Spectral Methods for Subgraph Detection

Nadya T. Bliss & Benjamin A. Miller
Embedded and High Performance Computing
MIT Lincoln Laboratory

Patrick J. Wolfe
Statistics and Information Laboratory
Harvard University

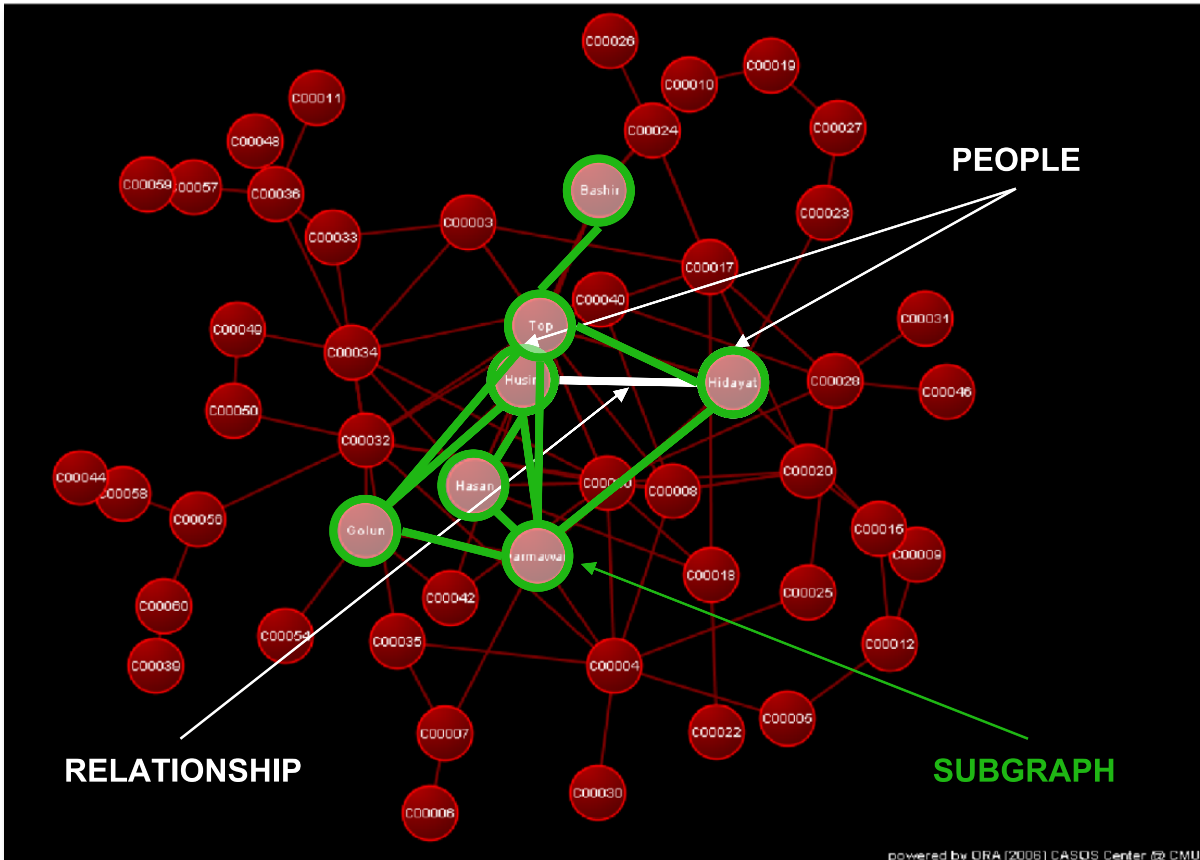
12 July 2010

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 author and are not necessarily endorsed by the United States Government.

Outline

- **Introduction**
- Approach
- Handling Large Graphs
- Summary

Application Examples



Wide variety of application domains

Social network analysis
- **Relationships between people**

Biology
- **Interactions between proteins**

Signal/image processing
- **Discrimination and classification**

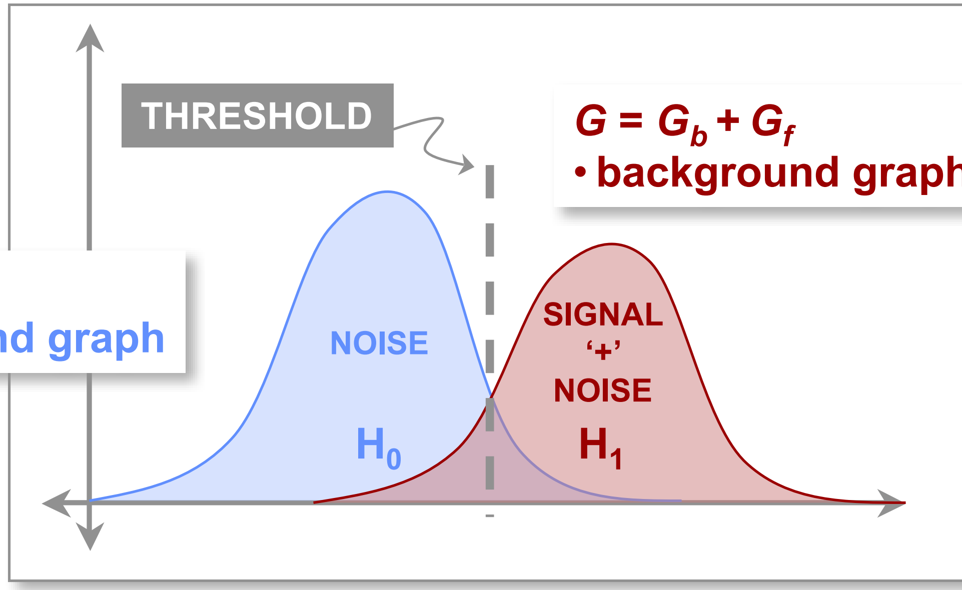
Computer Networks
- **Failure detection**

- **Detect anomalies in the social network (detection)**
- **Identify actors (individuals) involved (identification)**

Subgraph Detection Problem

Goal: Develop detection framework for finding subgraphs of interest in large graphs

