Anomaly Detection in Very Large Graphs

Modeling and Computational Considerations

Benjamin A. Miller, Nicholas Arcolano,
Edward M. Rutledge and Matthew C. Schmidt
MIT Lincoln Laboratory

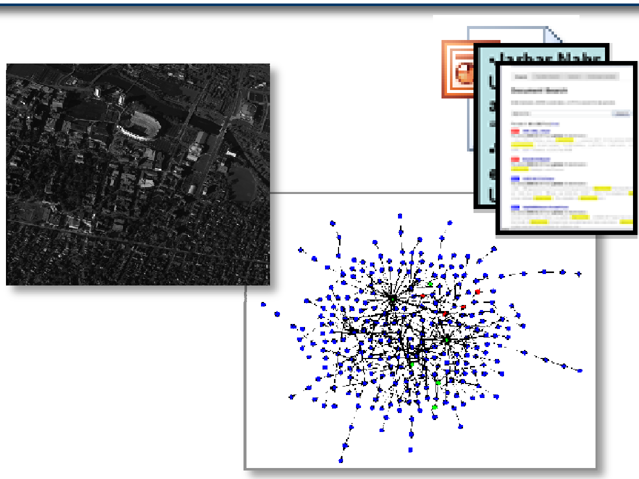
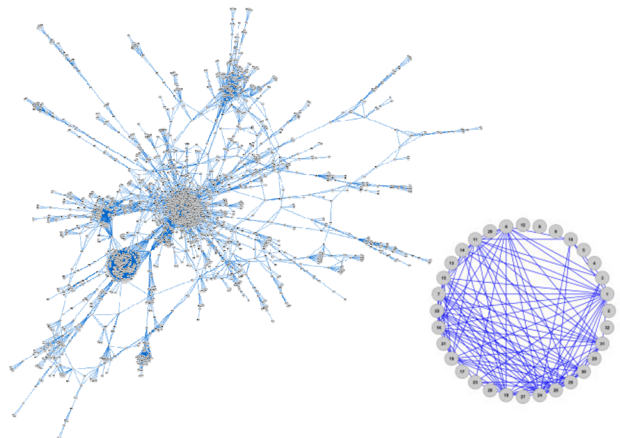
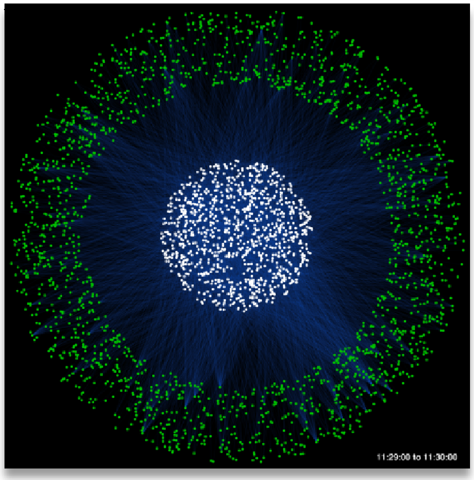
Nadya T. Bliss
ASURE

SIAM Conference on Computational Science and Engineering

27 February 2013

This work is sponsored by the Intelligence Advanced Research Projects Activity (IARPA) under Air Force Contract FA8721-05-C-0002. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

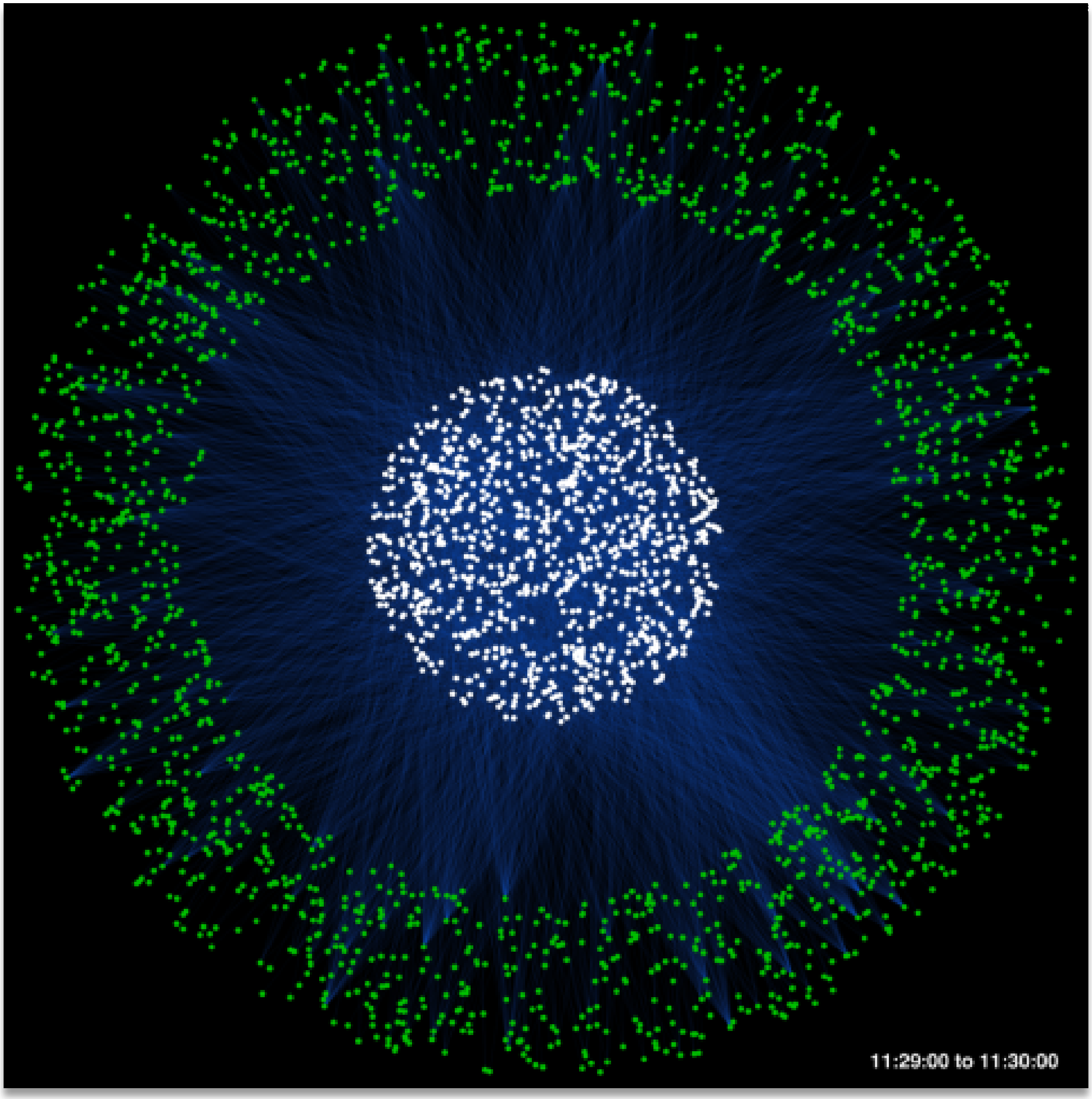
Applications of Graph Analytics

ISR	Social	Cyber
		
• Graphs represent entities and relationships detected through multi-INT sources • 1,000s – 1,000,000s tracks and locations • GOAL: Identify anomalous patterns of life	• Graphs represent relationships between individuals or documents • 10,000s – 10,000,000s individual and interactions • GOAL: Identify hidden social networks	• Graphs represent communication patterns of computers on a network • 1,000,000s - 1,000,000,000s network events • GOAL: Detect cyber attack or malicious software

