Signal Processing for Graphs

Benjamin A. Miller
Nadya T. Bliss, Patrick J. Wolfe*, Michelle S. Beard, Nicholas Arcolano, Matthew Schmidt and Karl Ni

MIT Lincoln Laboratory
***Harvard University**

09 August 2011

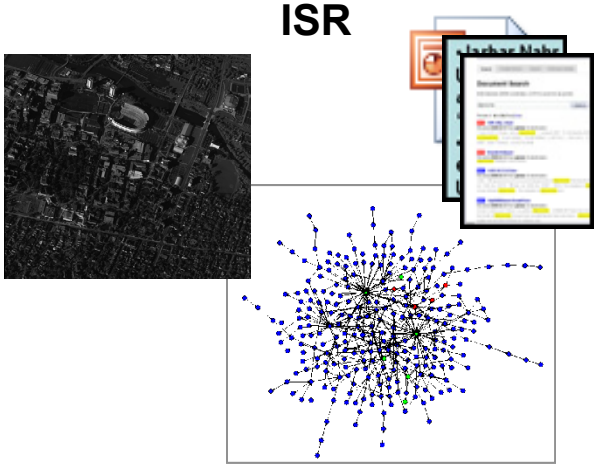
Graph Exploitation Symposium

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

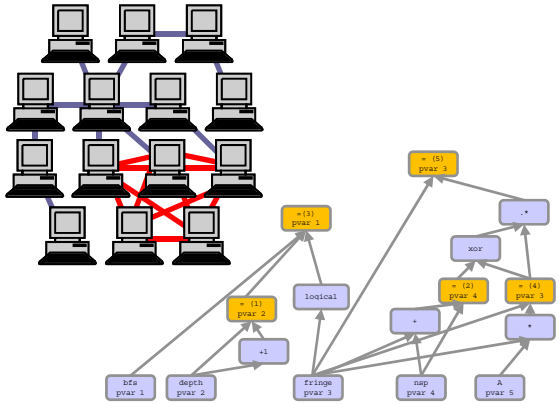
This work is sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Applications of Graph Analysis
From Data to Relationships

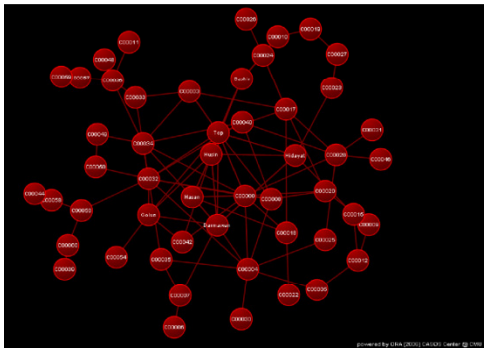
ISR



CYBER SECURITY



SOCIAL NETWORK ANALYSIS



- Extract entities detected from multi-INT sources
- Analyze relationships between entities

- Analyze communication patterns
- Analyze software execution patters
- Find behaviors consistent with cyber attack

- Consider interactions between individuals
- Locate anomalies and threat activity

Data encoding relationships between entities (graphs or networks) are highly relevant to a wide range of defense and intelligence applications

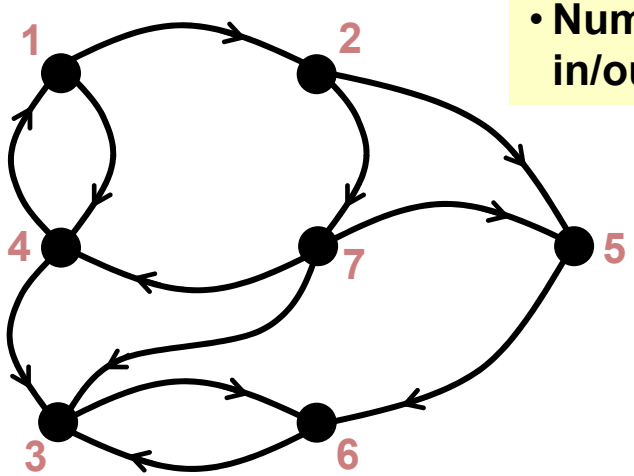
Graph Preliminaries (Formal Definition)

Graph G = (V,E) where
- **V = set of vertices**
- **E = set of edges**

Adjacency matrix representation:
- **Entry A(i,j) is non-zero if there exists an edge between vertices i and j**

Degree
- **Number of edges in/out of vertex v**

Graph G



Adjacency Matrix A

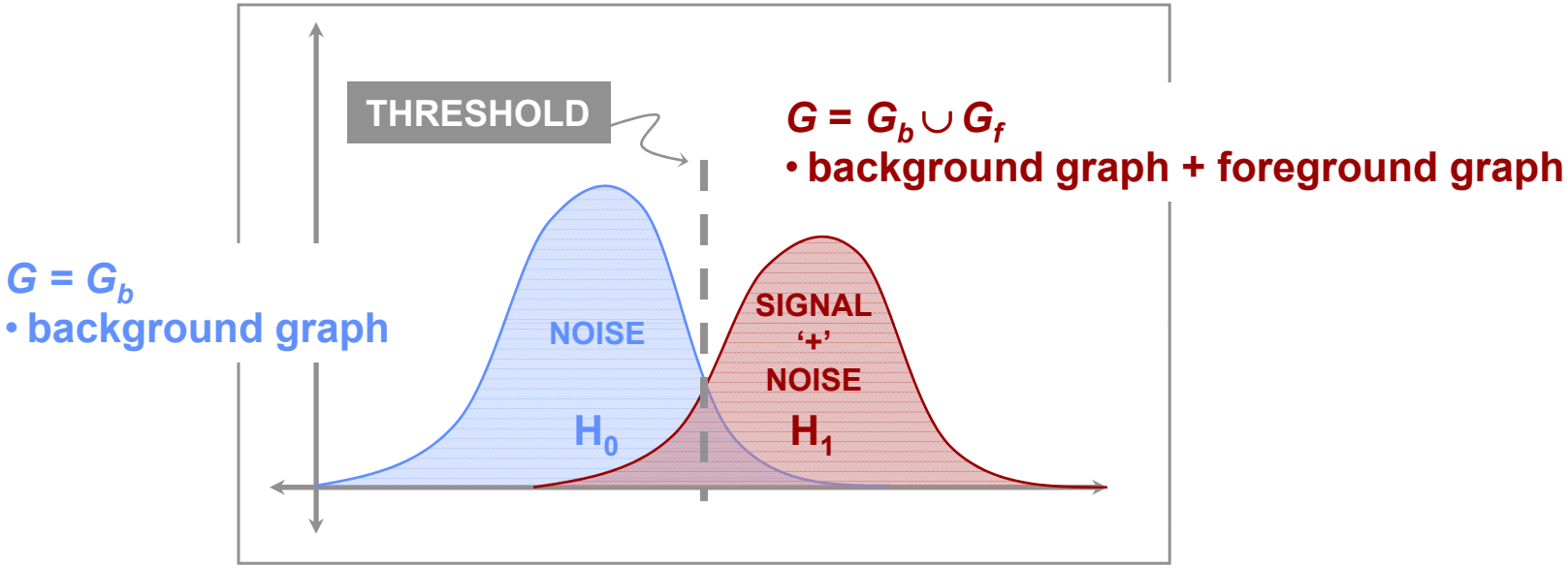
	1	2	3	4	5	6	7	Degree
1		●		●				2
2					●		●	2
3						●		1
4	●		●					2
5						●		1
6			●					1
7			●	●	●			3

GRAPH CHARACTERISTIC	ADJACENCY MATRIX REPRESENTION
G is undirected	A is symmetric
G is directed	A may or may not be symmetric
G is unweighted	A(i,j) is either 0 or 1
G is weighted	A(i,j) is a weight
G is a multi-graph	A is a multidimensional array

Time domain or multi-INT representation

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Subgraph Detection Problem



- H_0 : $G = G_b$
- H_1 : $G = G_b \cup G_f$
- Detection problem:
  - Given G, is H_0 or H_1 true?