$G = G_b$
- background graph

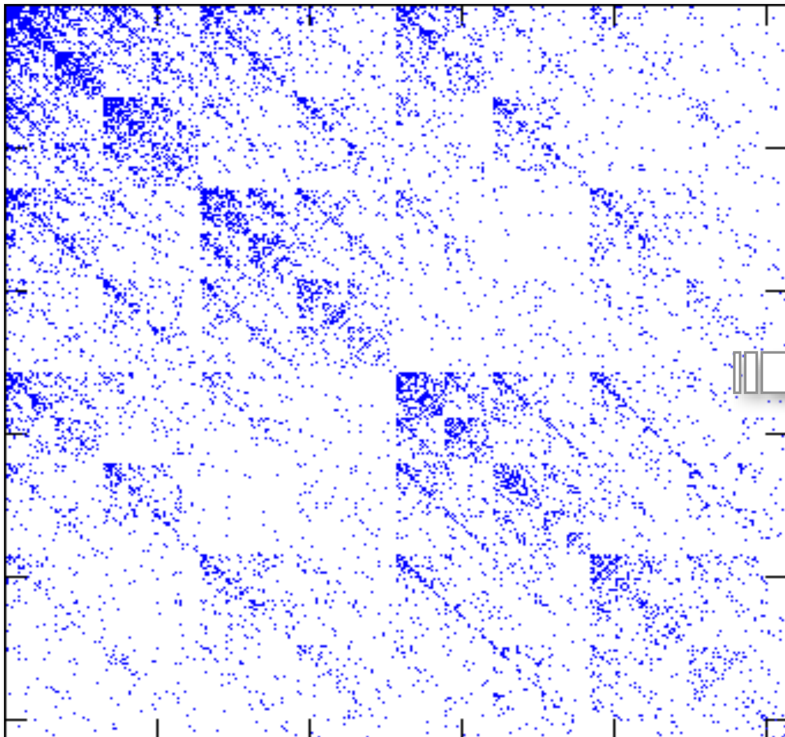


$G = G_b + G_f$
- background graph + foreground graph

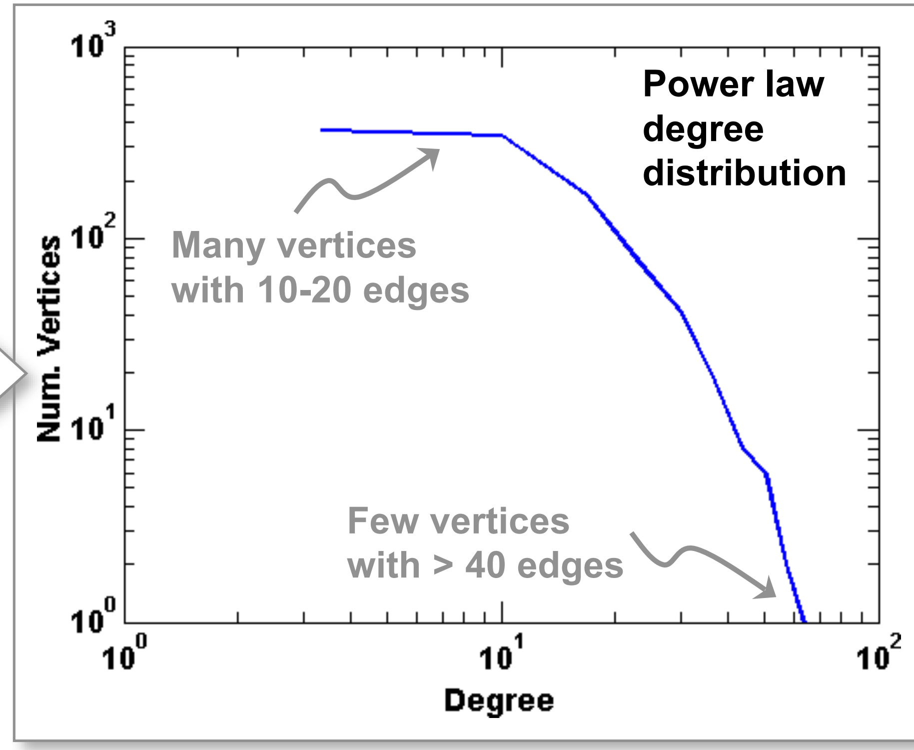
- $H_0 : G = G_b$
- $H_1 : G = G_b + G_f$
- Detection problem:
  - Given G, is H_0 or H_1 true?

Graph Detection Challenges
- Background/foreground models
- Non-Euclidean data
- High-dimensional space

H_0: Background Graph
-Power Law-



- A: Adjacency matrix of graph *G*
- 1024-vertex power law graph

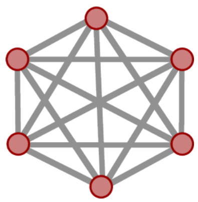


- Degree distribution of graph *G*

- Real world graphs exhibit power law properties
- Well-defined generators exist
- Structural complexity presents a challenge for detection

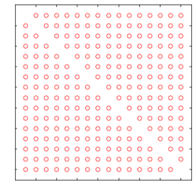
H_1: Background Graph + Foreground Graph
-Dense Subgraph in Power Law Graph-

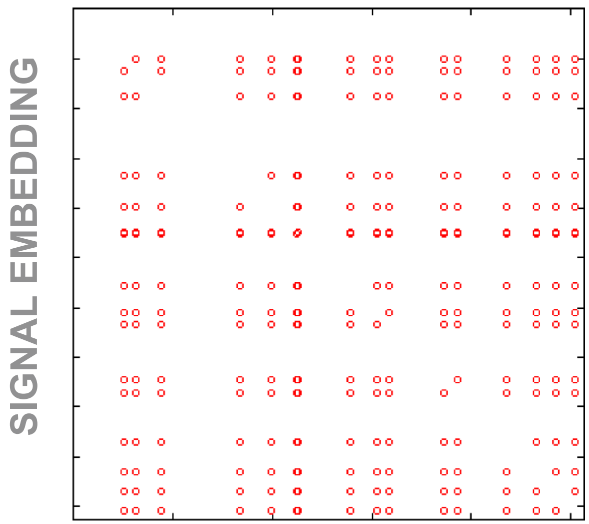
Subgraph, G_f



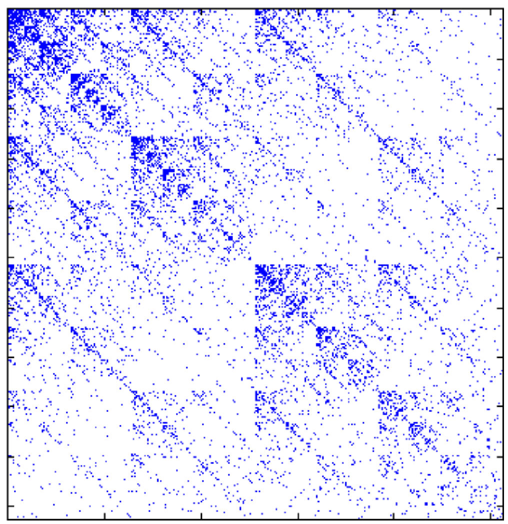
Signal (target signature): dense subgraph

