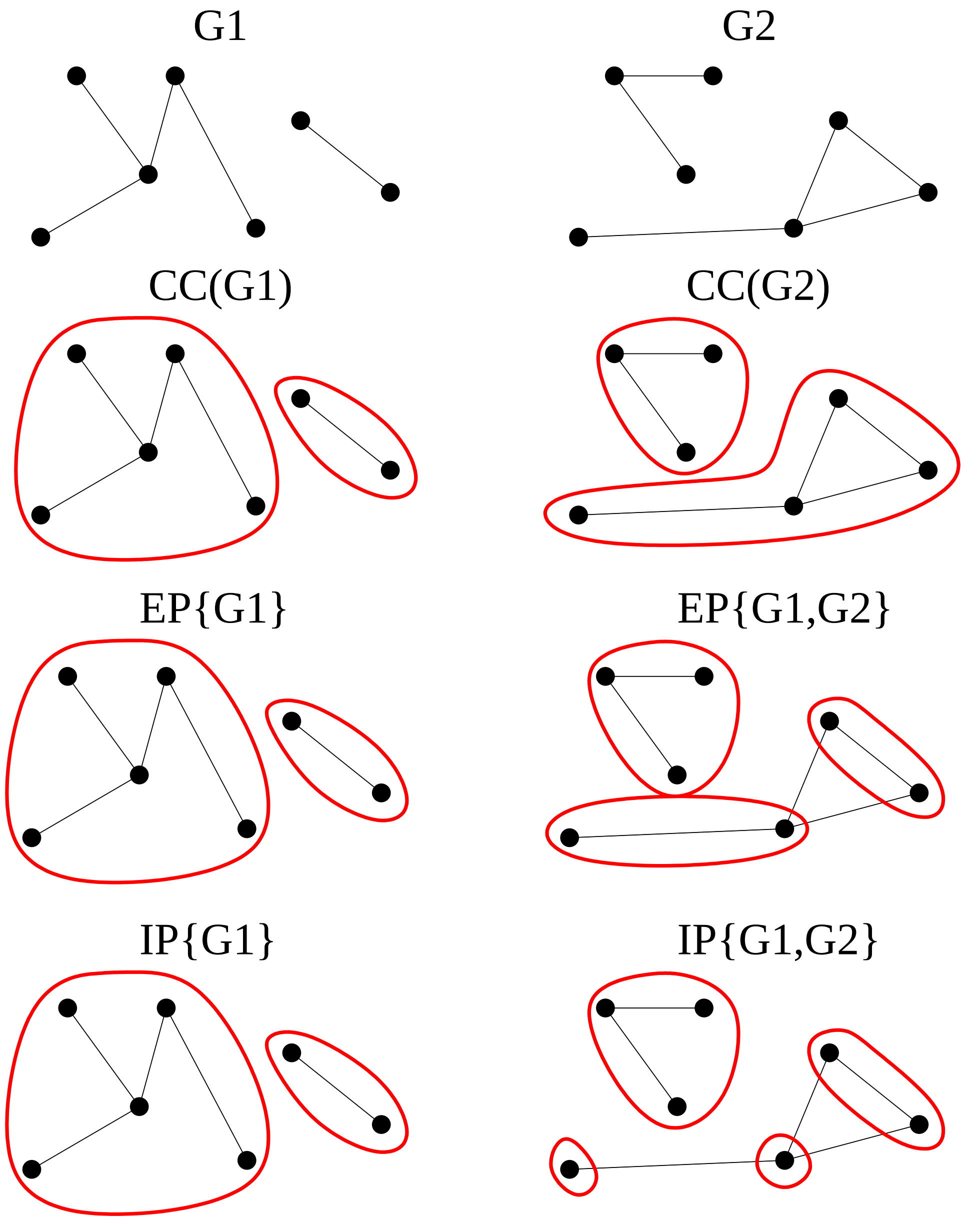
Implication: Int_Persistent can be implemented recursively on a partition obtained from Ext_Persistent.

Recursive Algorithm Int_Persistent

```
Int_Persistent({G_t}_{t=1}^T, V)
//Input: Graphs {G_t = (E_t, V)}_{t=1}^T.
//Output: A partition P = {V_j} of V.
{V_i}_{i=1}^K = Ext_Persistent({G_t}_{t=1}^T, V)
if K = 1, then
    P = {V_1};
else
    P = ∪_{k=1}^K Int_Persistent({G_t(V_k)}_{t=1}^T, V_k);
return P;
```

Computational complexity: $O(V \cdot E + V^2 \cdot T)$.