Graph Detection Challenges:

- Background models
- Foreground models
- High-dimensional space

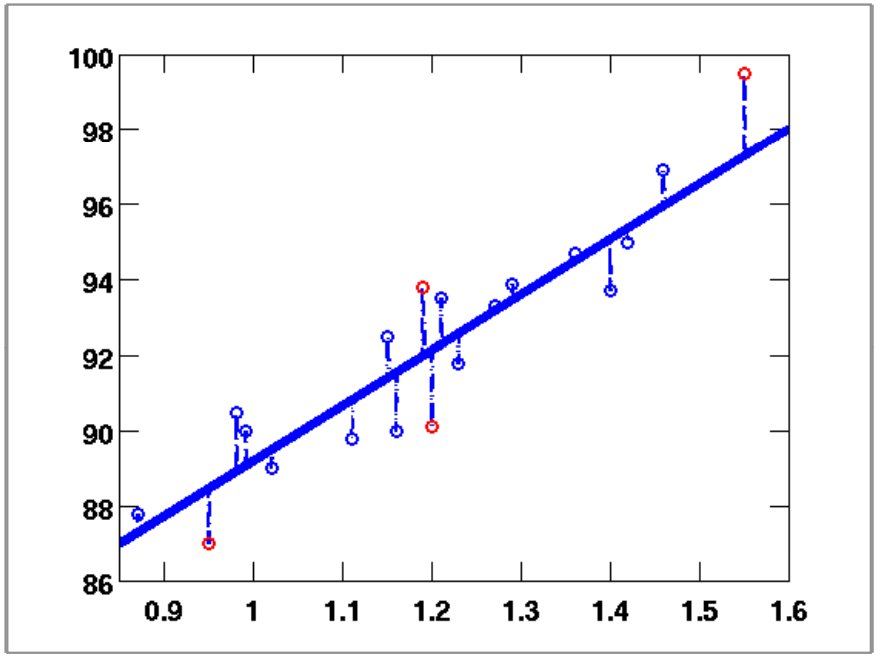
1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Outline

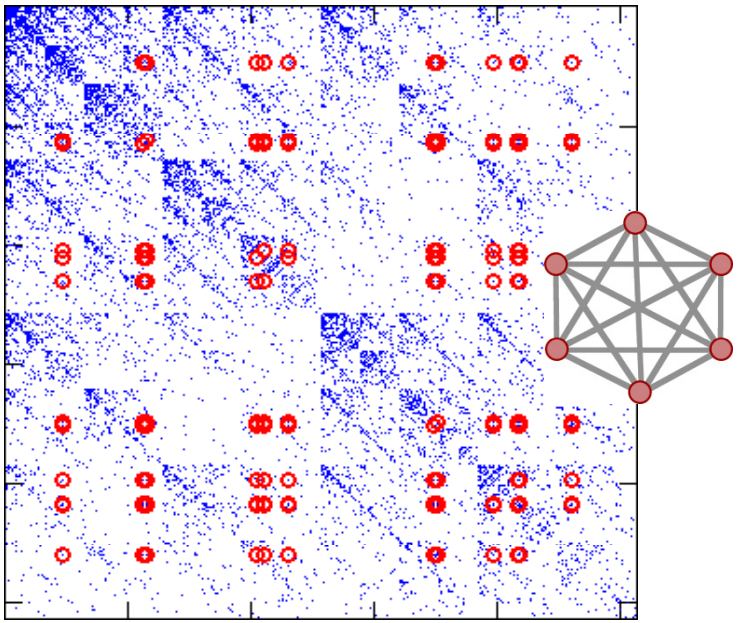
- Introduction
- **Approach**
- Results
- Summary

Graph Based Residuals Analysis

Linear Regression



Graph "Regression"



• Least-squares residuals from a best-fit line	• "Residuals" from a best-fit graph model
• Analysis of variance (ANOVA) describes fit	• Analysis of variance from expected topology
• "Explained" vs "unexplained" variance → signal/noise discrimination	• Unexplained variance in graph residuals → subgraph detection

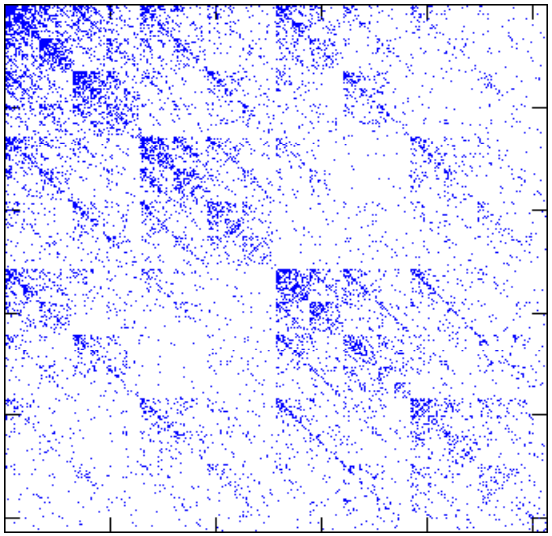
ANALYSIS OF MODULARITY

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Subgraph Detection Algorithm Overview

Processing chain for subgraph detection analogous to a traditional signal processing chain

MODULARITY MATRIX CONSTRUCTION → EIGEN DECOMPOSITION → COMPONENT SELECTION → DETECTION → IDENTIFICATION

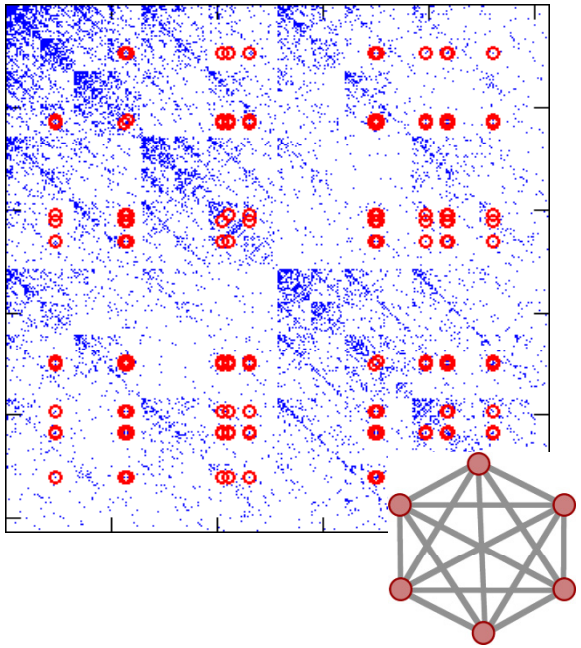


Input:

- A, adjacency matrix representation of G
- No cue

Output:

- v_s, set of vertices identified as belonging to subgraph G_f



1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Modularity Matrix* Construction

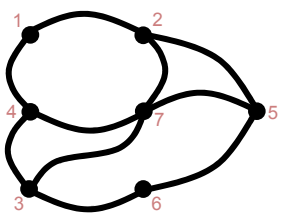
MODULARITY MATRIX CONSTRUCTION → EIGEN DECOMPOSITION → COMPONENT SELECTION → DETECTION → IDENTIFICATION

$$B = A - \frac{KK^T}{2M}$$

ADJACENCY MATRIX

DEGREE VECTOR

EXAMPLE: GRAPH G



$$\begin{array}{c|ccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ \hline 1 & & \bullet & & \bullet & & & \\ 2 & \bullet & & & & \bullet & & \bullet \\ 3 & & & & \bullet & & \bullet & \bullet \\ 4 & \bullet & & \bullet & & & & \bullet \\ 5 & & \bullet & & & & \bullet & \bullet \\ 6 & & & \bullet & & \bullet & & \\ 7 & & \bullet & \bullet & \bullet & \bullet & & \end{array}$$

$$- \frac{1}{20} \left\{ \begin{bmatrix} 2 \\ 3 \\ 3 \\ 3 \\ 3 \\ 2 \\ 4 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 & 3 & 3 & 2 & 4 \end{bmatrix} \right\}$$

NUMBER OF EDGES

- Commonly used to evaluate *quality* of division of a graph into communities
- Observed minus "expected" edges, yielding residuals

*M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E, 74:036104, 2006.

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Eigen Decomposition

MODULARITY MATRIX CONSTRUCTION → **EIGEN DECOMPOSITION** → COMPONENT SELECTION → DETECTION → IDENTIFICATION

$$B = UDU^T$$

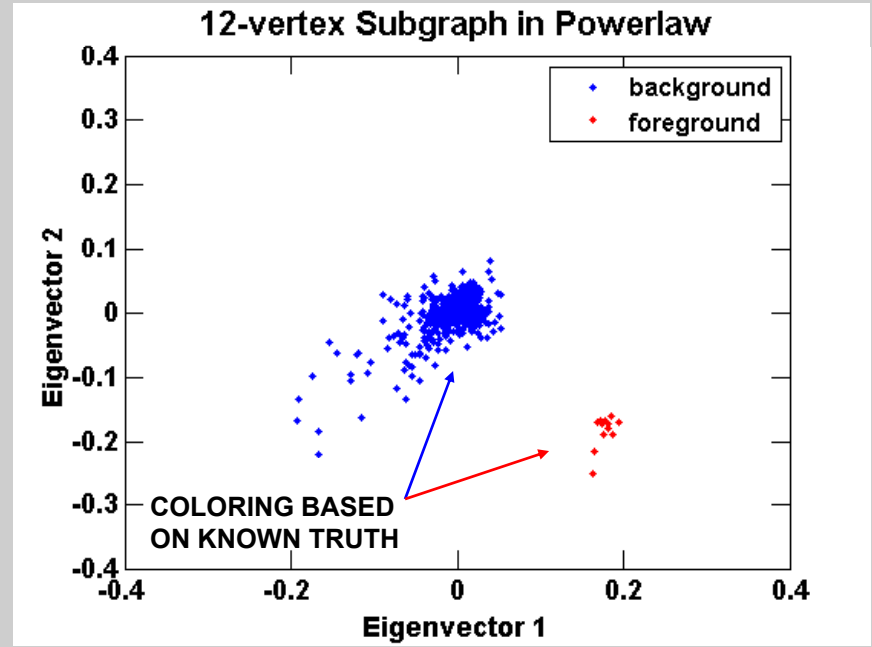
Eigenvalues, sorted by magnitude

$$\begin{bmatrix} \lambda_1 & & & & \\ & \lambda_2 & & & \\ & & \cdots & & \\ & & & \lambda_{N-1} & \\ & & & & \lambda_N \end{bmatrix}$$