Cross-Mission Challenge:
Detection of subtle patterns in massive, multi-source, noisy datasets

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Application Example: Botnet Detection in Web Proxy Data



Graph Statistics

- 90 minutes worth of traffic
- 1 frame = 1 minute of traffic
- Number of source computers: 4,063
- Number of web servers: 16,397
- Number of logs: 4,344,148

Malicious Activity Statistics

- Number of infected IPs: 1
- Number of event logs: 16,000
- % infected traffic: 0.37%
- Existing tools did not detect event
- Detection took *10 days* and required manual log inspection

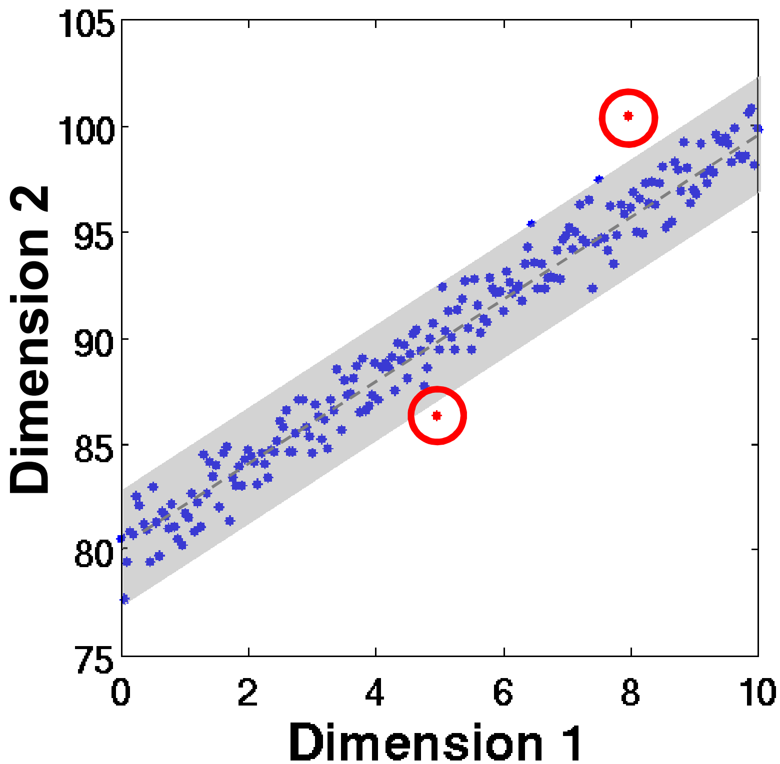
Challenge: Detect weak signal activity in large, noisy background

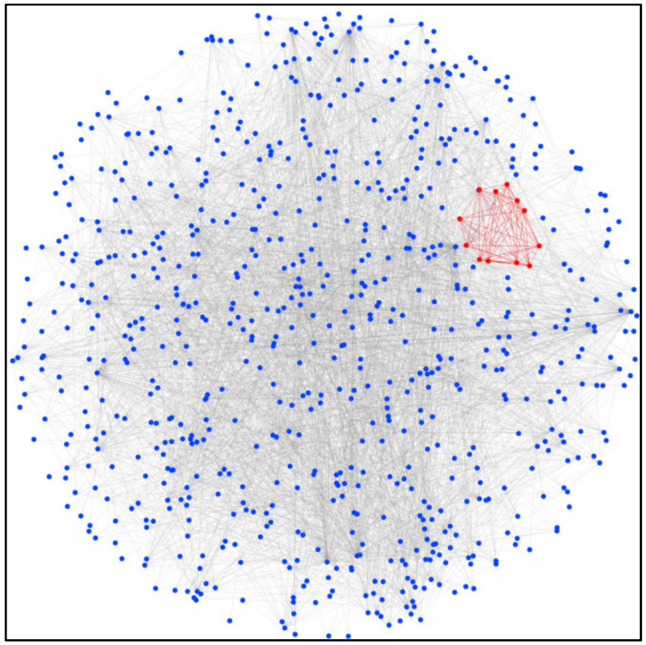
See M. Schmidt et al., *Utilizing Spectral Methods for Uncued Anomaly Detection in Large-scale, Dynamic Networks*, in MS250

Outline

- Introduction
- **Algorithmic Framework**
- Recent Algorithm Developments
- Demonstration at Scale
- Model Complexity
- Summary

Analysis of Graph Residuals





Linear Regression

Graph Regression

Graph Processing Chain

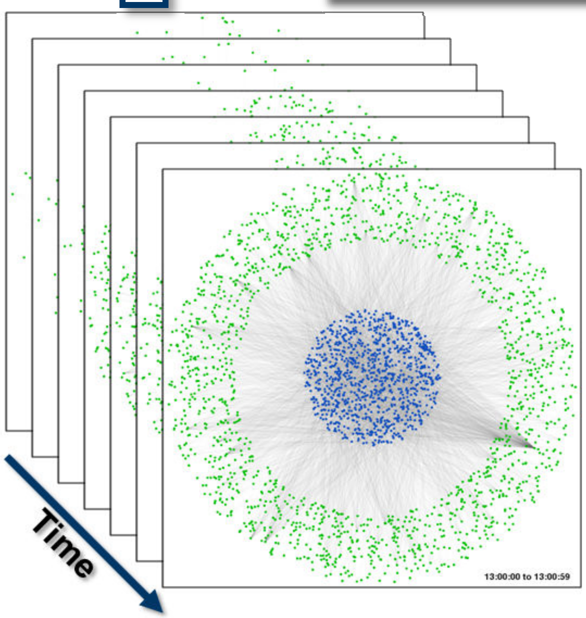
TEMPORAL INTEGRATION → GRAPH MODEL CONSTRUCTION → RESIDUAL DECOMPOSITION → COMPONENT SELECTION → ANOMALY DETECTION → IDENTIFICATION

Input

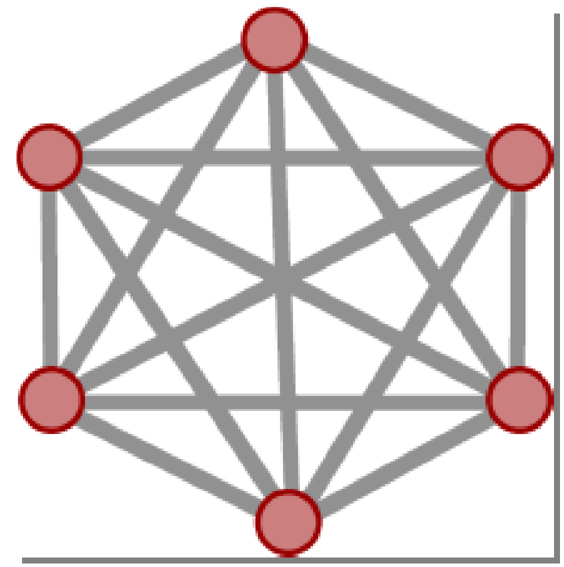
- A(t) – A series of adjacency matrices representing a time series of graphs G(t)
- No cue