Example: Int_Persistent

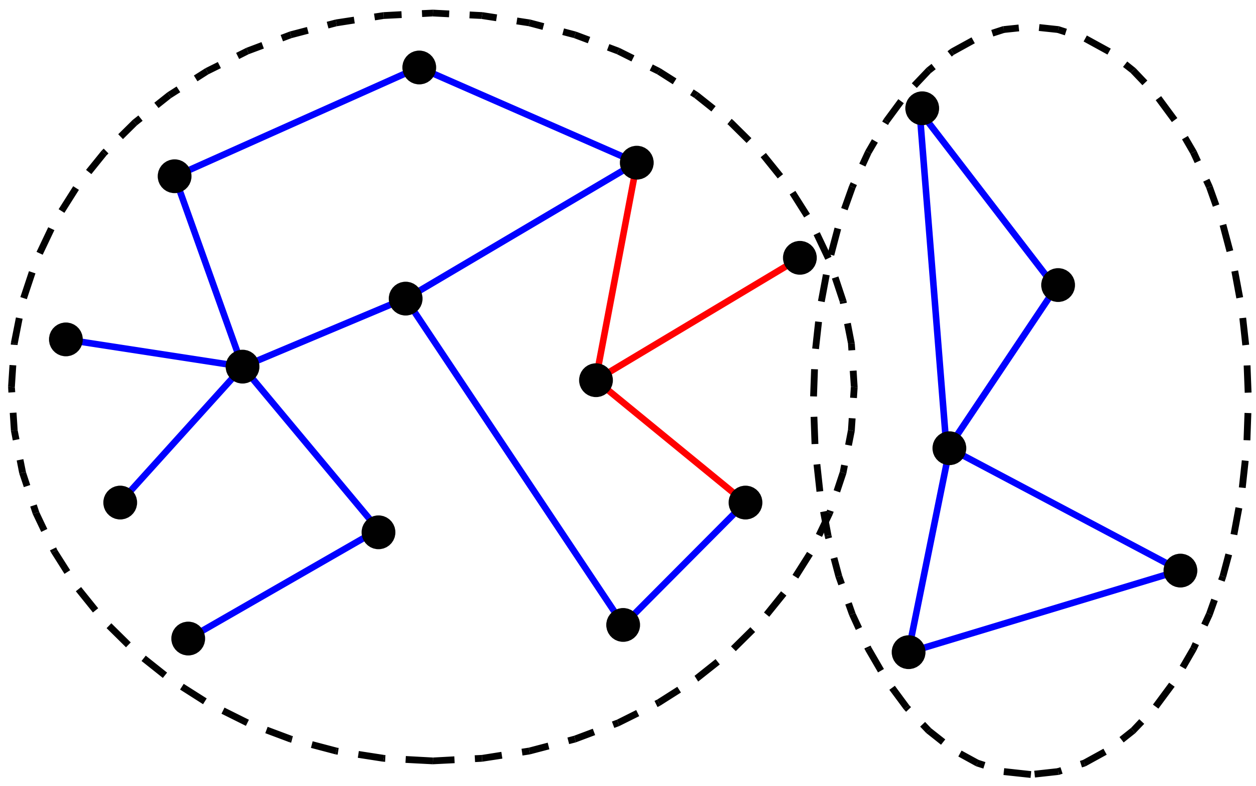


$\boxed{\textbf{Status}}$

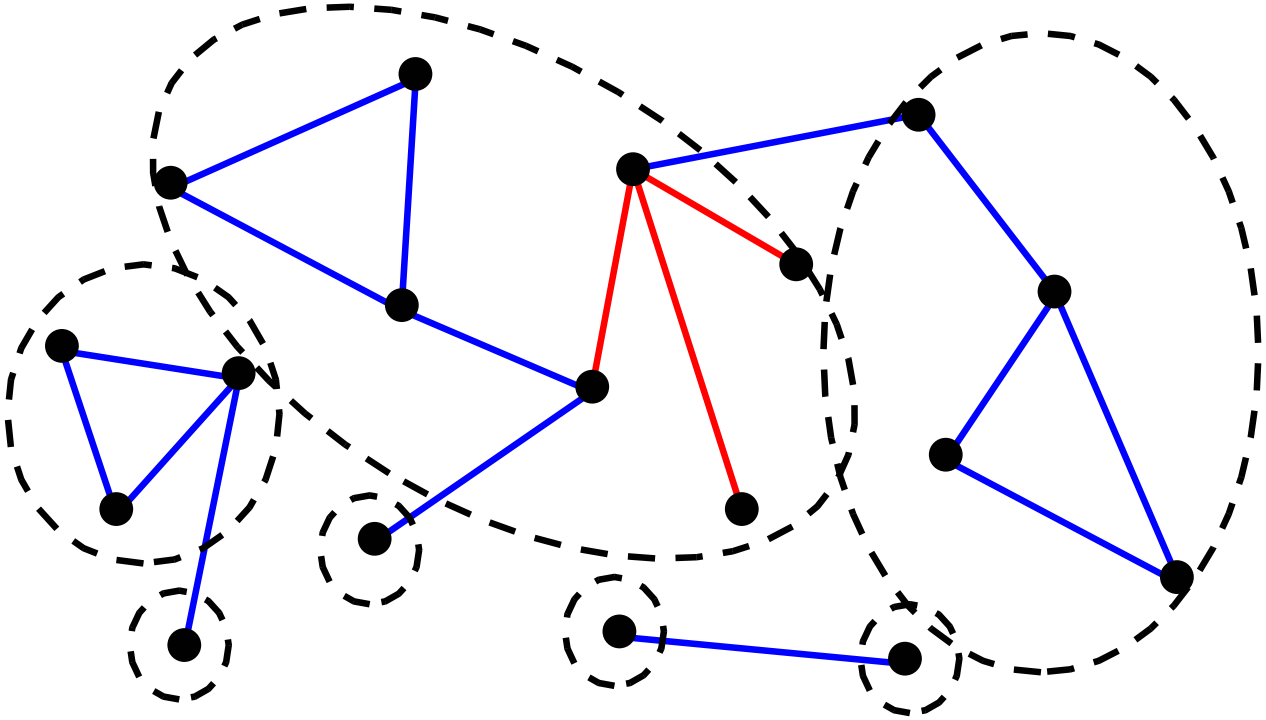
There is a polynomial time algorithm to construct a partition into maximal internally persistent components.