Corresponding eigenvectors

$$[\mathbf{u}_1 \mid \mathbf{u}_2 \mid \ldots \mid \mathbf{u}_{N-1} \mid \mathbf{u}_N]$$

Principal components



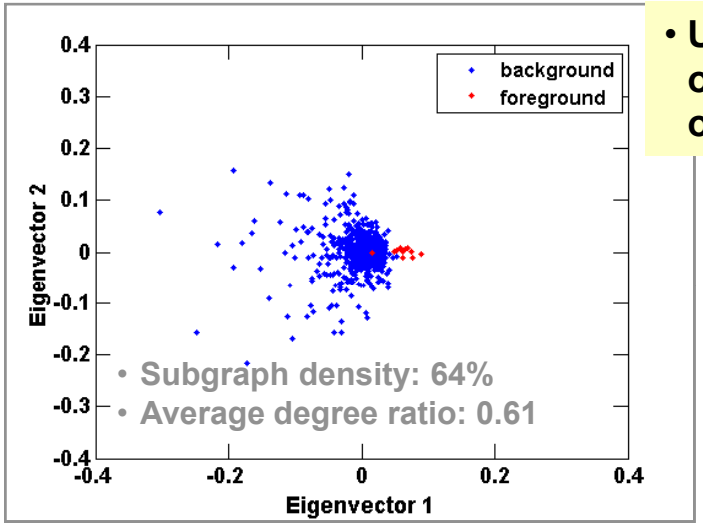
- Each point represents a vertex in *G*
- Vertices in *G*: 1024
- Vertices in G_f: 12
- Uncued background/foreground separation

Projection onto principal components of the modularity matrix yields good separation between background and foreground

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Component Selection
Detection of Weak Signals

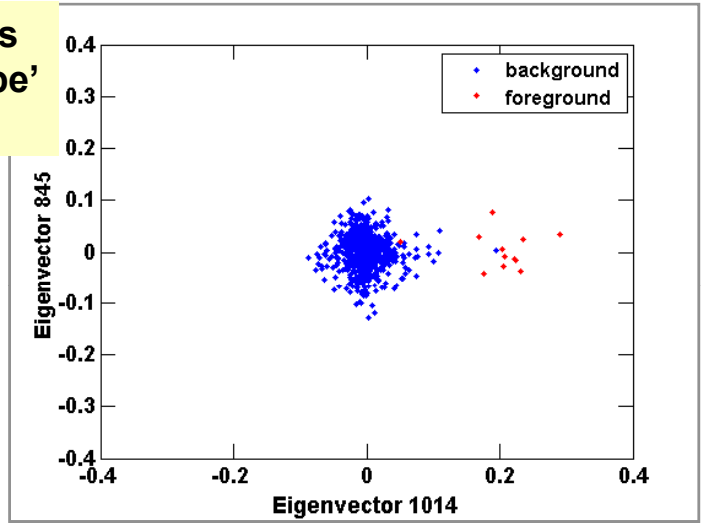
MODULARITY MATRIX CONSTRUCTION → EIGEN DECOMPOSITION → **COMPONENT SELECTION** → DETECTION → IDENTIFICATION



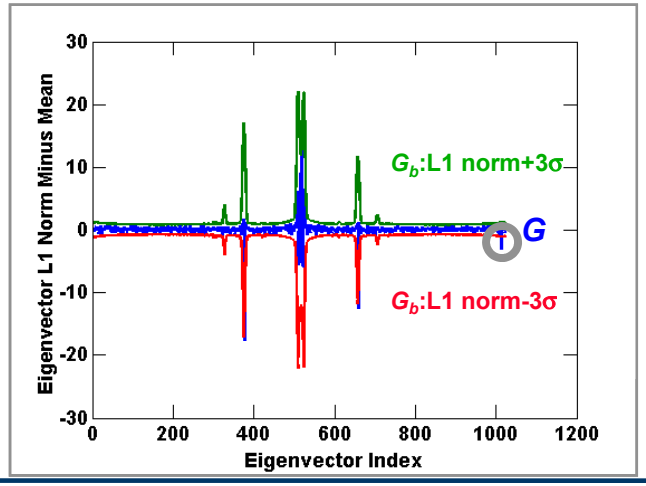
- Use L_1 norms of eigenvectors of G_b to generate an 'envelope' of normal graphs

L_1 Norm of vector u

$$\| \mathbf{u} \|_1 = \sum | u |$$



- Principal component analysis unable to separate weak foreground graph signature



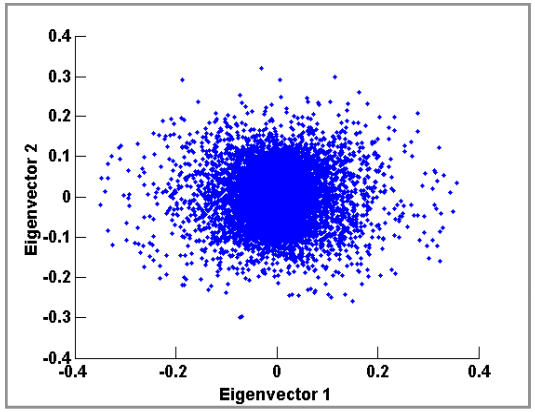
- Select eigenvectors outside the envelope for projection
- Selected components yield good separation

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Toy Detection Example

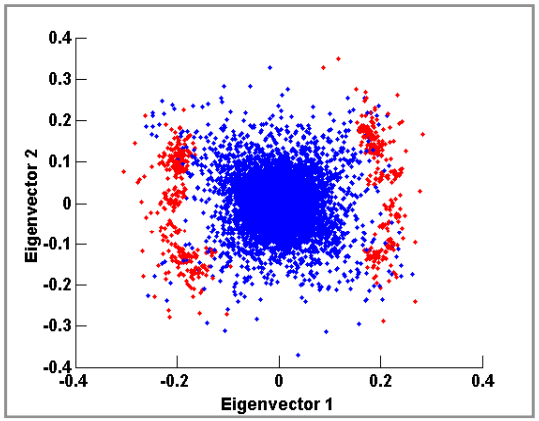
MODULARITY MATRIX CONSTRUCTION → EIGEN DECOMPOSITION → COMPONENT SELECTION → **DETECTION** → IDENTIFICATION

H_0



Background (normal behavior), G_b only

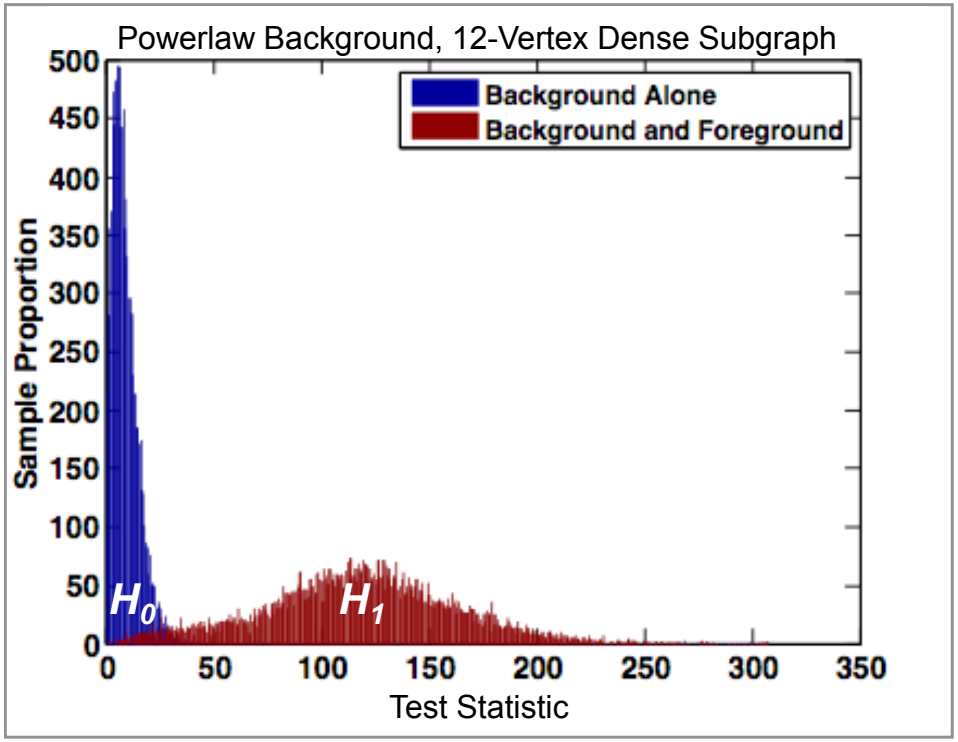
H_1



Background and foreground (includes anomalous behavior), $G_b + G_f$

TEST STATISTIC:

SYMMETRY OF THE PROJECTION ONTO SELECTED COMPONENTS



H_0 and H_1 distributions are well separated

B.A. Miller, N.T. Bliss and P.J. Wolfe, "Toward signal processing theory for graphs and non-Euclidean data," in *Proc. ICASSP*, 2010.