Output

- $v_s(t)$ – A potentially dynamic set of vertices representing anomalous subgraphs of G(t)



Graph processing chain applies a signal processing approach to anomalous subgraph detection



LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Research Focus Areas

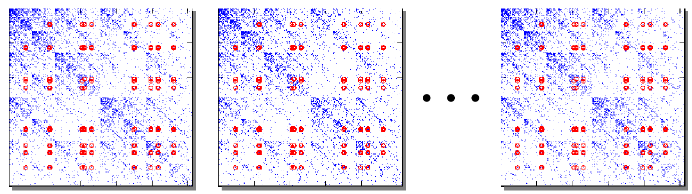
TEMPORAL INTEGRATION → GRAPH MODEL CONSTRUCTION → RESIDUAL DECOMPOSITION → COMPONENT SELECTION → ANOMALY DETECTION → IDENTIFICATION

Residuals Models

$$B = A - \mathrm{E}[A]$$

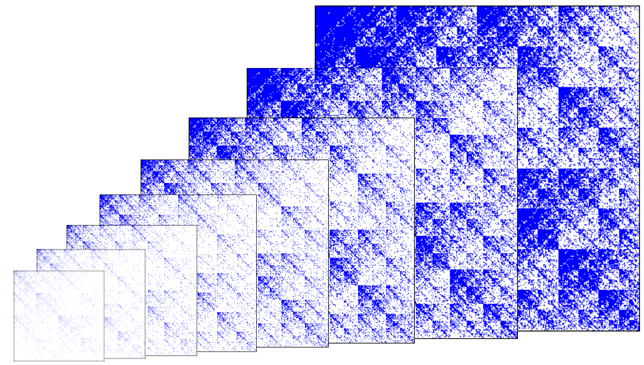
- What information (other than degree) can be used to derive graph residuals?

Integration Techniques



- How can we combine the recent past to better determine the presence of an anomaly?

Massive Data Analysis



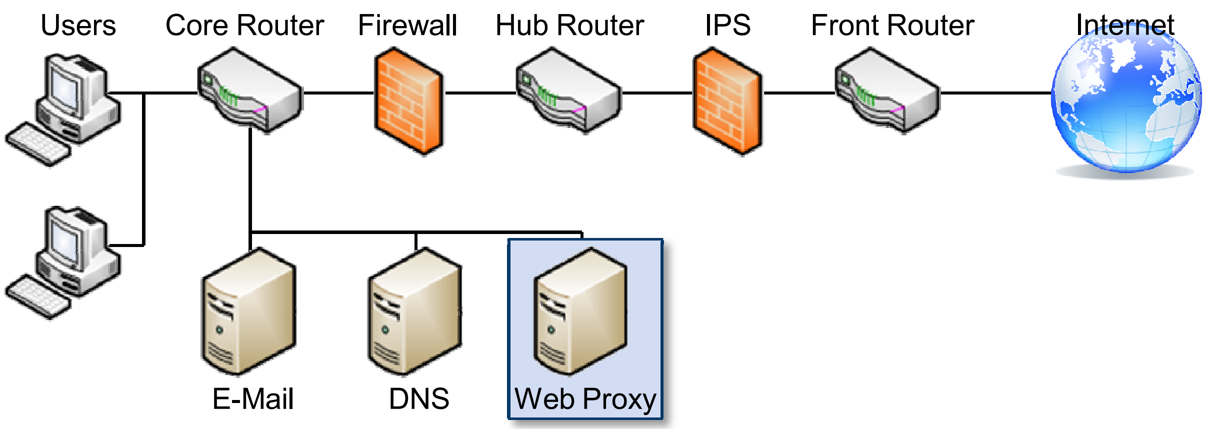
- How can we extend and adapt our techniques for data on the scale of 1B vertices?

All algorithm research is informed by properties of real data

Datasets

Thompson Reuters' Web of Science

- Database of over 42 million documents between 1900 and 2010
- Records include authors, subjects, and cited documents
- Resolution of 1 year
- Various graphs can be constructed from data (e.g. , citation and coauthorship)



Web Proxy Logs

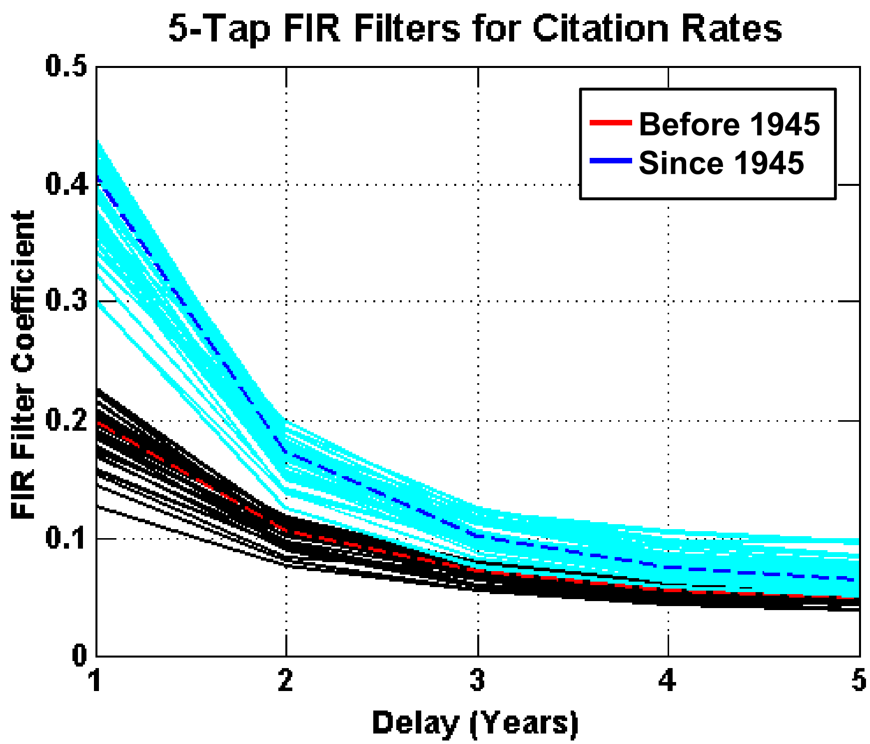
- Logs from a web proxy server in an institution's local area network
- About 4000 internal computers connecting to 250k web servers
- Resolution of 1 second
- Connectivity graph augmented by additional fields (e.g., URL, referrer)

Outline

- Introduction
- Algorithmic Framework
- **Recent Algorithm Developments**
- Demonstration at Scale
- Model Complexity
- Summary

Preferential Attachment with Memory

- **Preferential attachment is a popular model for graph evolution**
  - **New nodes connect to existing ones with probability proportional to degree**
  - **Does not account for recency**
- **New model: generate new attachment rates based on a linear combination of the number of recent connections**
- **Current attachment rate for v_i is modeled as** $\lambda_i(t) = \sum_{m=1}^{M} k_i^{\mathrm{in}}(t-m)h(m)$