That is only half the story.

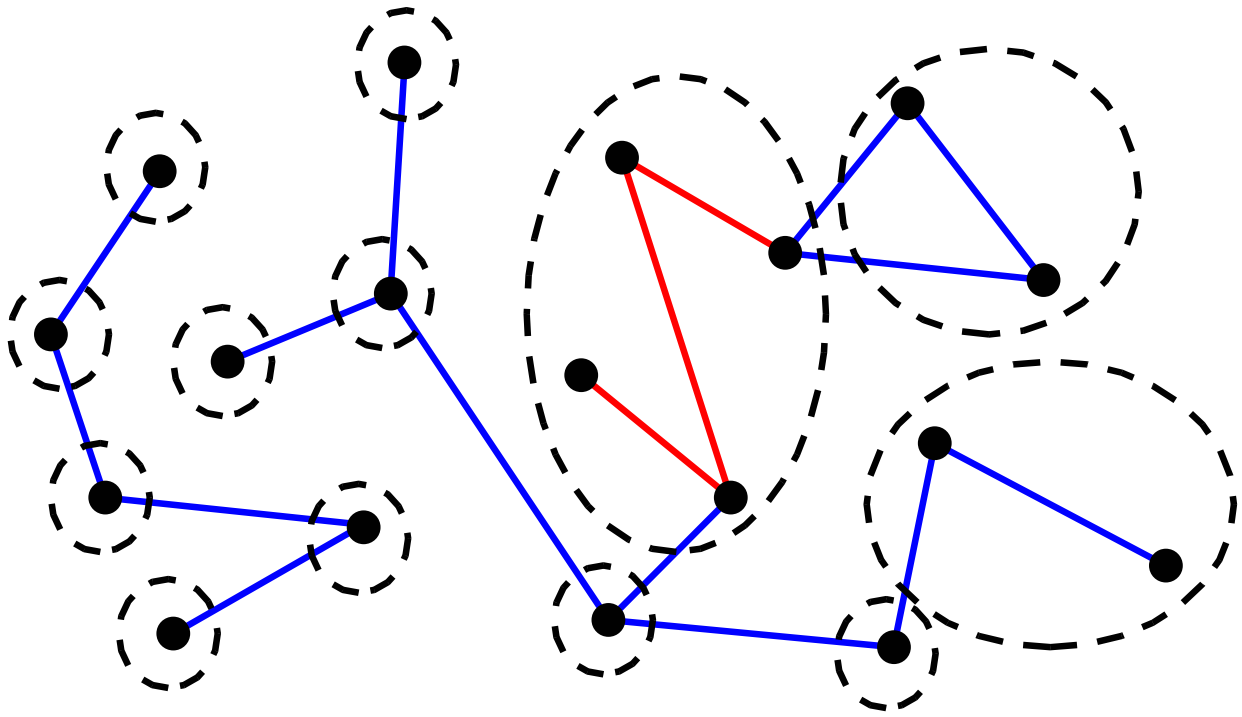
Example



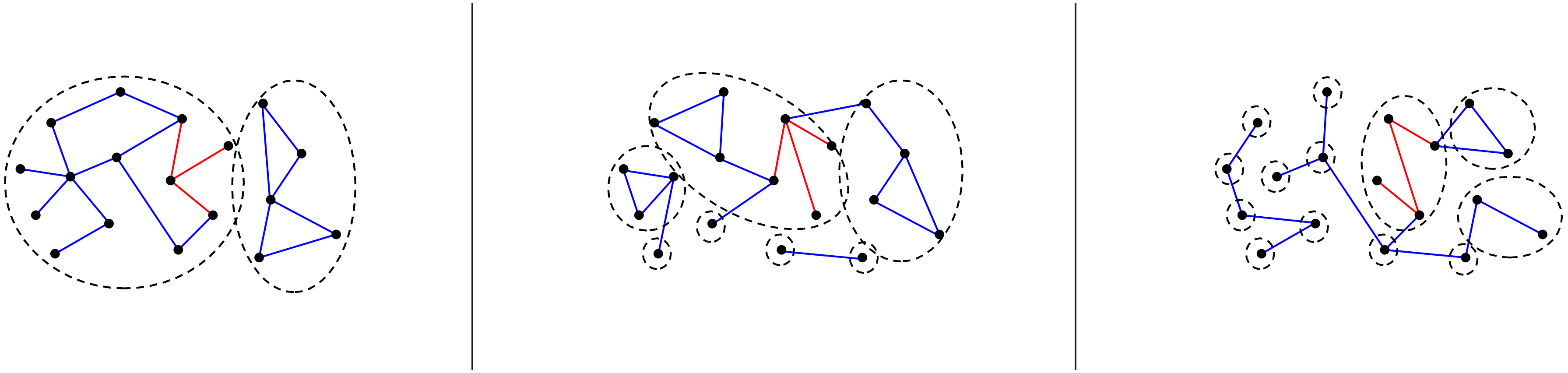
Example



Example



Example



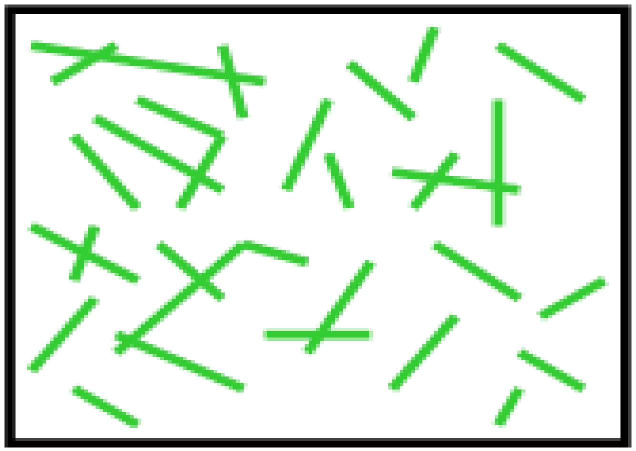
What about background groups that appear persistent?

How do we know that a discovered persistent component is a hidden group and not due to the background communications?

Random Background

If the background (non-planning related) communications are random, then the fortuitous persistence of background groups will be short lived.

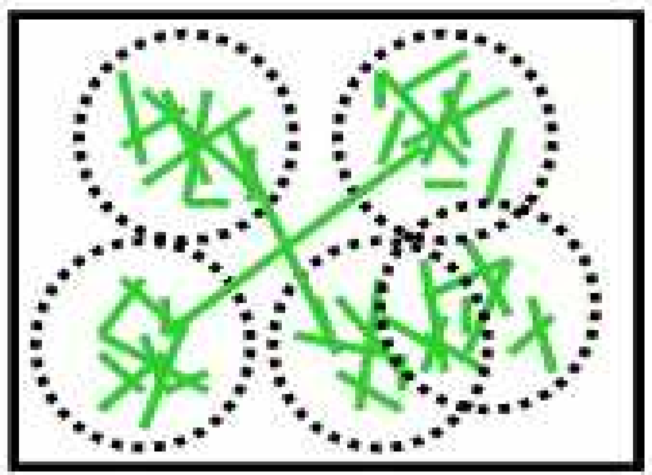
Two Background Models



Uniform Model $G(n, p)$:

n: number of actors.

p: probability of an edge between any two actors.



Group Model $Gr(n, n_g, m, p_g, p_e)$:

n_g: number of groups.

m: group size.

p_g: each group is a $G(m, p_g)$.

p_e: probability of an edge between two actors not sharing a group.

$\boxed{\text{Background IP Components}}$

How likely are chance background IP components?

For example: if the graph is connected w.p.1 due to background communications, then V will be **IP** w.p.1.

Connectivity of the background random graph model will play a role.

The Double Jump

Phase transitions in the connectivity of a $G(n,p)$ random graph:

$p = \frac{c}{n}$	$p = \frac{\ln n}{n} + \frac{x}{n},\ x > 0$
$L(G(n,p)) = \begin{cases} O(\ln n) & 0 < c < 1 \\ O(n^{2/3}) & c = 1 \\ \beta(c)n & c > 1, \beta(c) < 1 \end{cases}$	$P[L(G(n,p)) = n] \geq e^{-e^{-x}}$

$L(G(n,p))$ is the size of the largest component of a $G(n,p)$ graph.