G_f adjacency matrix

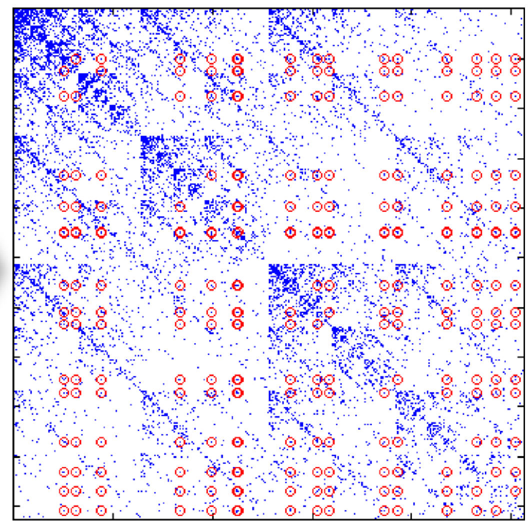




G_f on randomly selected vertices

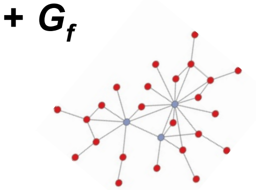


G_b



G_b + G_f

- **Realistic scenario with subgraph connected to background**
- **Well controlled but challenging example allows rigorous analysis**
- **Some subgraphs of interest exhibit high density**

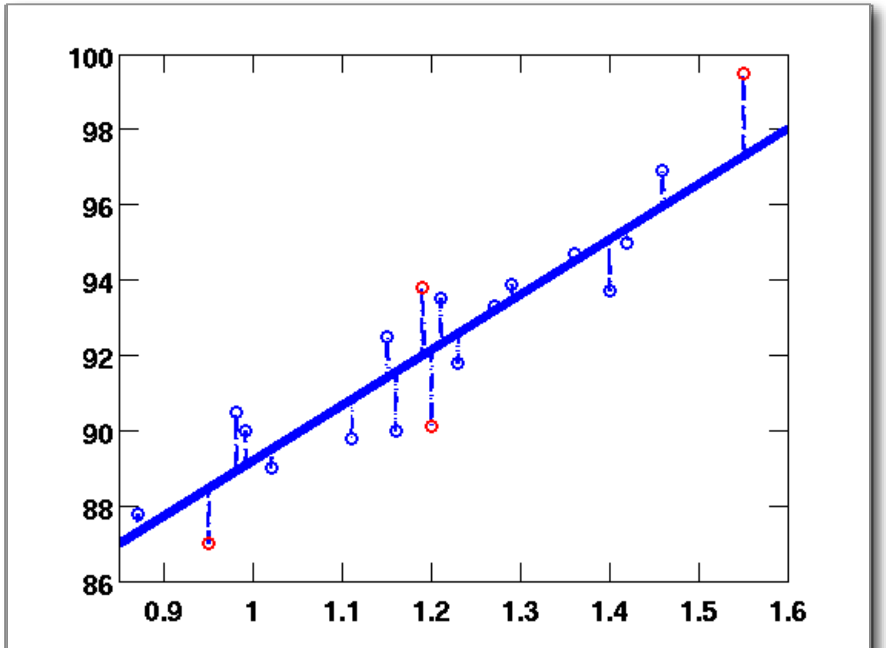
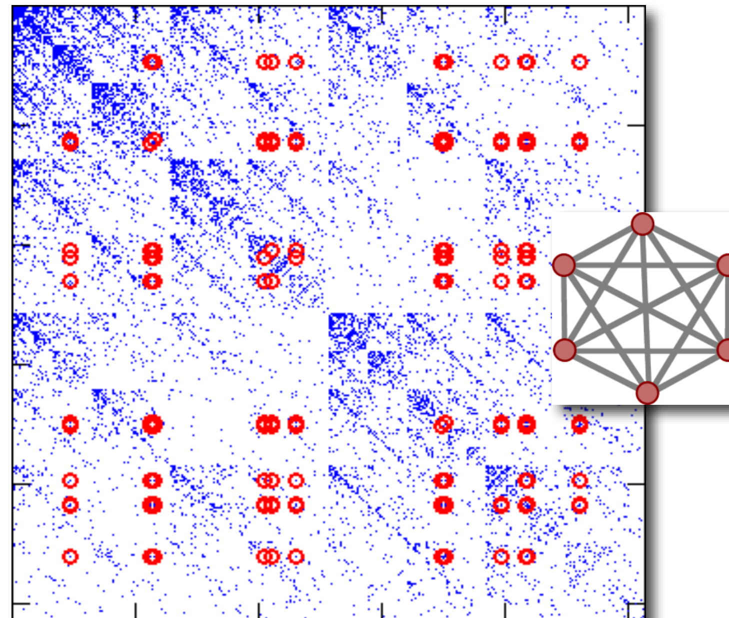


M. Skipper, *Network biology: A protein network of one's own proteins,* Nature Reviews Molecular Cell Biology 6, 824 (November 2005)

Outline

- Introduction
- **Approach**
- Handling Large Graphs
- Summary

Graph-Based Residuals Analysis

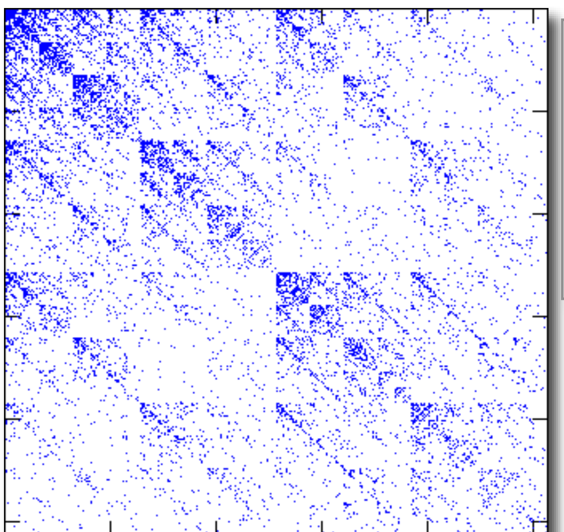
Linear Regression	Graph “Regression”
	
• Least-squares residuals from a best-fit line	• “Residuals” from a best-fit graph model
• Analysis of variance (ANOVA) describes fit	• Analysis of variance from expected topology
• “Explained” vs “unexplained” variance → signal/noise discrimination	• Unexplained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

ANALYSIS OF MODULARITY

Overview

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

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

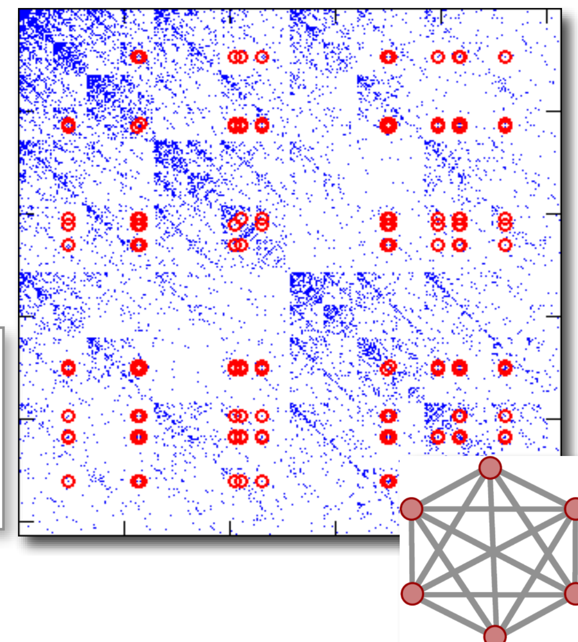


Input:

- *A*, adjacency matrix representation of *G*
- No cue

Output:

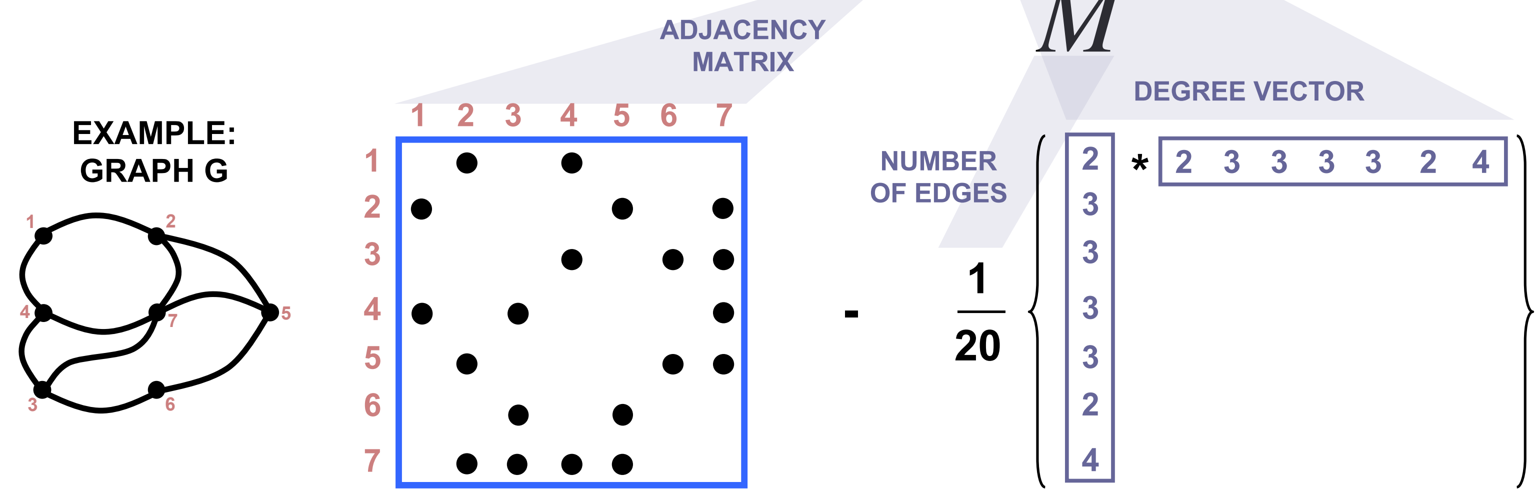
- v_s, set of vertices identified as belonging to subgraph G_f



Modularity Matrix* Construction

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

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



- Commonly used to evaluate *quality* of division of a graph into communities
- Application to subgraph detection
  - Target signatures have connectivity patterns distinct from the background
  - Can view target embedding as creation of a community

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

Eigen Decomposition

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

$$B = UDU^T$$

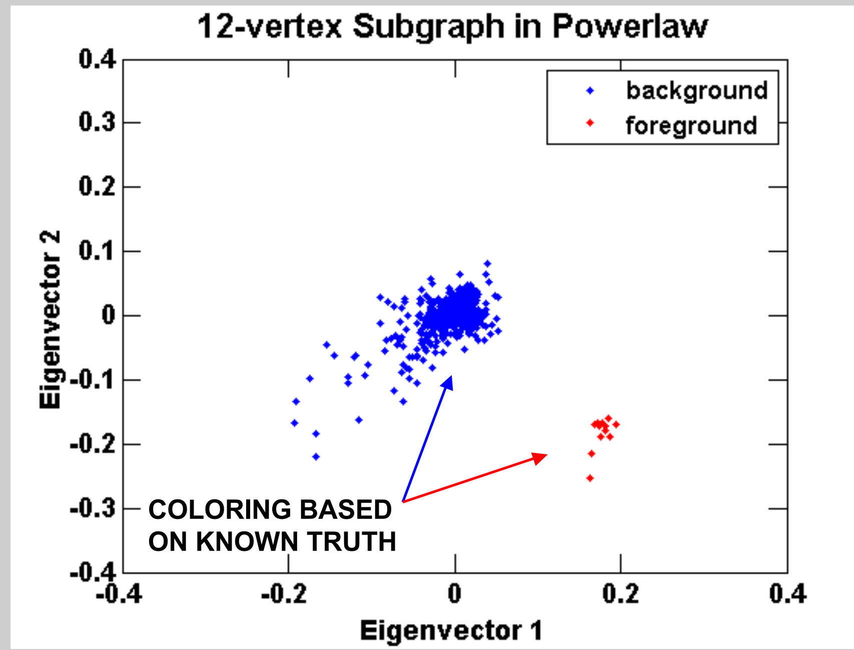
Eigenvalues, sorted by magnitude

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

Corresponding eigenvectors

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

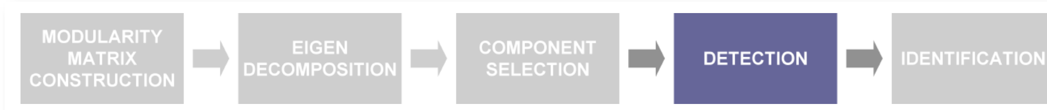
Principal components



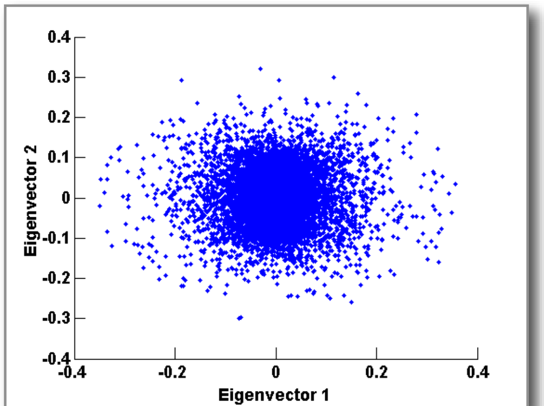
- Each point represents a vertex in *G*
- Vertices in *G*: 1024
- Vertices in G_f: 12
- Uncued background/foreground separation