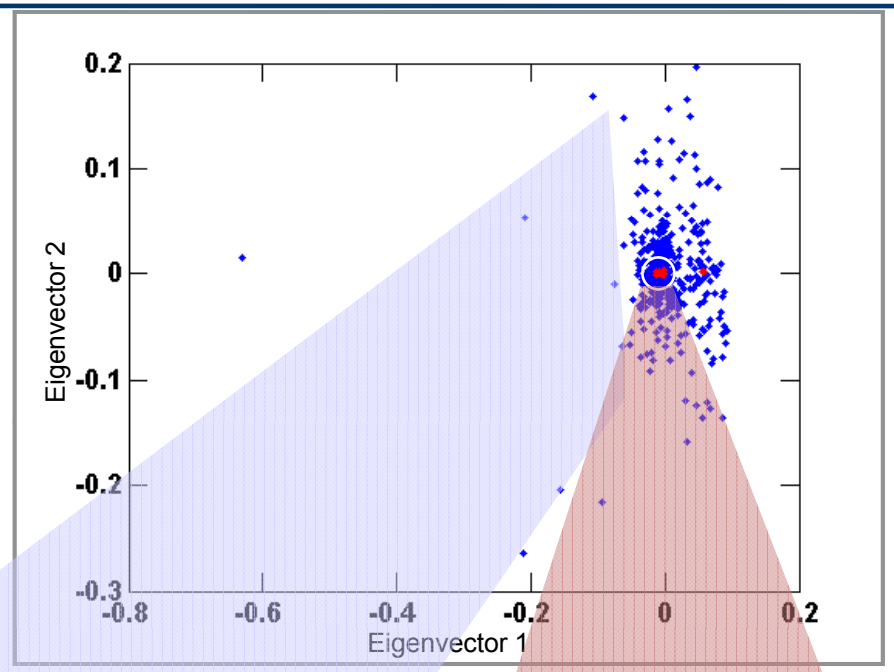
1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Outline

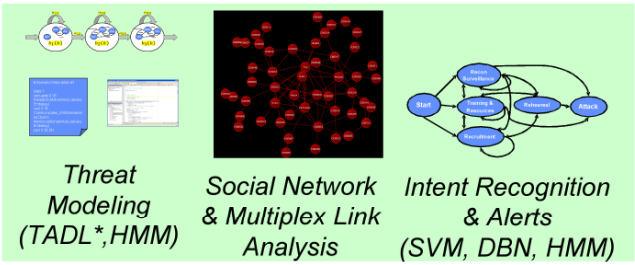
- Introduction
- Approach
- **Results**
  - **Analysis of Static Graphs**
  - Analysis of Dynamic Graphs
- Summary

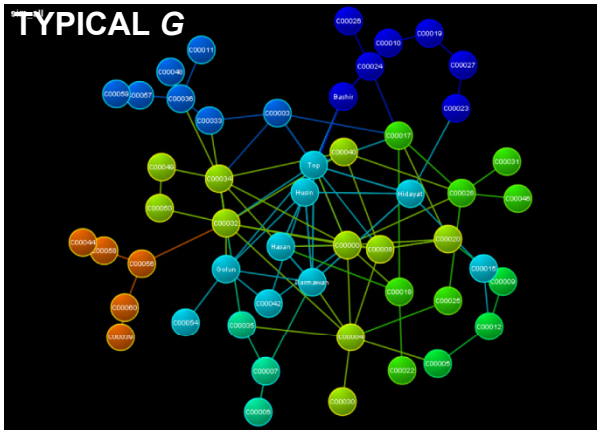
Social Network Analysis

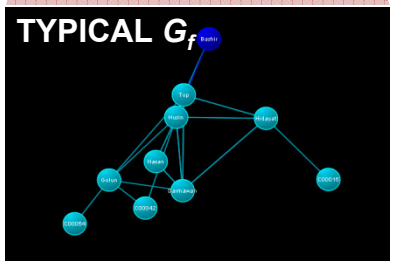
- Threat network dataset generated by the CT-SNAIR program
- Low density threat network embedded in a power law background
- Weak signature subgraph of interest
  - Average degree ratio: 0.44
  - Max eigenvalue ratio: 0.36
- Principal component analysis yields no foreground separation



Counter Terror Social Network Analysis and Intent Recognition





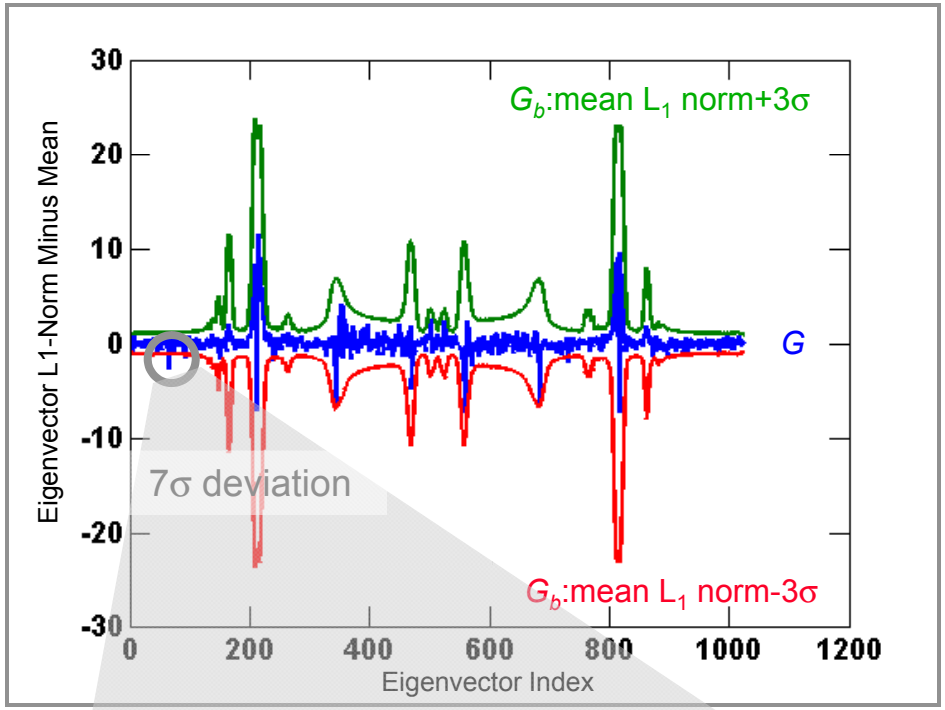


Networks generated from real scenarios

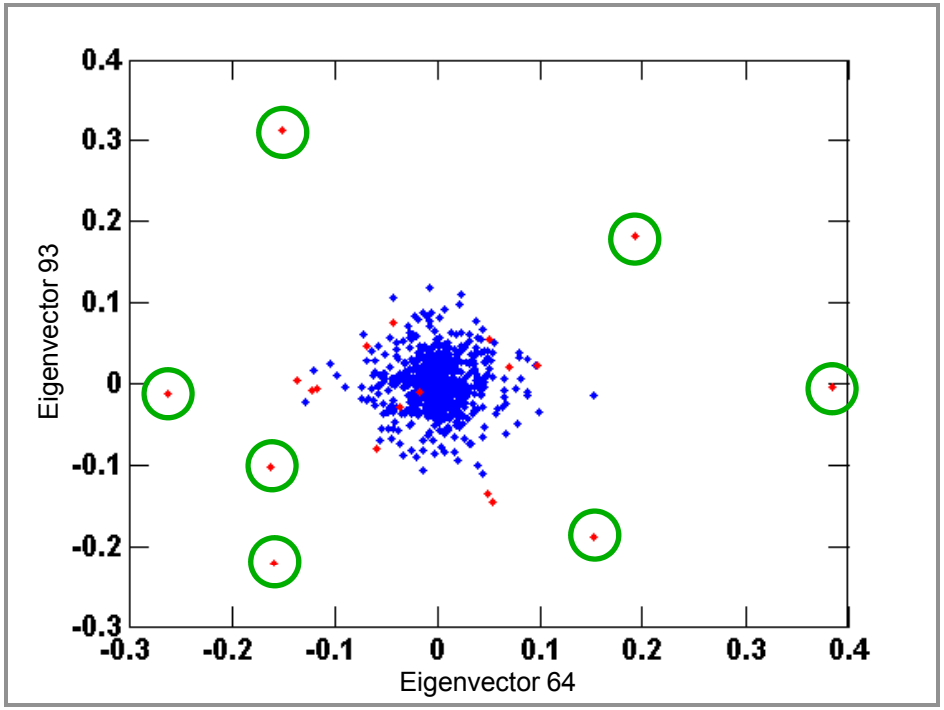
1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Threat Network Identification