Detecting the Hidden Group

Uniform Model

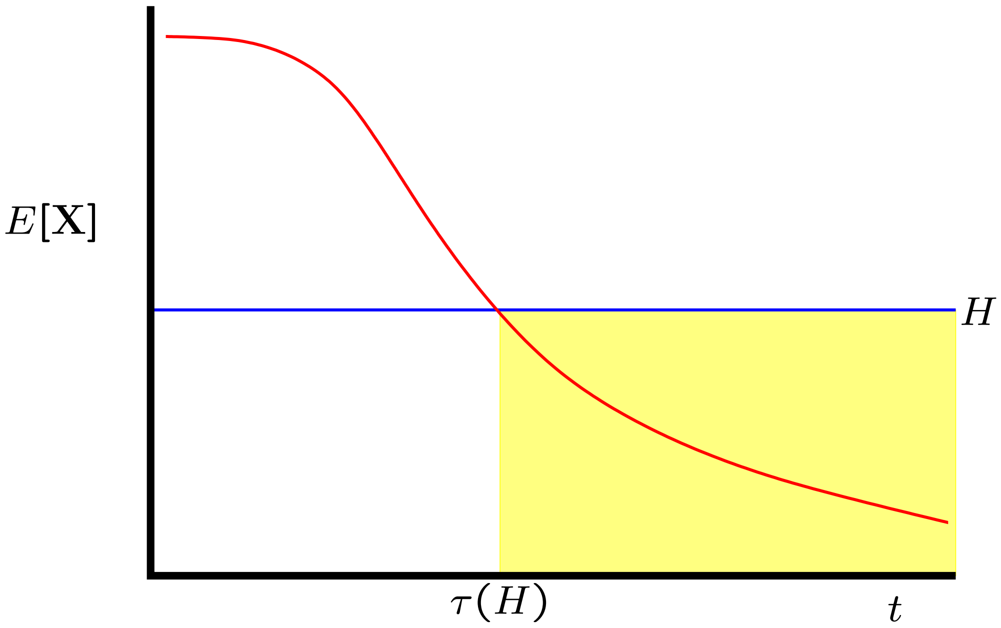
p	Detecting the Hidden Group
$\frac{c}{n}$	Should be easy
$\frac{c \ln n}{n}$	?
c	Don't quit your day job

What about the Group random model?

Random Persistent Components

$\mathbf{X}(\tau)$: size of largest persistent component in $G_1, \ldots, G_\tau$

Consider $E[\mathbf{X}(\tau)]$ – Expected size of the largest persistent component.



$\tau(H)$ – the time to detect a hidden group of size $\geq H$ reliably.

$$\boxed{\tau(H)}$$

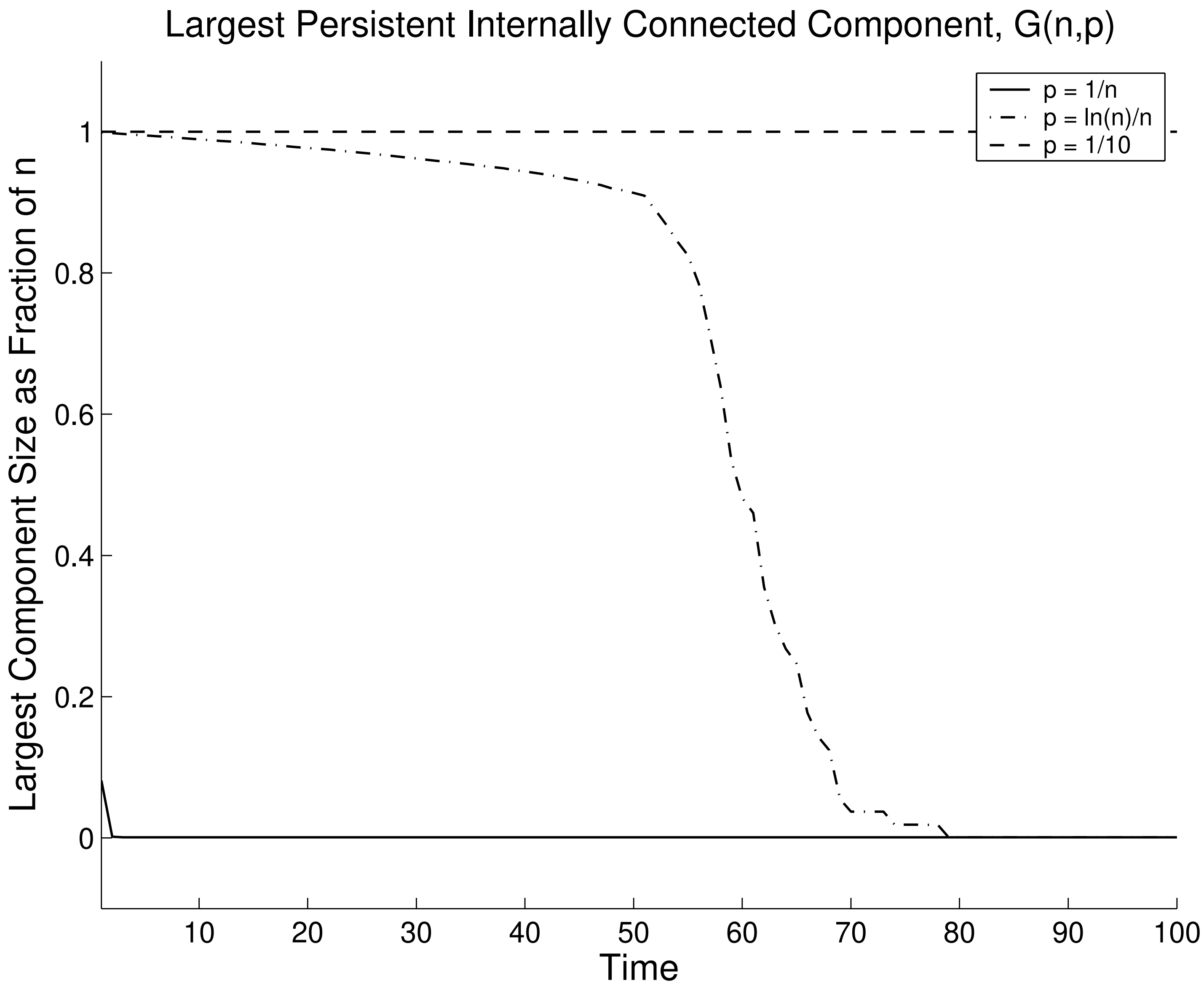
Challenging to compute even for simple random background models.