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

Detection

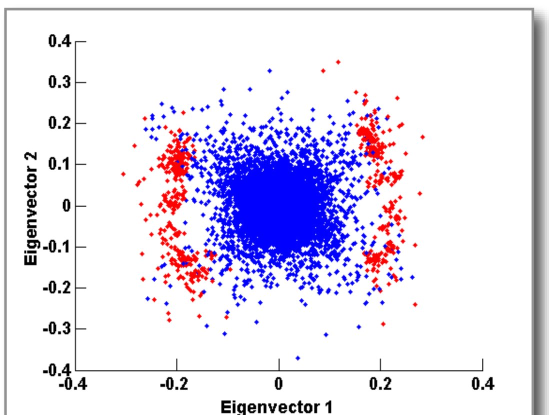


H_0



MULTIPLE TRIALS, G_b ONLY

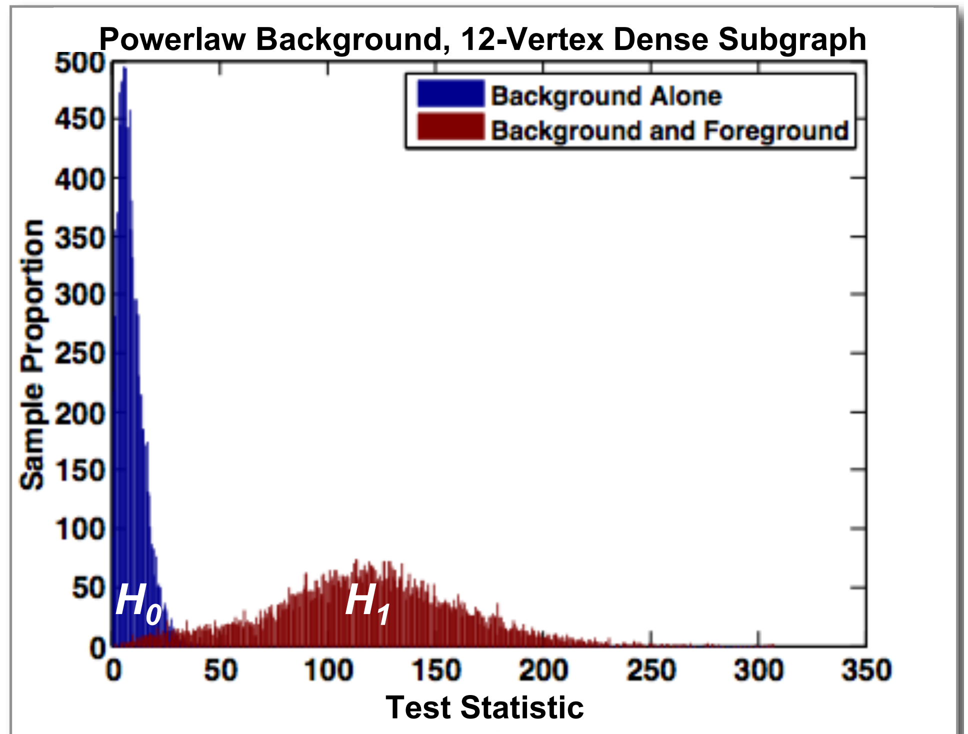
H_1



MULTIPLE TRIALS, $G_b + G_f$

TEST STATISTIC:

SYMMETRY OF THE PROJECTION ONTO SELECTED COMPONENTS

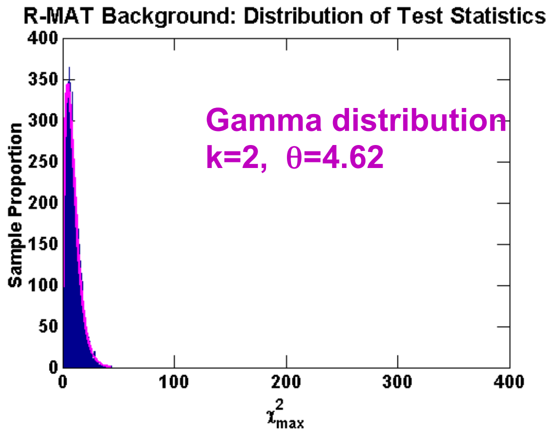


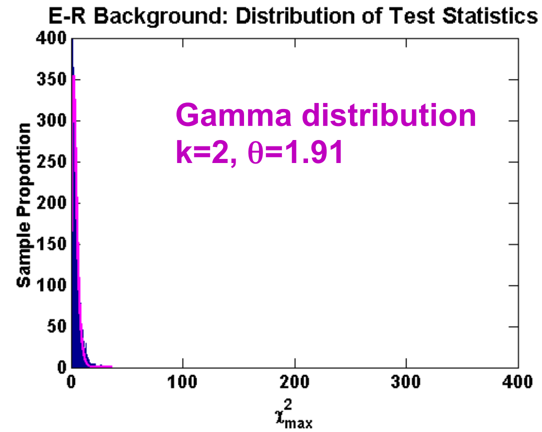
H_0 and H_1 distributions are well separated

Distribution of Test Statistics

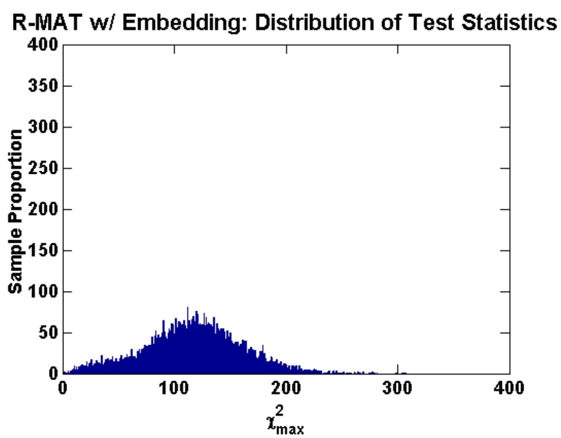
R-MAT | Erdös–Rényi

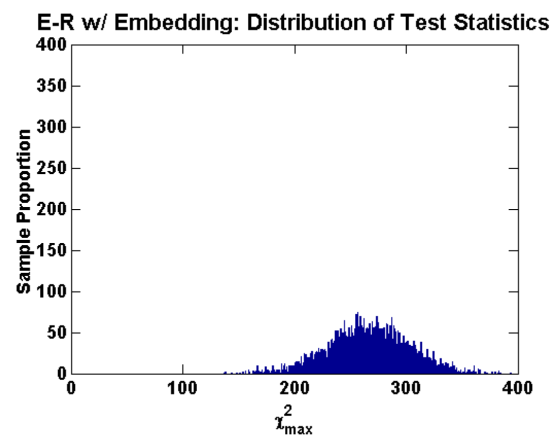
H_0





H_1





Embedding a 12-vertex fully connected subgraph significantly changes the test statistic for both background models