L_1 Norm of Eigenvectors of Residuals Matrix



Scatter Plot Using Selected Components

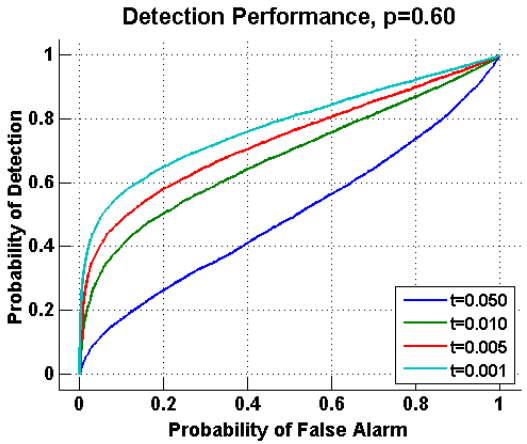


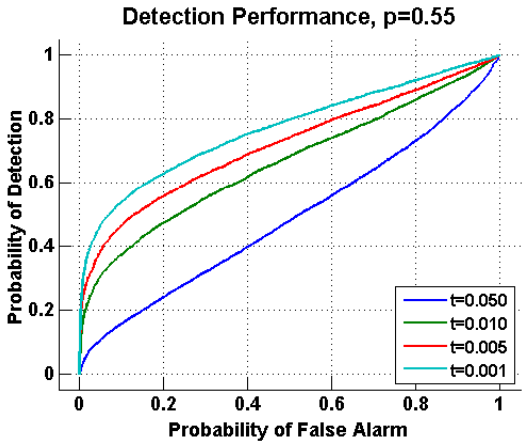
Background statistics (power law) allow detection of statistically significant deviation in the L_1 norm

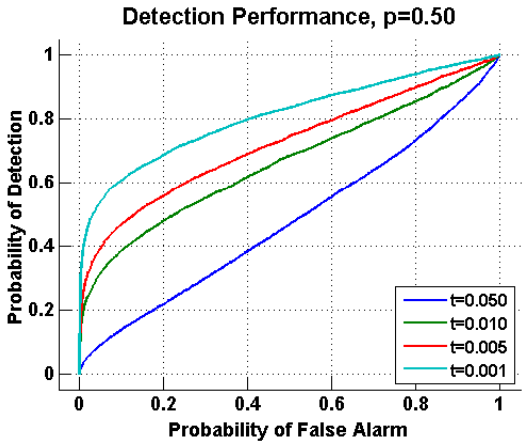
- **Significant number of threat vertices are separable and are identified by clustering**
- **Identified individuals can be used as cues**

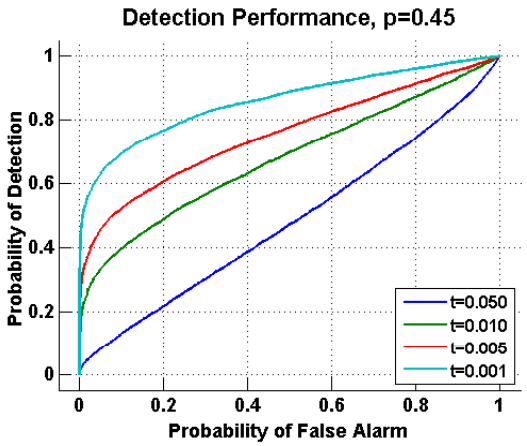
1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

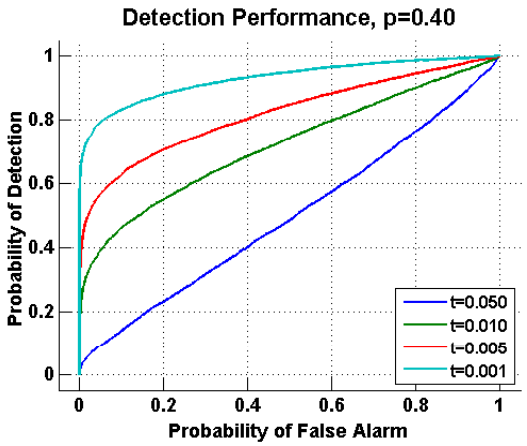
Detection Performance

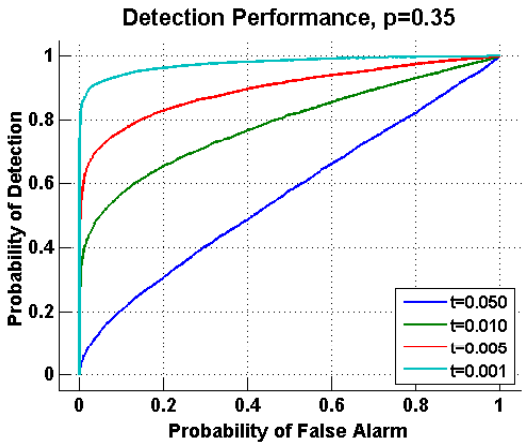












Detection performance improves as background strength decreases (across plots) and correlation with the background decreases (within plots)

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

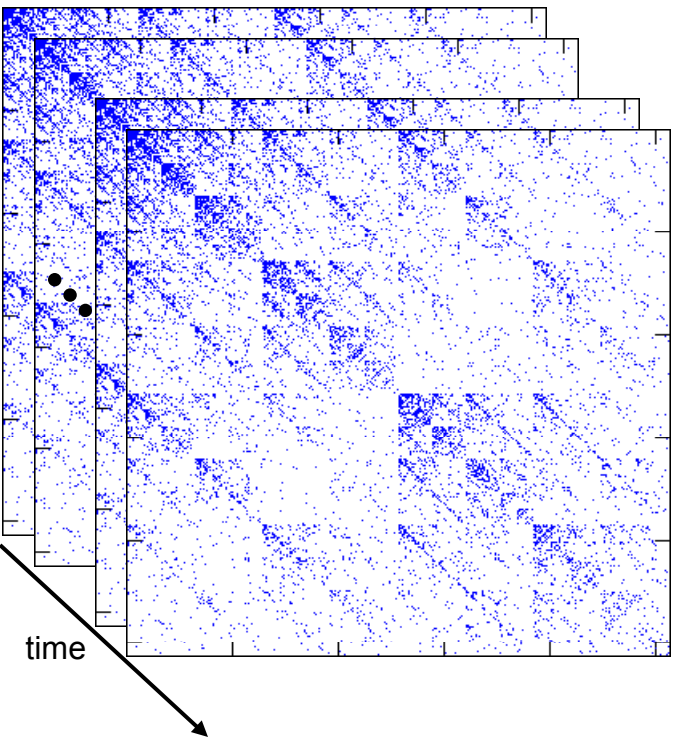
Outline

- Introduction
- Approach
- **Results**
  - Analysis of Static Graphs
  - **Analysis of Dynamic Graphs**
- Summary

Dynamic Graph Processing Chain

Processing chain for *dynamic* subgraph detection incorporates *temporal integration*

MODULARITY MATRIX CONSTRUCTION → TEMPORAL INTEGRATION → EIGEN DECOMPOSITION → COMPONENT SELECTION → DETECTION → IDENTIFICATION

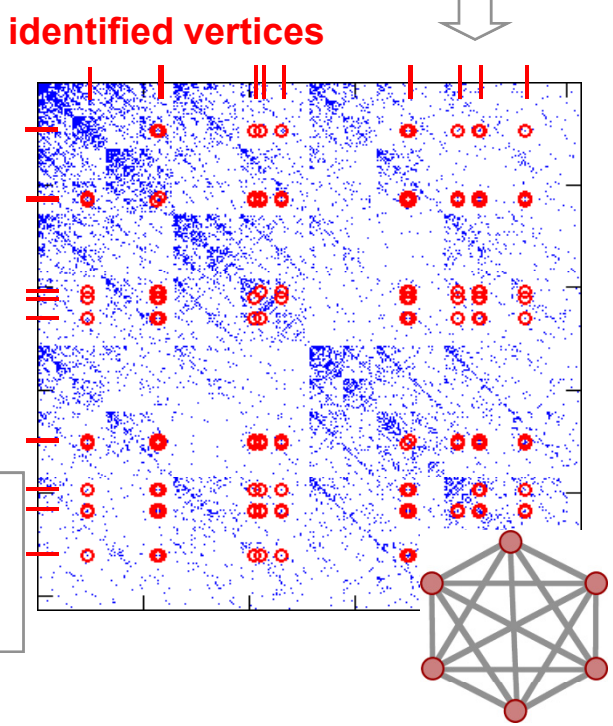


Input:

- ***A(t)*, *time-varying* adjacency matrix representation of *G***
- ***No cue***

Output:

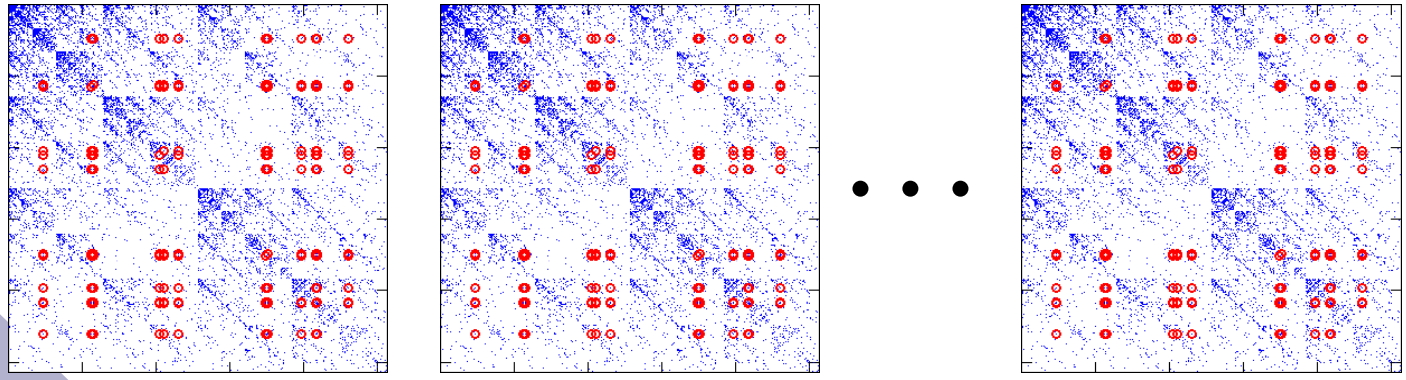
- **v_s, set of vertices identified as belonging to subgraph G_f**



1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Temporal Integration

MODULARITY MATRIX CONSTRUCTION → **TEMPORAL INTEGRATION** → EIGEN DECOMPOSITION → COMPONENT SELECTION → DETECTION → IDENTIFICATION



Modularity (residuals) matrix at time *n*

$$\frac{B(n)}{h(0)} + \frac{B(n-1)}{h(1)} + \cdots + \frac{B(n-\ell+1)}{h(\ell-1)}$$

Filter coefficients (scalars)

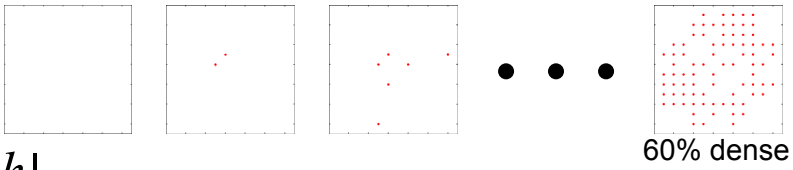
$$\tilde{B}(n) = \sum_{k=0}^{\ell-1} h(k) B(n-k)$$: **Weighted sum of residuals**

Temporal integration achieved by creating a linear combination of individual time-step graphs

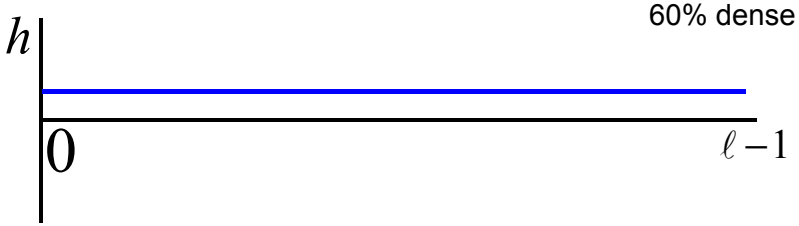
1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

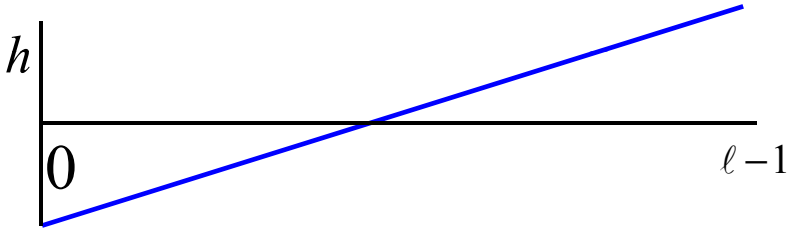
Filter Design

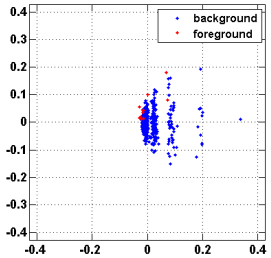
Foreground: subgraph with linearly increasing density

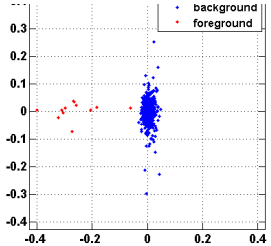


No cue **to foreground vertices; entire graph is filtered**









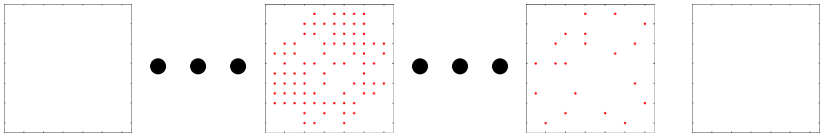
Test Statistic

0.2

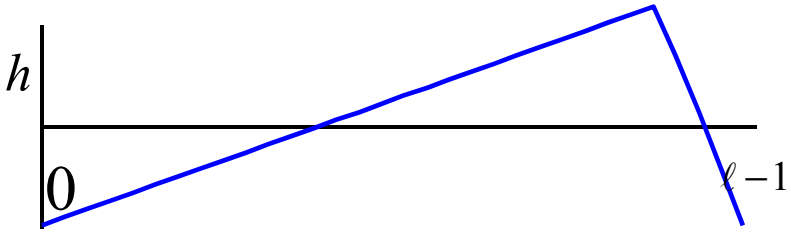
210.6

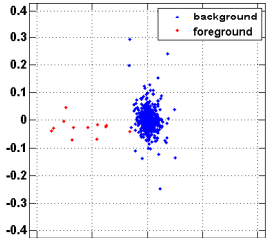
- FG vertex
- BG vertex
- ⬆ test stat. better

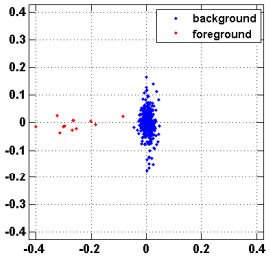
Foreground: subgraph that densifies and disperses



Foreground behavioral information yields better separation







51.2

183.0

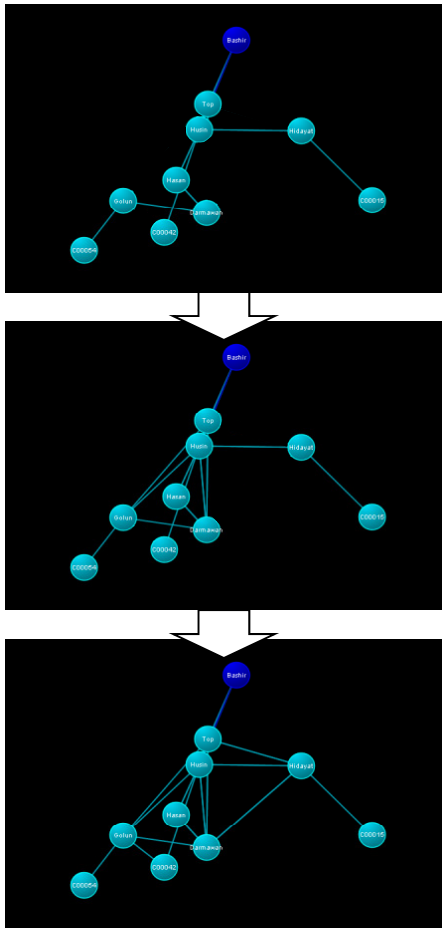
1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Application: Threat Network Evolution Detection