Simulation

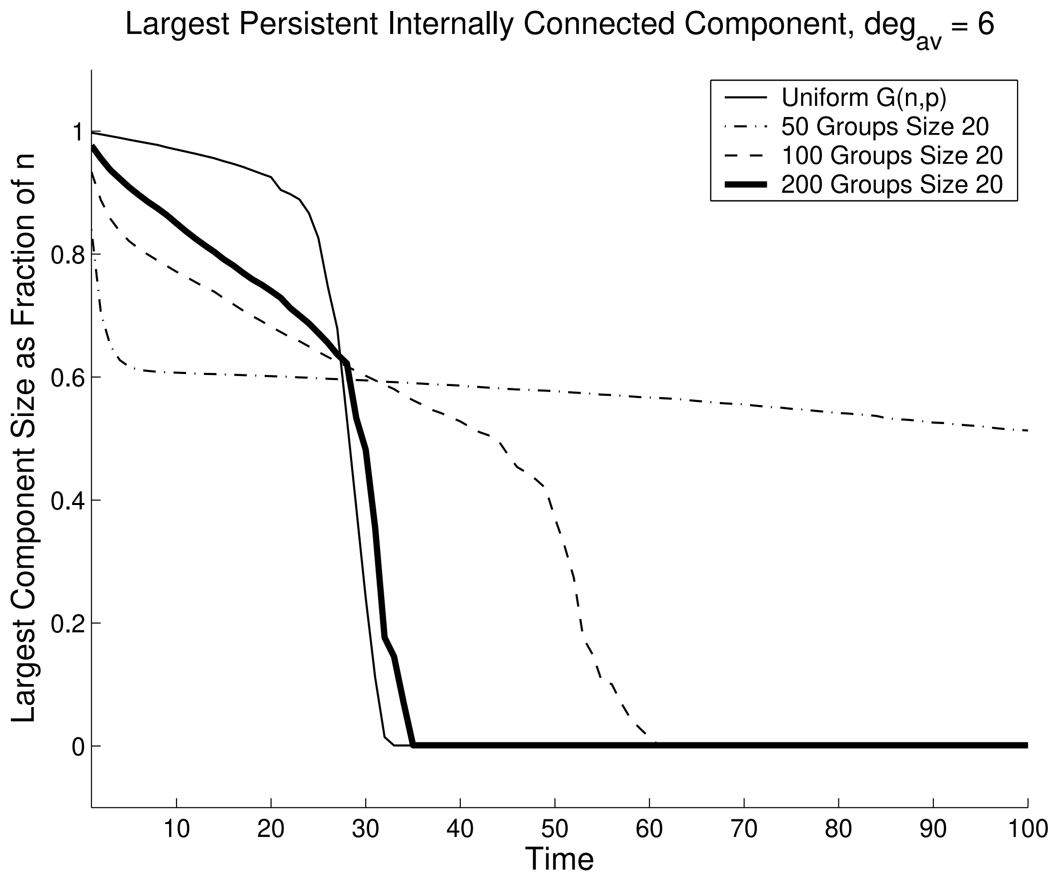
Random Models: $\tau(H)$ can be computed **offline**.

Real Data: $\tau(H)$ can be computed from the communication history.