Detection Performance

Pd: True Positive Rate

$$TPR = \frac{TP}{P}$$

positives identified

all positives

Pfa: False Positive Rate

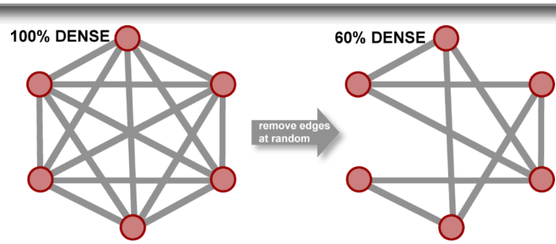
$$FPR = \frac{FP}{N}$$

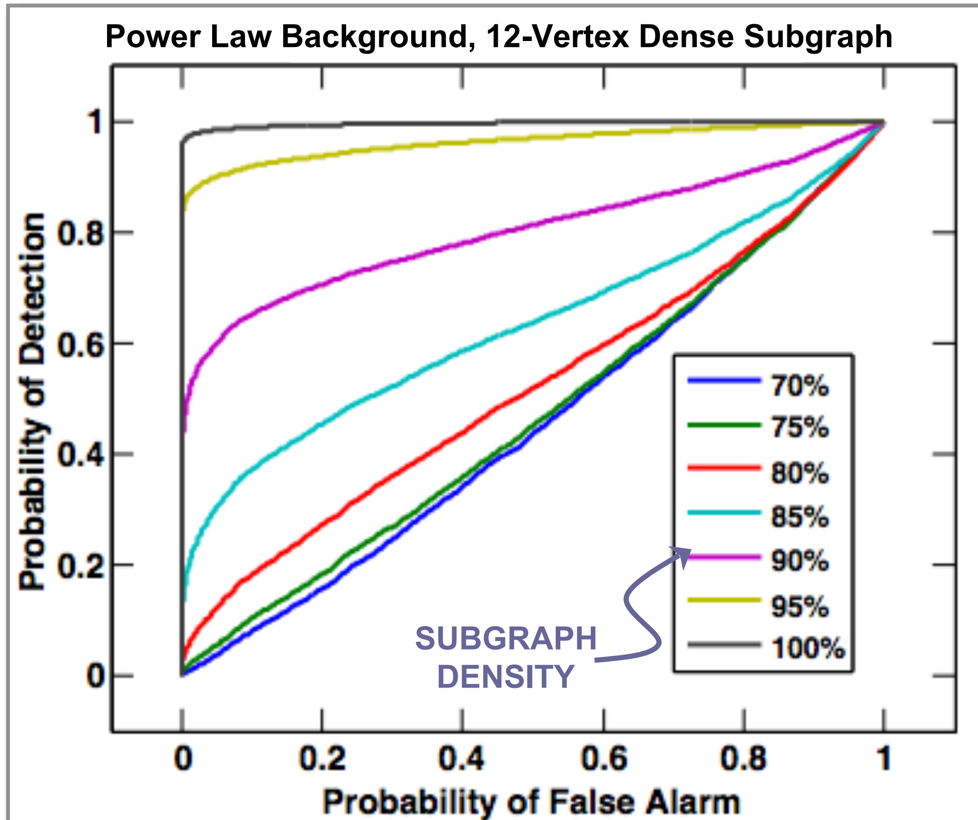
negatives identified as positives

all negatives

Detection:
Positive: *G* contains G_f

Variable Detector Characteristic: Threshold



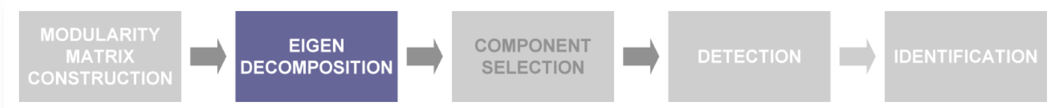


Reliable, uncued detection of tightly connected groups

Outline

- Introduction
- Approach
- **Handling Large Graphs**
- Summary

Eigendecomposition of the Modularity Matrix -Revisited-



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

Eigenvalues, sorted by magnitude

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

Corresponding eigenvectors

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

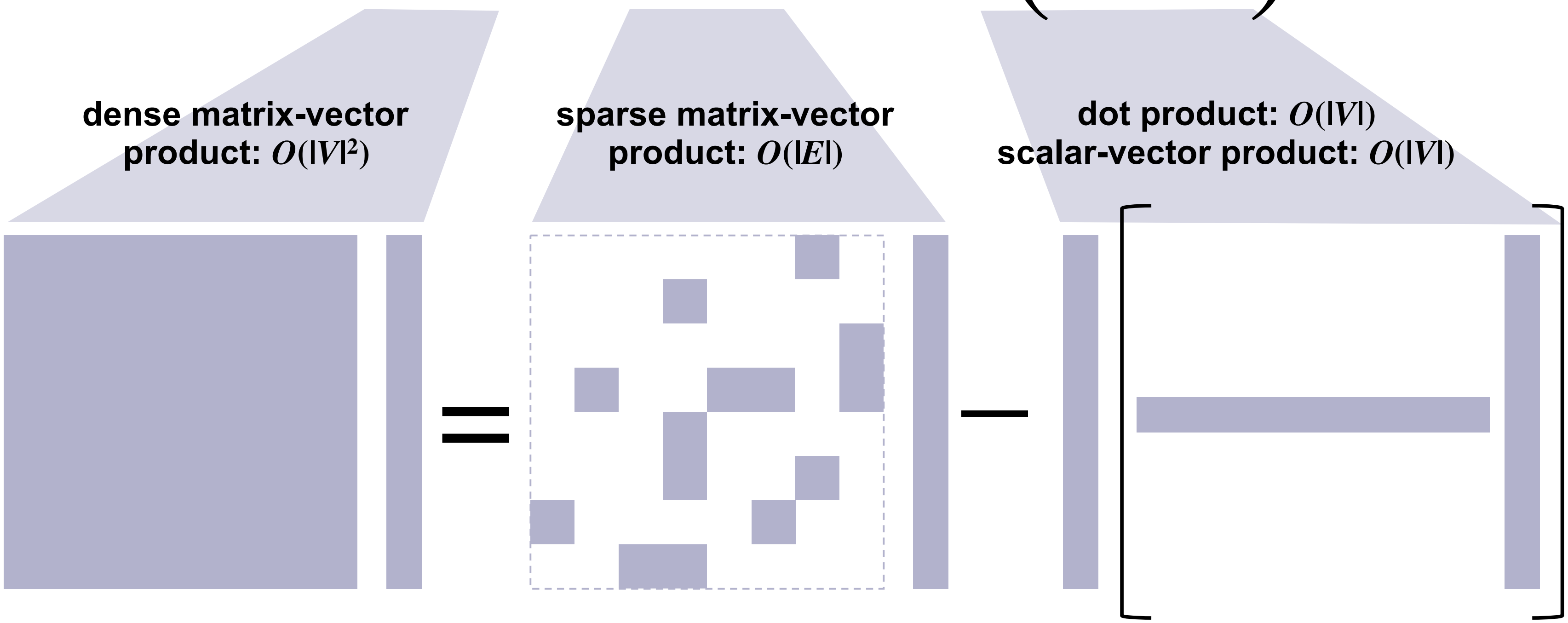
- **B is dense and thus cannot be stored for large graphs**
- **Solution: compute eigenvectors *without storing B in memory***