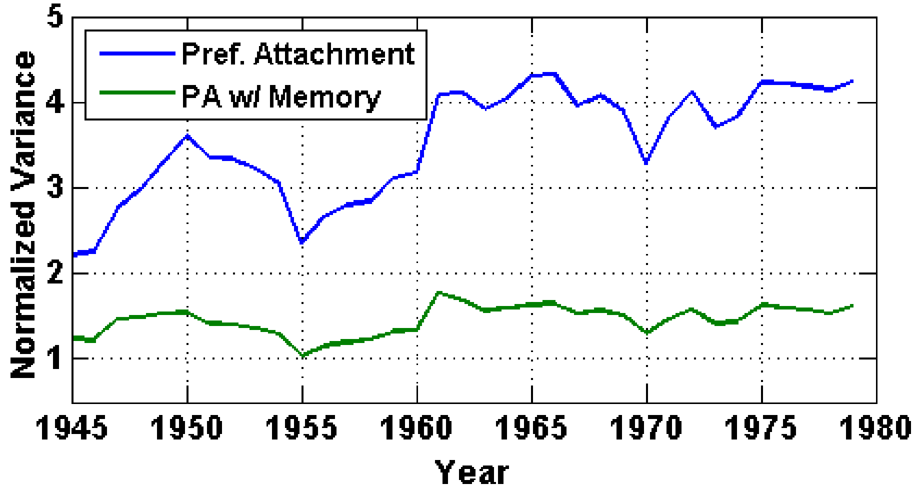
Maximum likelihood fit of coefficients: more correlation with recent citations than older ones

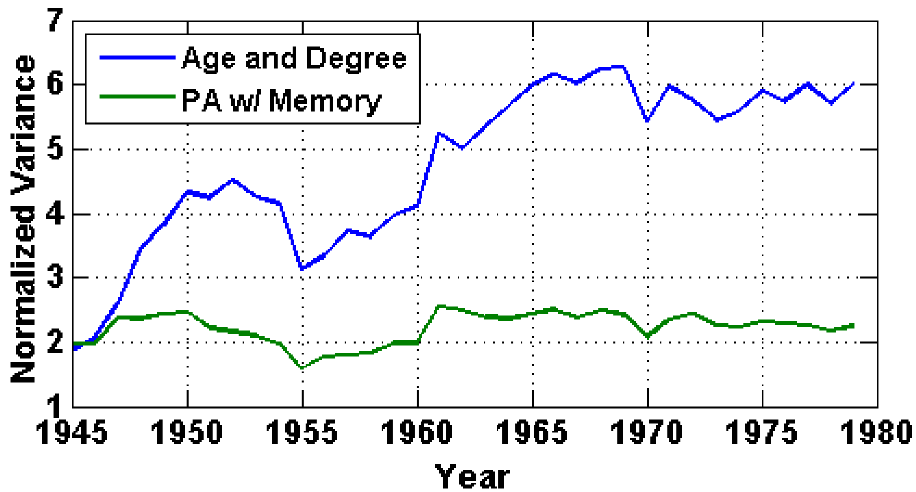
New perspective on preferential attachment proves powerful for topology-based modeling of dynamic data

B. A. Miller and N. T. Bliss, "A stochastic system for large network growth," *IEEE Signal Process. Lett.*, vol. 19, no. 6, pp. 356–359, 2012.

Fitting Model to Citation Data

- **Existing models use attachment probabilities based on degree and vertex age**
  - **New model also encompasses existing models using 1-tap FIR and autoregressive models**
- **When fitting, we use these formulas for Poisson rates $\lambda_i(t)$ and find the maximum likelihood estimate for the parameters**
- **Evaluate fit by reduced chi-squared statistic:** $\frac{1}{|V|}\sum_{i=1}^{|V|}\frac{(k_i(t)-\lambda_i(t))^2}{\lambda_i(t)}$



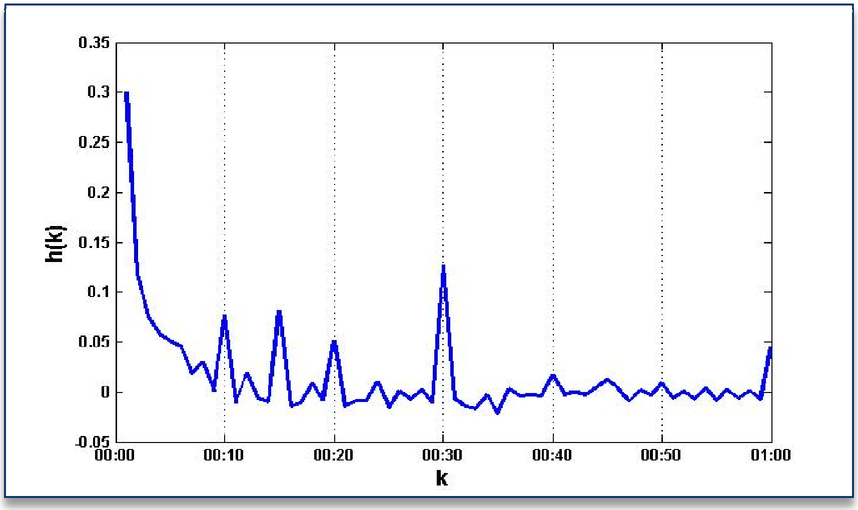


Lower normalized variance implies a better fit to the true citation rates, suggesting more robust residuals analysis

Moving Average Adjacency Filter

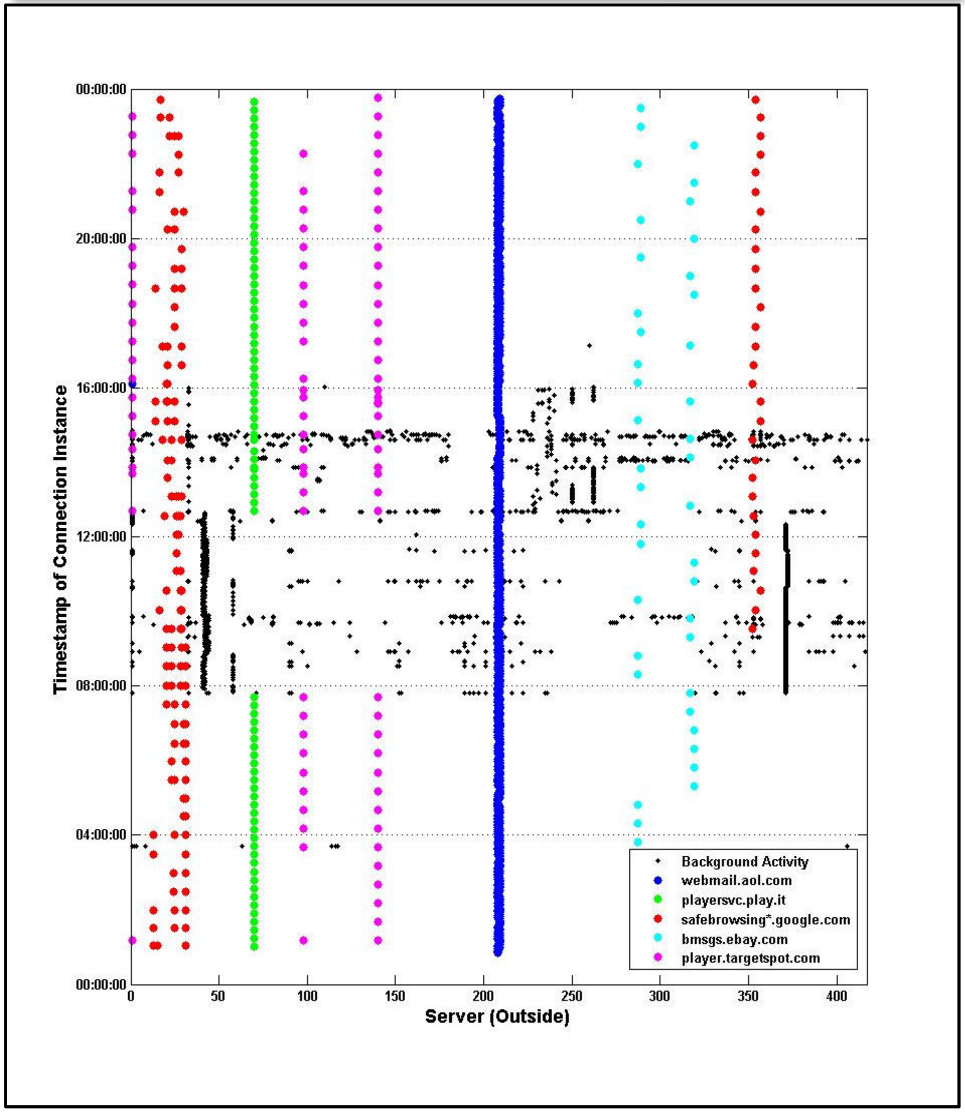
A significant portion of web proxy activity comes from automated services that regularly communicate with a server or set of servers