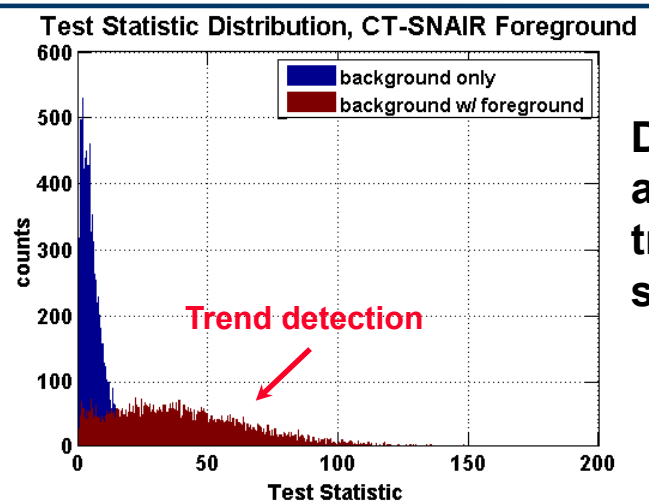
recruitment

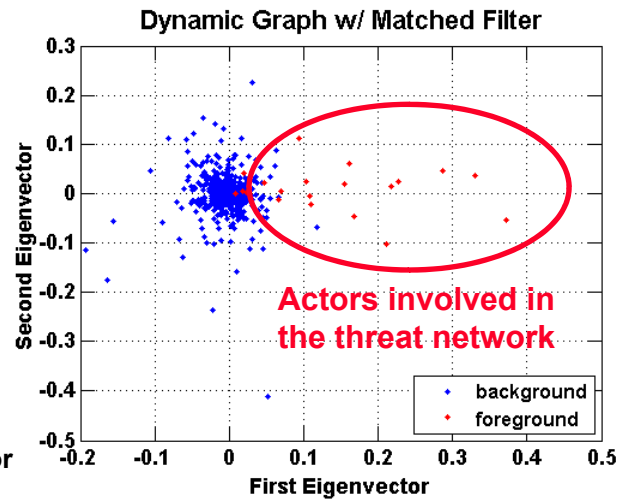
planning

execution



*Threat network foreground created under the Counter Terror Social Network Analysis and Intent Recognition effort





CONOPS:

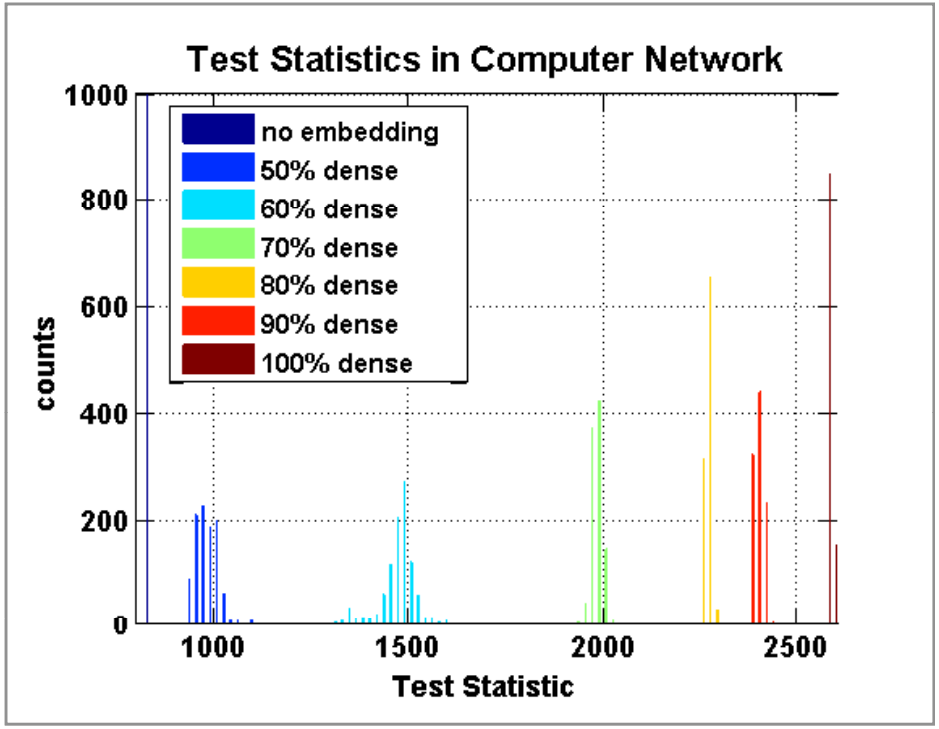
Detect anomalous trends in a social network