Approach: create a function that accepts a vector x and returns Bx without computing B; compute the eigenvectors of this function

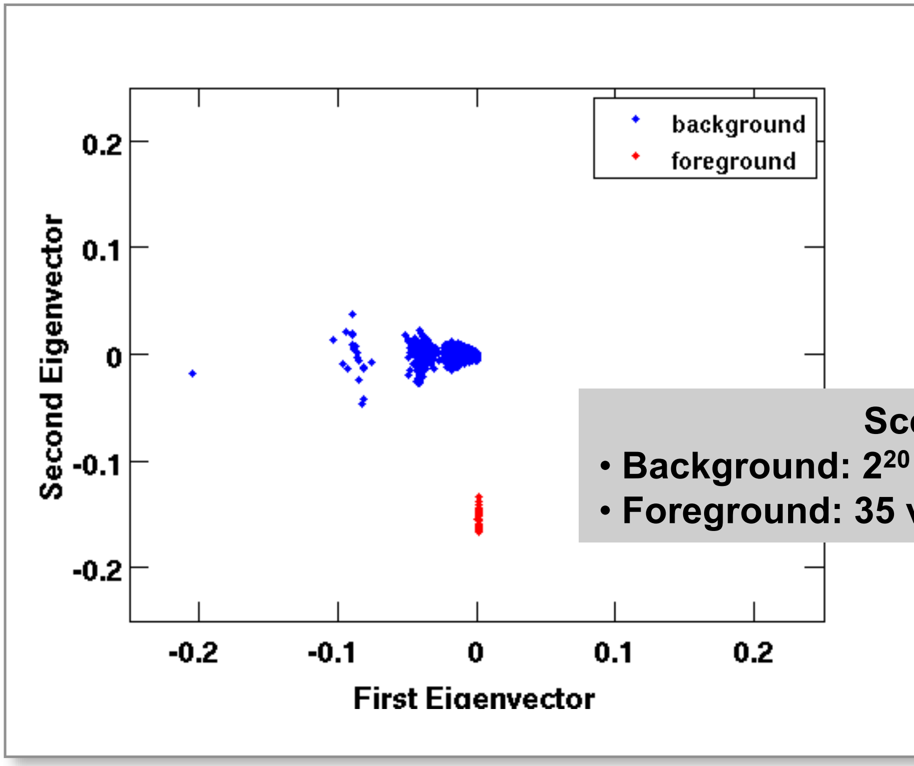
Computing Eigenvectors of Large Graphs

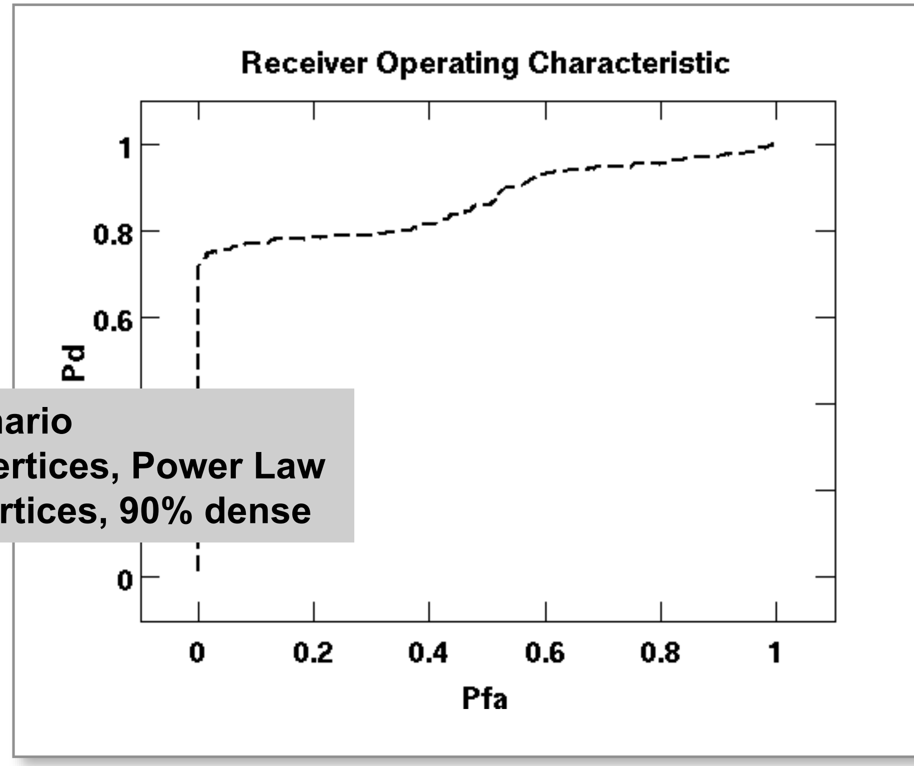
- Bx can be computed without computing B
  - Multiplication by B can be expressed as multiplication by a sparse matrix (A), plus a vector dot product and scalar-vector product
  - This method is both space- and time-efficient
- The eigenvectors of $f(x) = Ax - K(K^T x)/M$ are the eigenvectors of B