Linear fit of current connections to previous observations



Objective: Leverage this behavior to predict these edges and filter them out

Server connections for a single source



LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Detecting Anomalous Coordinated Behavior

Objective

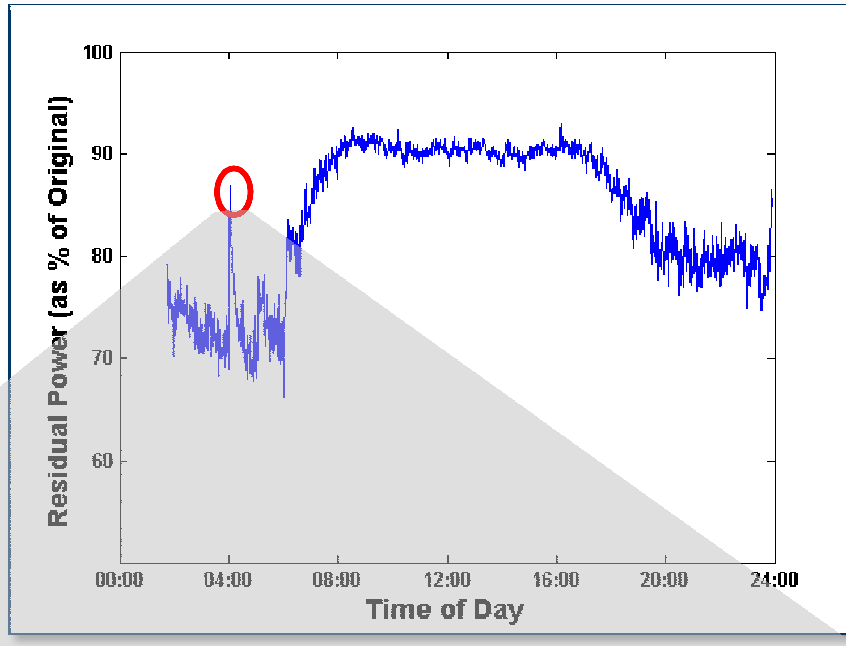
- Identify sources and servers that are communicating in a coordinated but non-repetitive manner
- Behavior is characteristic of some malicious activities (e.g. DDoS attacks, Botnets)
- Used a filter based on the previous 1 hour of traffic

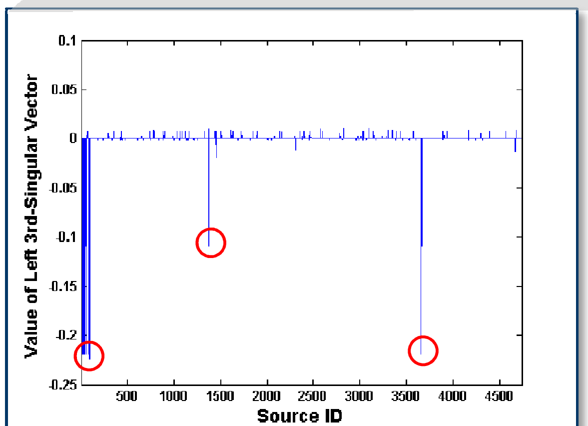
Anomaly Characteristics

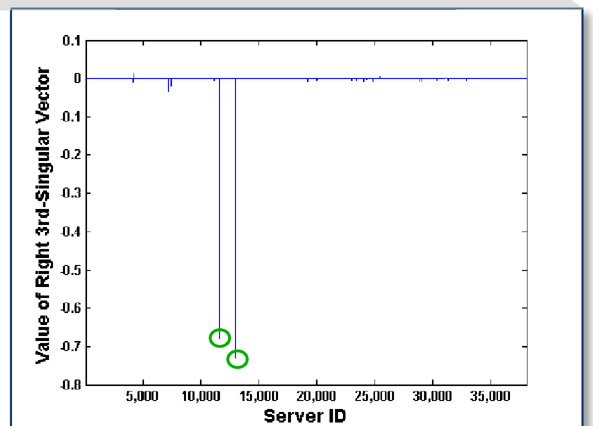
Anomalies are time steps where the dynamic model is an anomalously bad fit for the observed graph

Able to identify an anomalous event of 20 sources connecting to 2 servers in 3 seconds

Dynamic Model Goodness-of-Fit



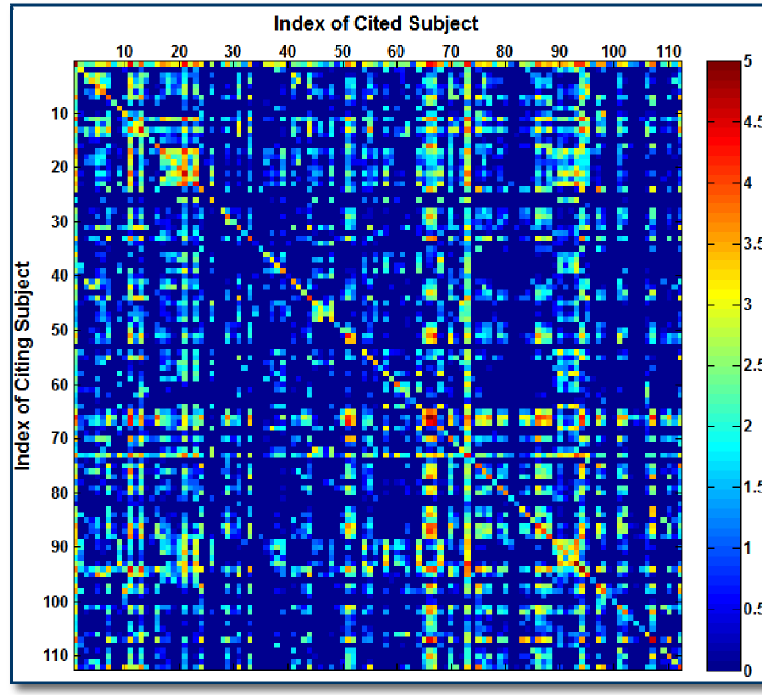




Generalized Linear Models for Graph Regression

- **After building a graph, there is typically substantial side information**
  - **This can be viewed as attributes of the nodes and edges**
  - **Example: citation graph with author, subject, and journal attributes**