Experiments: Uniform Model

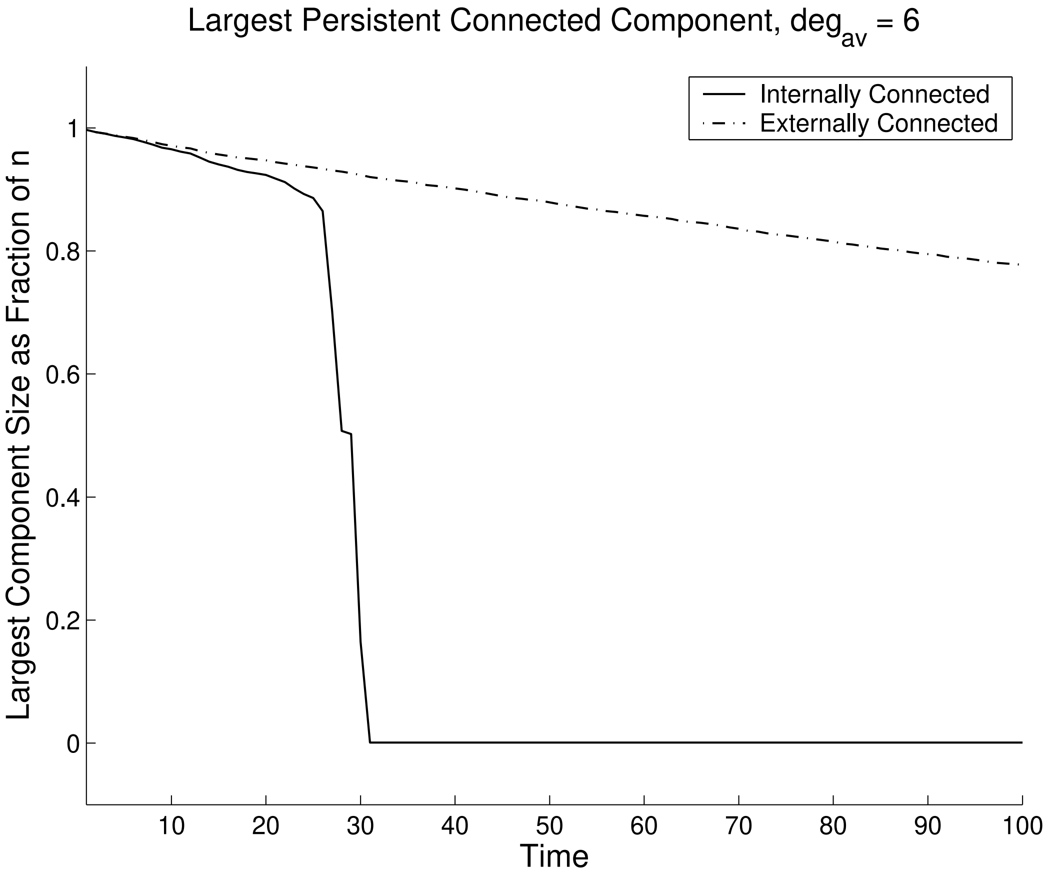


Uniform Model vs. Group Model



	Group Model ($deg_{av} = 6$)			Random Model
n_g (# of groups)	50	100	200	$G(n, \frac{6}{n})$
$\tau(1)$	> 100	63	36	32

IP vs. EP



deg_{av}	$\tau(1)$ for **EP**	$\tau(1)$ for **IP**
2	28	2
6	> 100	32

Summary

Easier to detect the hidden group (statistically)

– Less dense background communication.

– Less structured background communication.

– **Paranoid/Secretive hidden group**.

$\boxed{\text{General Hidden Group Model}}$

– Hidden group is not active during every communication cycle. The hidden group is active for some time period, $[t_1, t_2]$.

– The hidden group does not communicate during every cycle in the time period of its activity.

– The entire hidden group need not participate in the planning at every cycle in which it communicates.

General Problem is Hard

Problem: Frequently_Mostly_Connected (FMC)

Input: A sequence of graphs $\{G_i(V, E_i)\}_{i=1}^{T}$ on the same vertex set V; positive integers $s \leq |V|$ and $k \leq T$, and the fraction ϵ, $0 < \epsilon \leq 1$;

Question: Is there a subset $S \subseteq V$ with $|S| \geq s$ which induces an ϵ-partially connected subgraph in at least k of the $\{G_i\}$?

Theorem. Problem FMC is NP-complete.

Proof. Reduction from Balanced Complete Bipartite Subgraph [Garey & Johnson, (problem GT24)]

Conclusions and Ongoing Work

– Two facets: **algorithmic**, **statistical**.
  – Streaming data.
  – More realistic random models; real data.

– If you get to choose your battles, detect **paranoid** hidden groups in **unstructured**, **sparce** backgrounds.

– Taking **communication intensities** into account.

– Approaching the general problem:
  – More structure on the hidden group
  – Heuristics

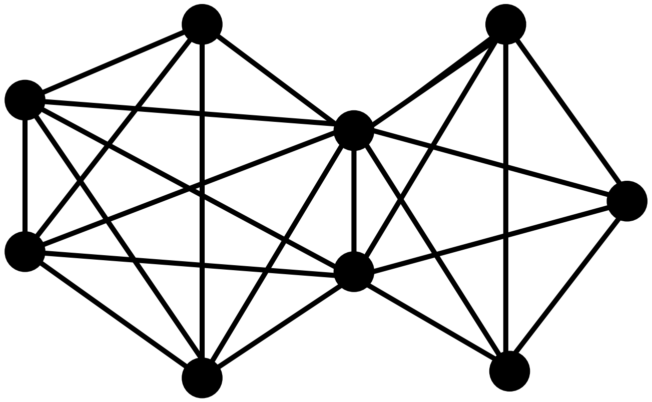
– What about non-maximal **IP** components.

– **Evolving hidden groups**.