$$Bx = Ax - K(K^T x)/M$$



Detection Performance
-Large Graphs-

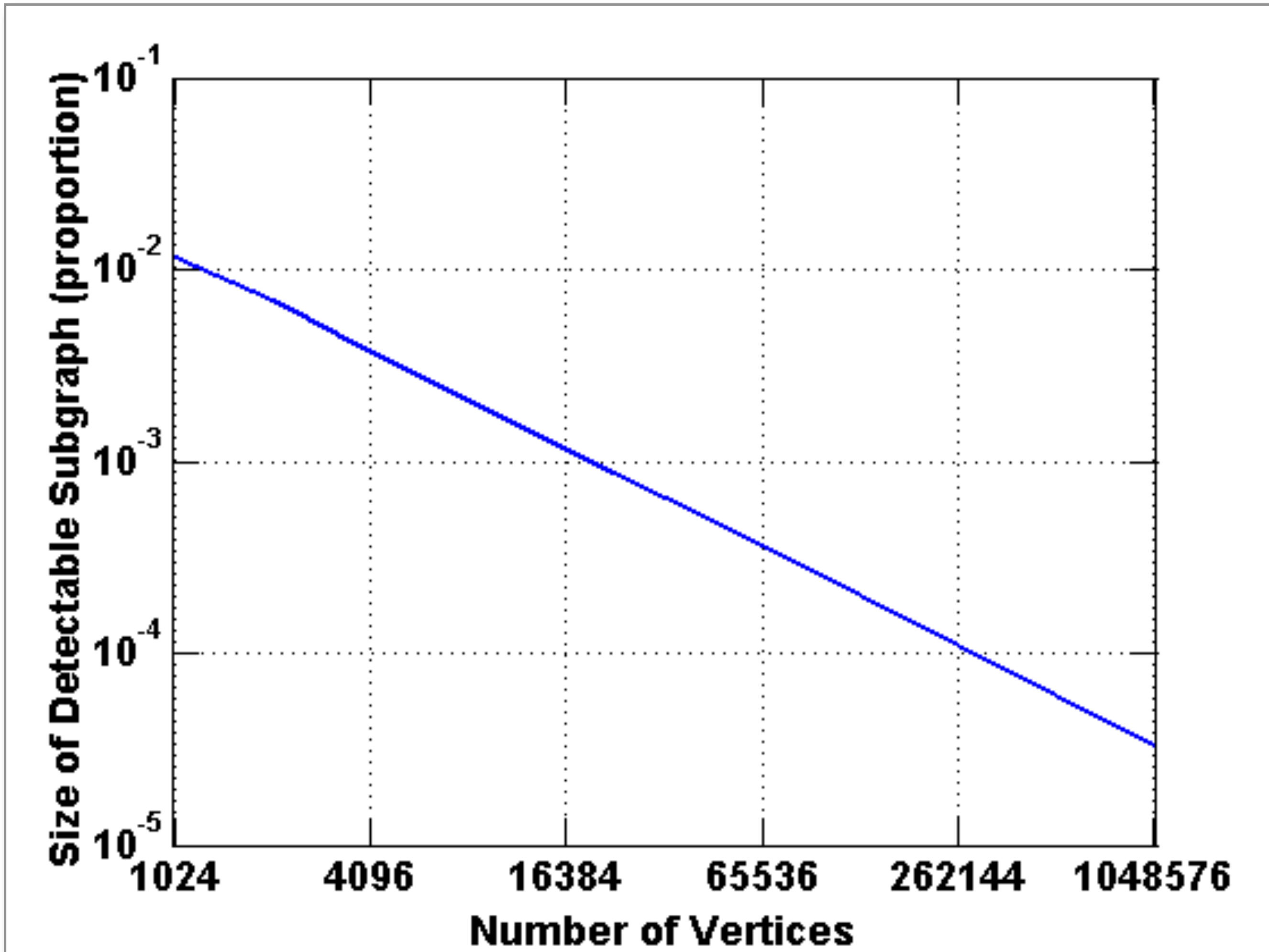




Scenario
- **Background: 2^{20} vertices, Power Law**
- **Foreground: 35 vertices, 90% dense**

- **Spectral subgraph detection algorithm can be optimized by exploiting matrix properties**
- **Analysis of 2^{20} vertex graph can be performed in minutes (~10) on a single laptop**

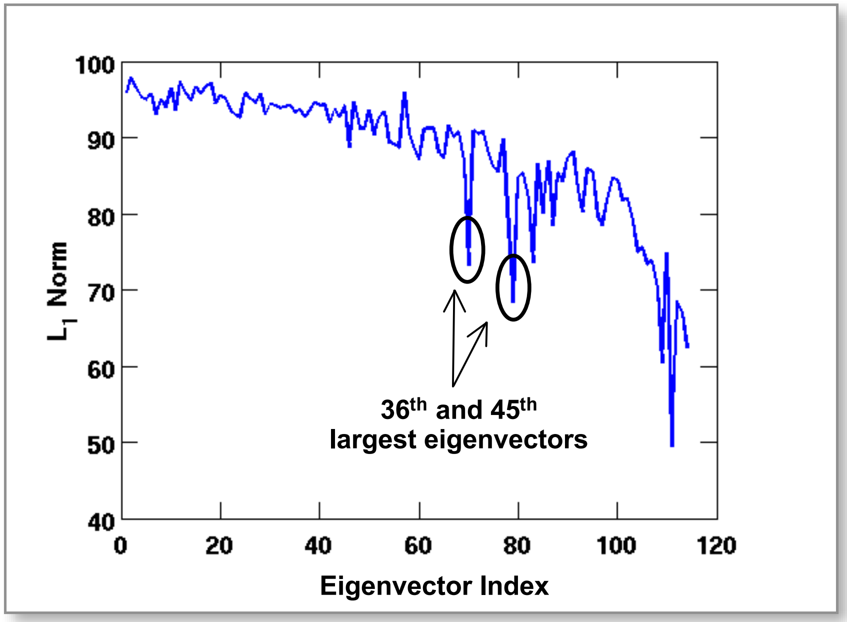
Detectability
-With Increasing Background Size-

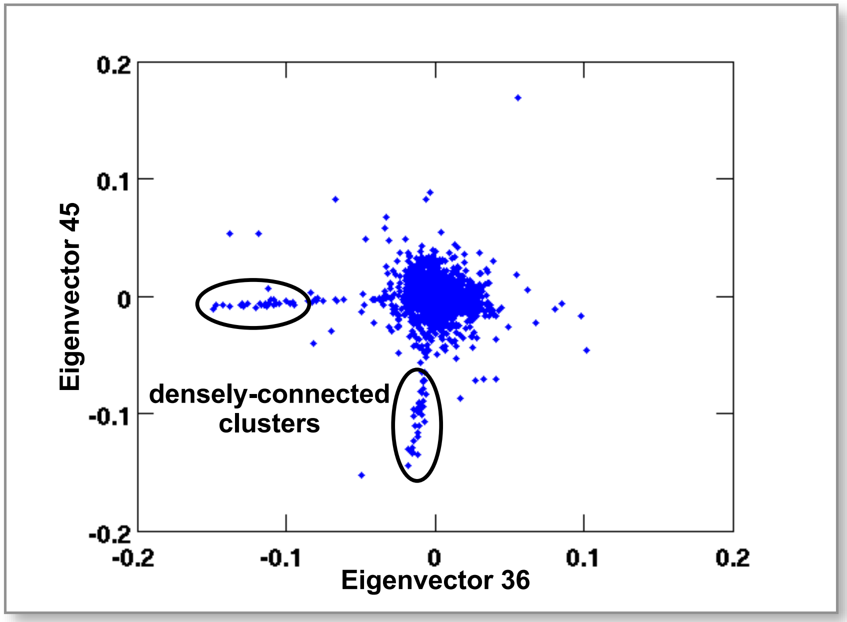


Algorithm exhibits desired performance: as size of the background graph increases, minimum detectable subgraph size remains small

Epinions Data Analysis
-Large Graph Example-

- Who-trusts-whom network from the Epinions consumer review site
  – 75,879 vertices, 405,740 edges
- Modularity matrix: too large to store in memory
- Approach: compute eigenvectors of $f(x) = Ax - K(K^T x)/M$
  – 200 eigenvectors in 155 seconds using MATLAB





Summary

- **Subgraph detection is an important problem**
- **Detection framework for graphs enables algorithms and metrics development**
- **Results on simulated and real datasets demonstrate the power of the approach**
  - **Demonstrated good detection performance**
  - **Extended approach to very large graphs**
- **Understanding background statistics (noise and clutter model) is of key importance**
- **Current research**
  - ***Weak signature foregrounds***
  - **Subgraph formation detection**