- **Model data in a regression framework**
  - **Use an extension of linear regression to data in a restricted domain**



Generalized Linear Model (GLM)

- **Expected observation:**

$$E(y_i) = g(\mathbf{x}_i^T \beta)$$

observed data (edges) | **attributes for *i*-th observation** | **regression weights**

- g **: R** $\rightarrow [0,1]$ **is the "link function"**
- **Example link: *logistic function***

$$g(x) = \frac{1}{1 + \exp(-x)}$$

- **GLM incorporates additional metadata into edge probabilities**
- **Can be used to model the effect of subject area on citation probability**

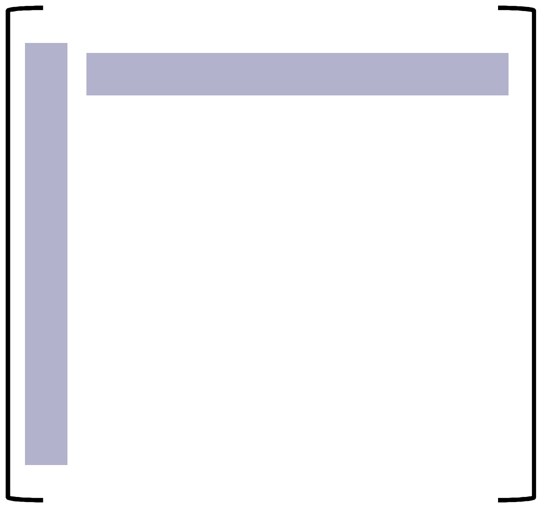
N. Arcolano and B. A. Miller, "Statistical models and methods for anomaly detection in large graphs," ***SIAM Ann. Meeting*, 2012.**

Outline

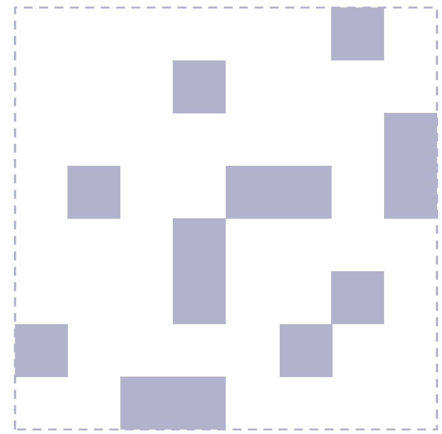
- Introduction
- Algorithmic Framework
- Recent Algorithm Developments
- **Demonstration at Scale**
- Model Complexity
- Summary

Complexity of Rank-1 and Sparse Models

rank-1 expected value model



sparse expected value model



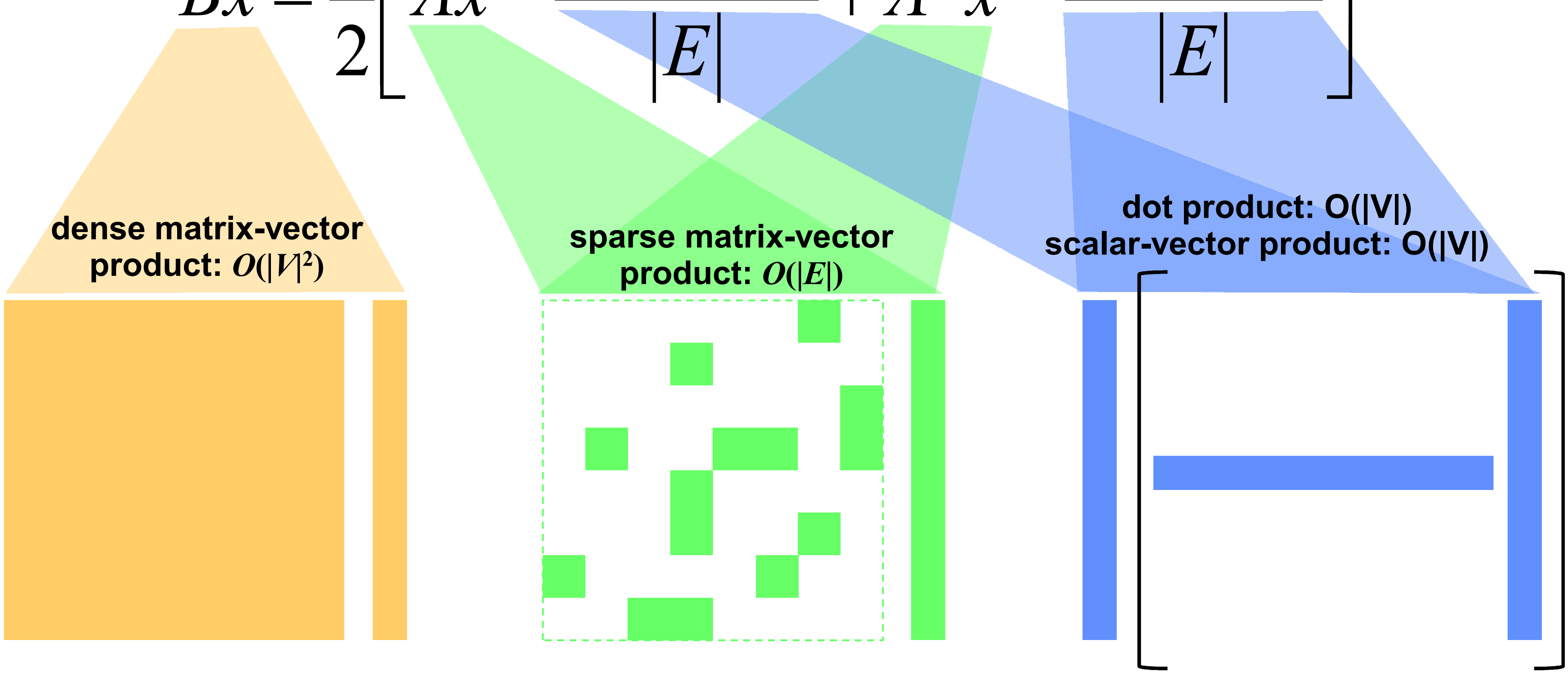
- In most applications, adjacency matrices are sparse
- Expected value matrices, typically, are not
- Special structures can be exploited for residuals computation
  - For rank-1 expected values, such as modularity and preferential attachment, this can be performed as a dot product and scalar-vector product
    - This yields a complexity of $O((|E|k+|V|k^2+k^3)h)^*$ to compute k eigenvectors
  - For sparse expected values, such as the moving average filter with memory depth T, this is a sequence of sparse matrices
    - This yields a complexity of $O((T|E|k+|V|k^2+k^3)h)^*$ to compute k eigenvectors