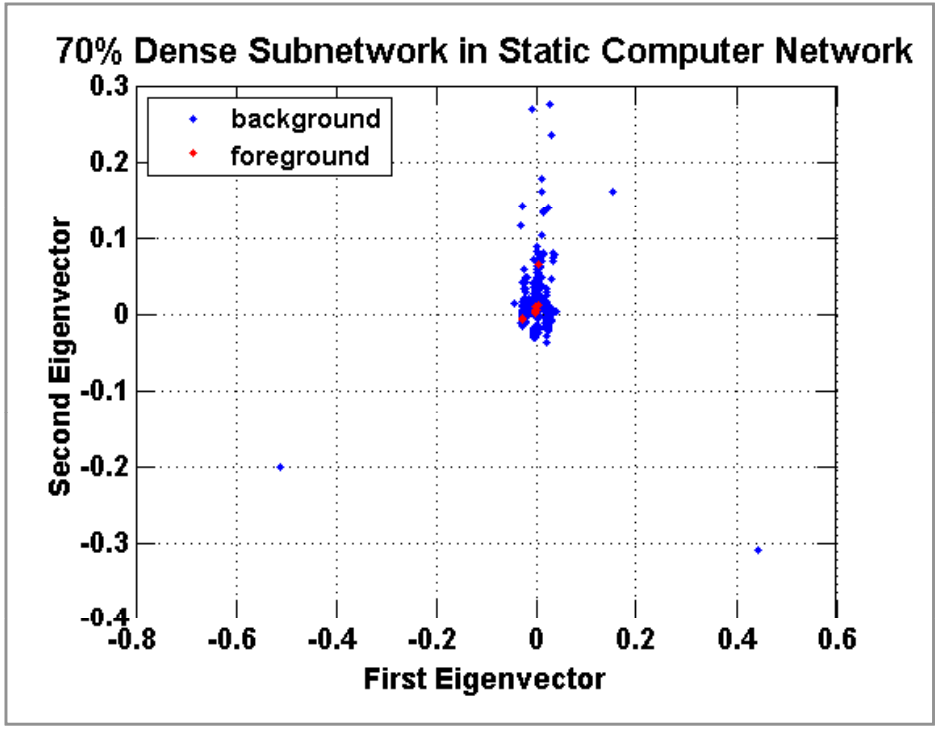
Identify actors (individuals) involved

Time integration allows for identification of weak signature coherent activity

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Application: Cyber Security



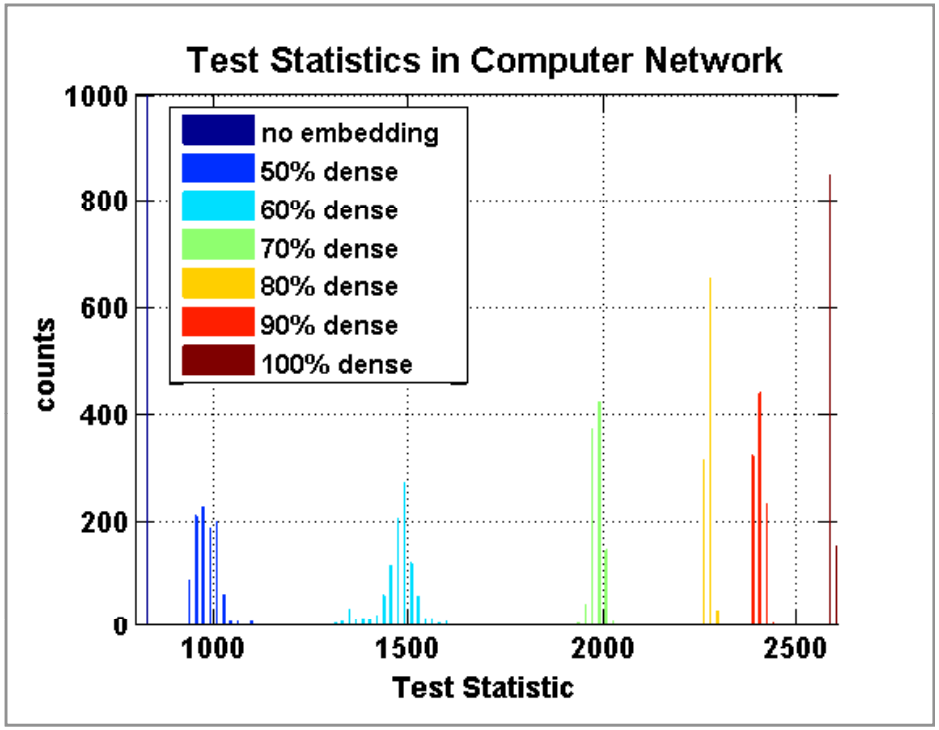


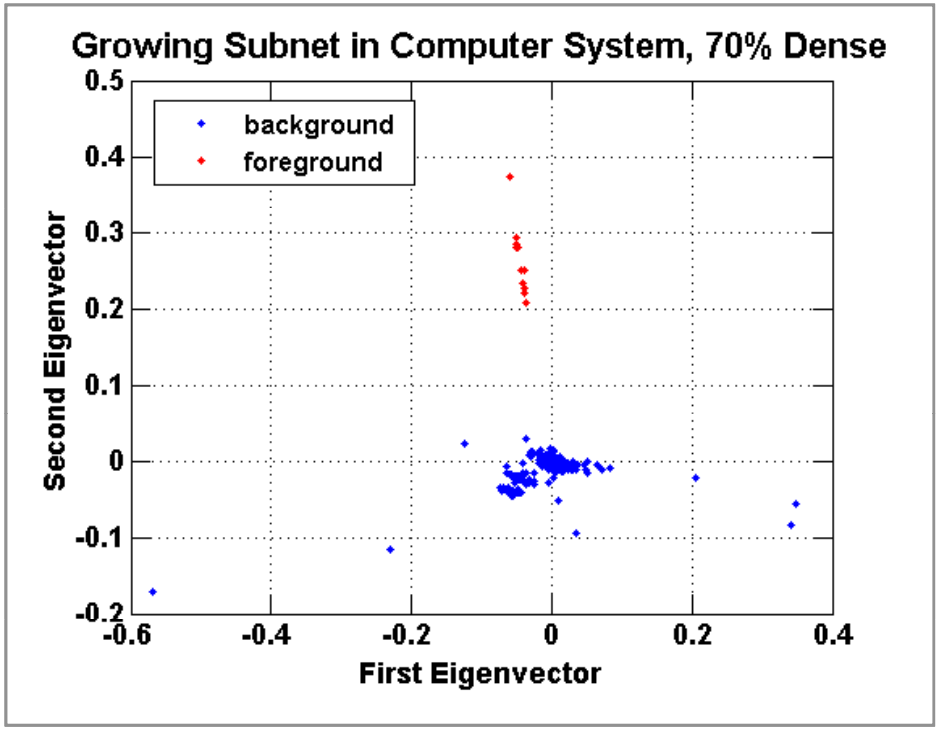
- Anomaly detection in computer networks can help counter cyber attacks
- Background: Real-life, dynamic, publicly-available* computer network (~3200 nodes, powerlaw)
- Foreground: Increasingly dense subgraph; simulating a subnetwork with exceptionally high inter-node communication

*Available on SNAP: http://snap.stanford.edu

1951–2011
LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Application: Cyber Security

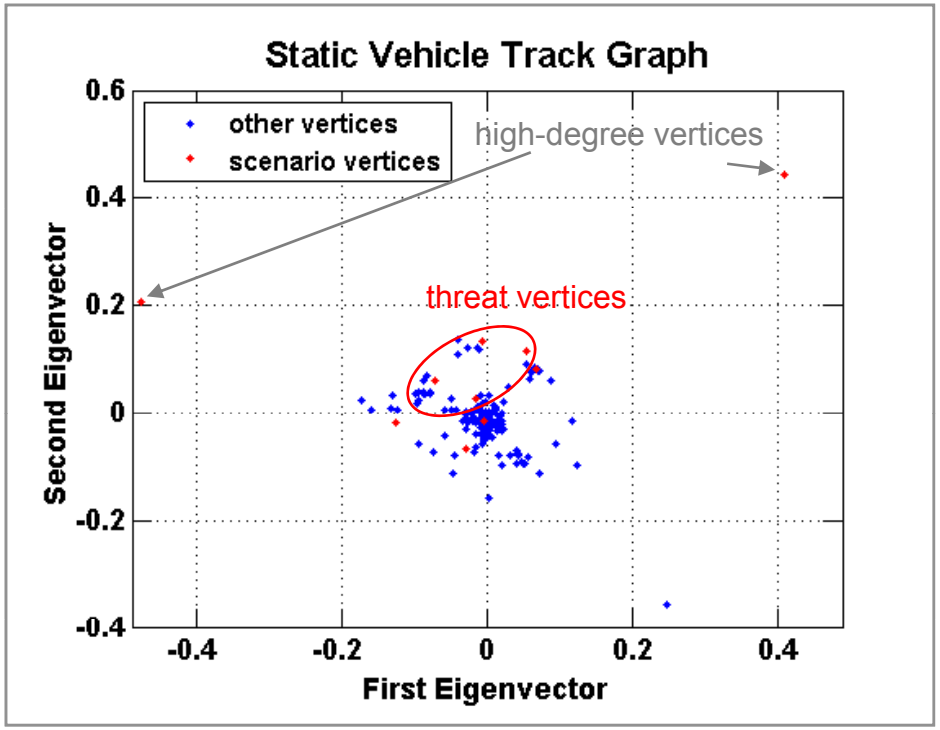




- Anomaly detection in computer networks can help counter cyber attacks
- Background: Real-life, dynamic, publicly-available* computer network (~3200 nodes, powerlaw)
- Foreground: Increasingly dense subgraph; simulating a subnetwork with exceptionally high inter-node communication

*Available on SNAP: http://snap.stanford.edu

Application: Vehicle Track Analysis

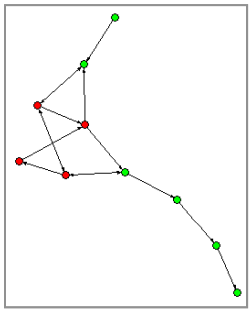


- Vehicle track graph generated using tracks from an EO dataset
- Vertices are sites, edges represent tracks
- Temporal data (i.e., when the tracks begin and end) is also recorded
- Scripted threat scenario among tracks

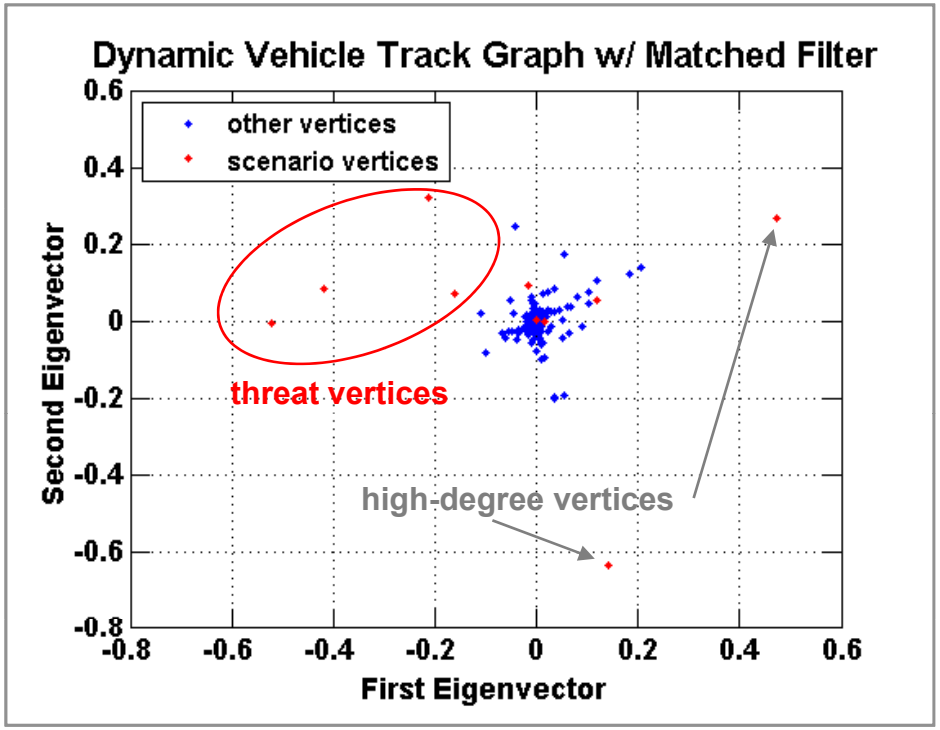
Scripted scenario:

threat vertices

other vertices



Application: Vehicle Track Analysis

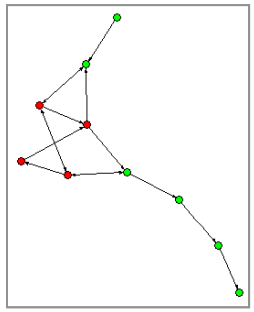


- Vehicle track graph generated using tracks from an EO dataset
- Vertices are sites, edges represent tracks
- Temporal data (i.e., when the tracks begin and end) is also recorded
- Scripted threat scenario among tracks

Scripted scenario:

threat vertices

other vertices



Summary

- Graphs and networks are key data structures for representation and analysis of relationships in complex multi-INT datasets
- Subgraph detection is an important problem for IC and DoD
- Detection framework for graphs enables algorithms and metrics development
- Results on simulated and real datasets demonstrate the power of the approach
- Background statistics (noise and clutter model) and time domain integration are of key importance in detecting weak signatures