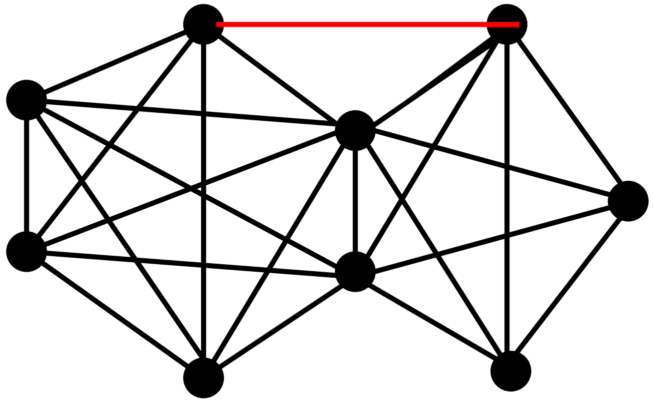
PART II

Overlapping Communities

Example



G



H

Computing Clusters:

– How many edges can we add before two clusters become one?

Defining Clusters

Social Network Context:

- Communities are clusters.
- Communities are characterized by more intense communications
- A cluster is a locally maximal subgraph:

  adding a new (removing an old) vertex decreases the density

How to define density:

- Depends on applications
- May depend on the number of inside/outside edges the set

Possible Density Definitions.

$$\frac{e^{in}(S)}{|S|};$$

$$\frac{e^{in}(S)}{e^{in}(S)+\alpha e^{out}(S)};$$

$$\frac{2e^{in}(S)}{|S|(|S|-1)};$$

$$\frac{e^{in}(S)}{e^{in}(S)+\alpha\frac{|S|-1}{|\bar{S}|}e^{out}(S)}$$

$$|S|/radius(G_S);$$

$$|S|/diameter(G_S)$$

Iterative Scan

$C \leftarrow seed;\ \ w \leftarrow W(C);$

`increased` $\leftarrow$ TRUE;

while `increased` do

  for all $v \in V(G)$ do

    if $v \in C$ then

      $C' \leftarrow (C - v)$

    else

      $C' \leftarrow C \cup \{v\};$

    if $W(C') > W(C)$ then

      $C \leftarrow C';$

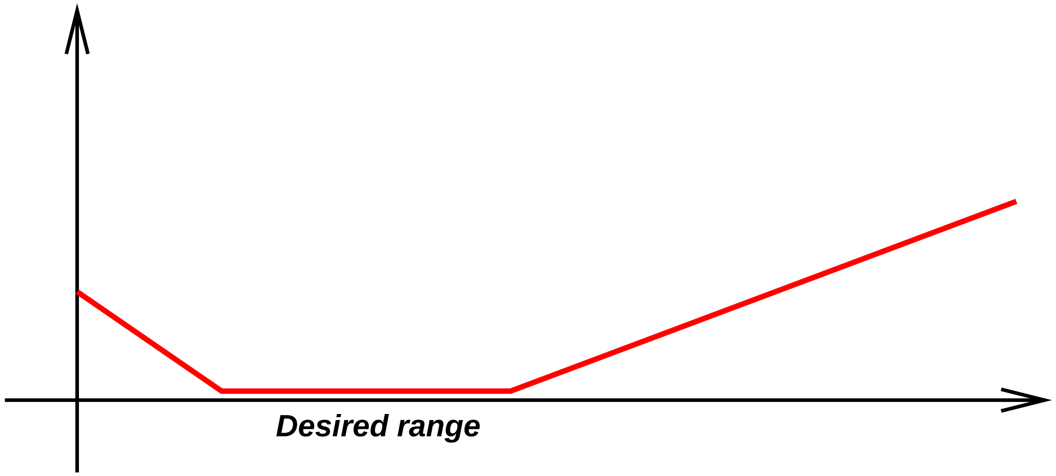
  if $W(C) = w$ then

    `increased` $\leftarrow$ FALSE;

  else

    $w \leftarrow W(C);$

return C



Rank Removal (*RaRe*)

Procedure $RaRe(G, W)$;

Global $R \leftarrow \emptyset$;

ComputeRanks($V(G)$); /* $\phi_p(v) = c \sum_{u,v} \frac{\phi_p(u)}{deg^-(v)} + \frac{(1-c)}{n}$ */

$\{H_i\}$ connected components of G;

for all H_i,

    if $|V(H_i)| \leq max$

        H_i is a core_cluster

    else

        ClusterComponent(H_i);

Clusters $\{C_i\} \leftarrow$ core_clusters;

for all $v \in R$ do

    for all clusters C_i do

        if $W(v \cup C_i) > W(C_i)$

            add v to C_i;

Rank Removal (*continue*)

Procedure $ClusterComponent(H)$;
if $|V(H_i)| \leq max$
    $T \leftarrow \{t$ highest rank vertices in $H\}$;
    $R \leftarrow R \cup T$' $H \leftarrow H - T$;
    $\{F_j\}$ are connected components of H;
    for all F_j do
        ClusterComponent(F_j);
else
    if $min \leq |V(H)|$ do
        mark H as a core_cluster;

Run Time of IS and RaRe