*h: number of iterations

Computational Scaling

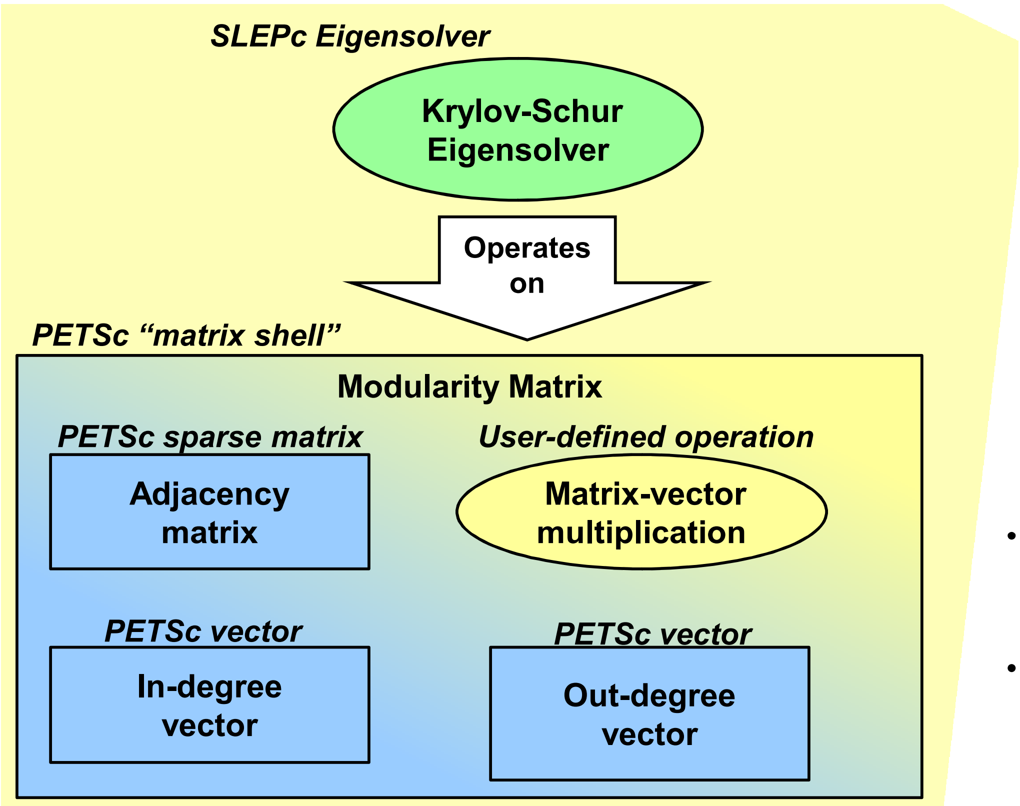
Matrix-vector multiplication is at the heart of eigensolver algorithms

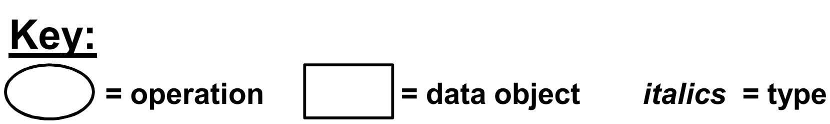
$$Bx = \frac{1}{2}\left[Ax - \frac{k_{out}(k_{in}^T x)}{|E|} + A^T x - \frac{k_{in}(k_{out}^T x)}{|E|} \right]$$

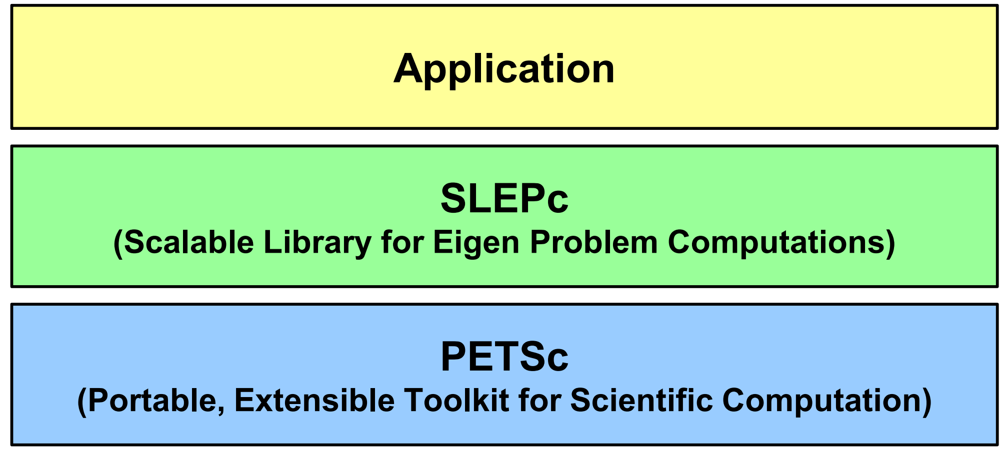


***Bx* can be computed without storing *B* (modularity matrix)**

Implementing Eigen Decomposition of the Modularity Matrix using SLEPc





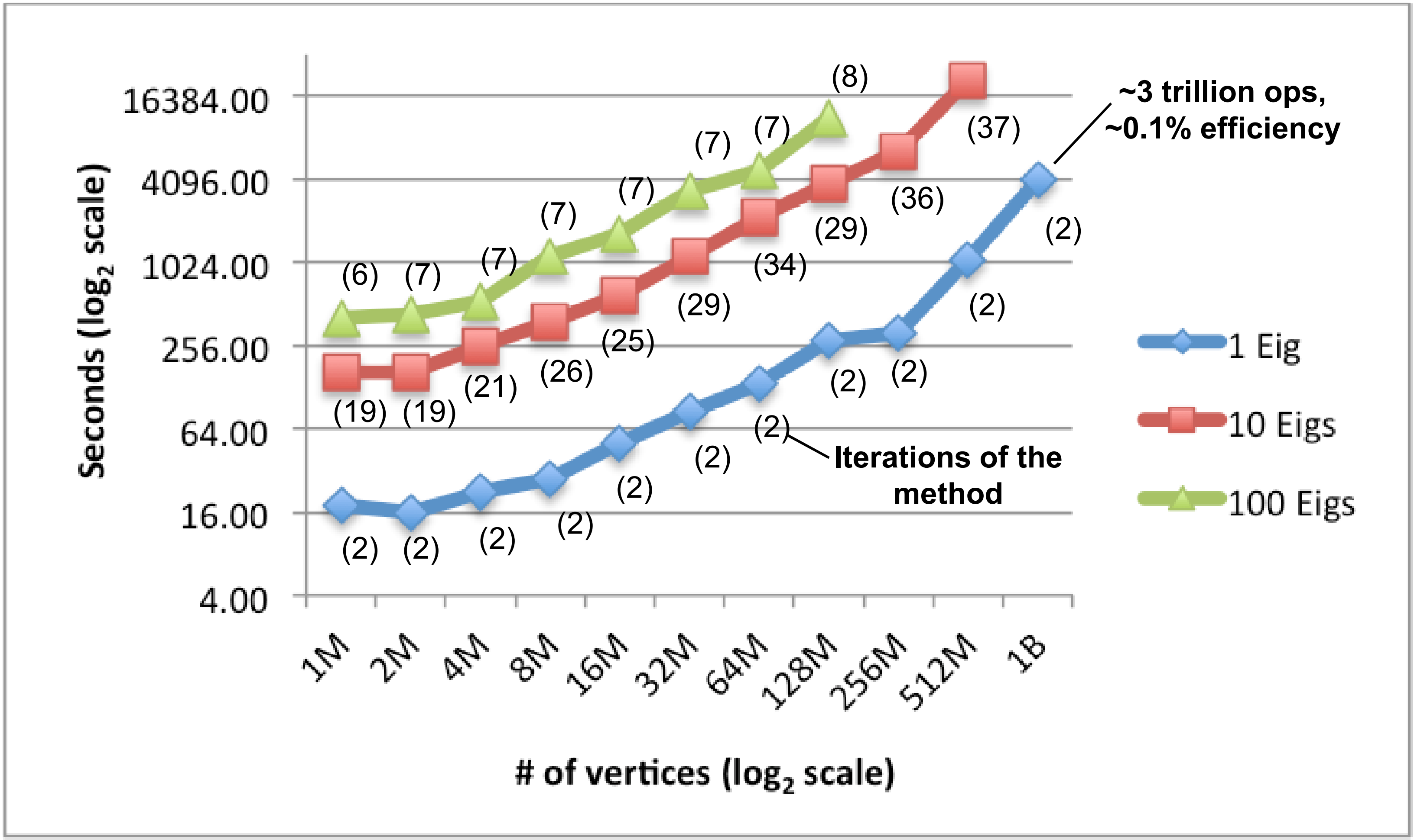


- PETSc "matrix shell" enables efficient modularity matrix implementation
- Used default PETSc/SLEPc build parameters and solver options
  - Compressed Sparse Row (CSR) matrix data structure
  - Double precision (8 byte) values for matrix and vector entries
  - Krylov-Schur eigensolver algorithm
- Limitation: current implementation will not scale past 2^{32} vertices
  - Uses 32 bit integers to represent vertices
  - Only tested up to 2^{30} vertices

SLEPc/PETSc supports efficient implementation of modularity matrix eigen decomposition

E. M. Rutledge, B. A. Miller, and M. S. Beard, "Benchmarking parallel eigendecomposition for residuals analysis of very large graphs," in Proc. IEEE HPEC, 2012.

Results
SLEPc 64 Node Average Execution Time



10 leading eigenvalues (64M vertex data set):

$\lambda_1 = 85.403845$

$\lambda_2 = 41.146193$

$\lambda_3 = 41.093851$

$\lambda_4 = 40.993092$

$\lambda_5 = 40.963347$

$\lambda_6 = 40.907482$

$\lambda_7 = 40.854498$

$\lambda_8 = 40.824815$

$\lambda_9 = 40.765026$

$\lambda_{10} = 40.735158$

- **Able to compute 2 eigenvectors for 1 billion node graph (in ~9 hrs)**
- **Problem size limited by memory**
- **Larger problems could be solved with >64 compute nodes**

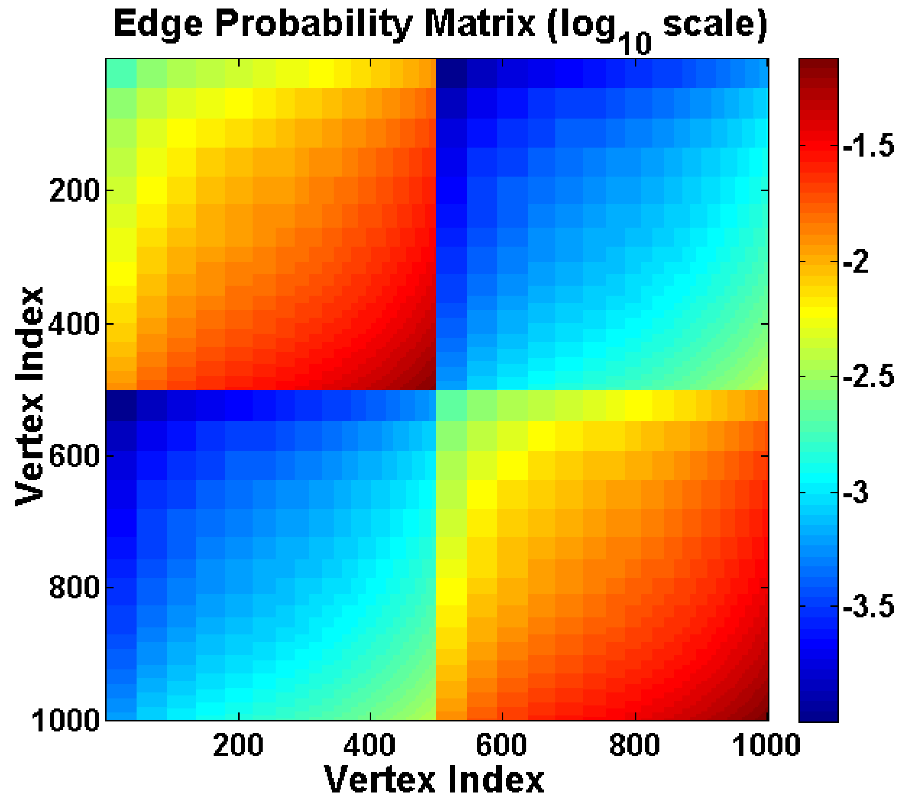
E. M. Rutledge, B. A. Miller, and M. S. Beard, "Benchmarking parallel eigendecomposition for residuals analysis of very large graphs," in Proc. IEEE HPEC, 2012.

Outline

- Introduction
- Algorithmic Framework
- Recent Algorithm Developments
- Demonstration at Scale
- **Model Complexity**
- Summary

GLM Empirical Example

- **10,000-trial Monte Carlo anomaly detection simulation**
- **For each trial, the observation is a 1,000-vertex graph**
- **Each graph is generated by a Chung–Lu/Stochastic Blockmodel hybrid**
  - **Partitioned into two halves**
  - **Each half has higher probability of internal than external connectivity**
  - **Each vertex also has a "popularity" parameter**
- **Two scenarios for embedded anomaly (8-vertex Erdős–Rényi graph)**
  - **All 8 vertices on one side of the partition**
  - **4 vertices on each side**
- **Detection based on spectral norm of residuals matrix**



$$p_{ij} = \frac{1}{1+\exp(-\beta_{ij} - \beta_i - \beta_j)}$$

β_{ij}: dependent on whether i and j are both in the first half of the vertex set, both in the second half, or one in each

β_i, β_j: "popularity" parameter for individual vertices

GLM Residuals Matrices

Given True Probabilities	Given Approximate Probabilities	Estimated Approximate Probabilities
$p_{ij} = \frac{1}{1+\exp(-\beta_{ij} - \beta_i - \beta_j)}$	$p_{ij} = \frac{\exp(\beta_i + \beta_j)}{1+\exp(-\beta_{ij})}$	$P = \begin{bmatrix} \hat{w}_1 & 0 \\ 0 & \hat{w}_2 \end{bmatrix} \begin{bmatrix} 1 & \hat{\alpha} \\ \hat{\alpha} & 1 \end{bmatrix} \begin{bmatrix} \hat{w}_1^T & 0 \\ 0 & \hat{w}_2^T \end{bmatrix}$

Given True Probabilities

- Use the matrix of Bernoulli parameters that generated the observed graph
- Demonstrates performance in an idealized situation
- Dense, possibly full rank expected value

Given Approximate Probabilities

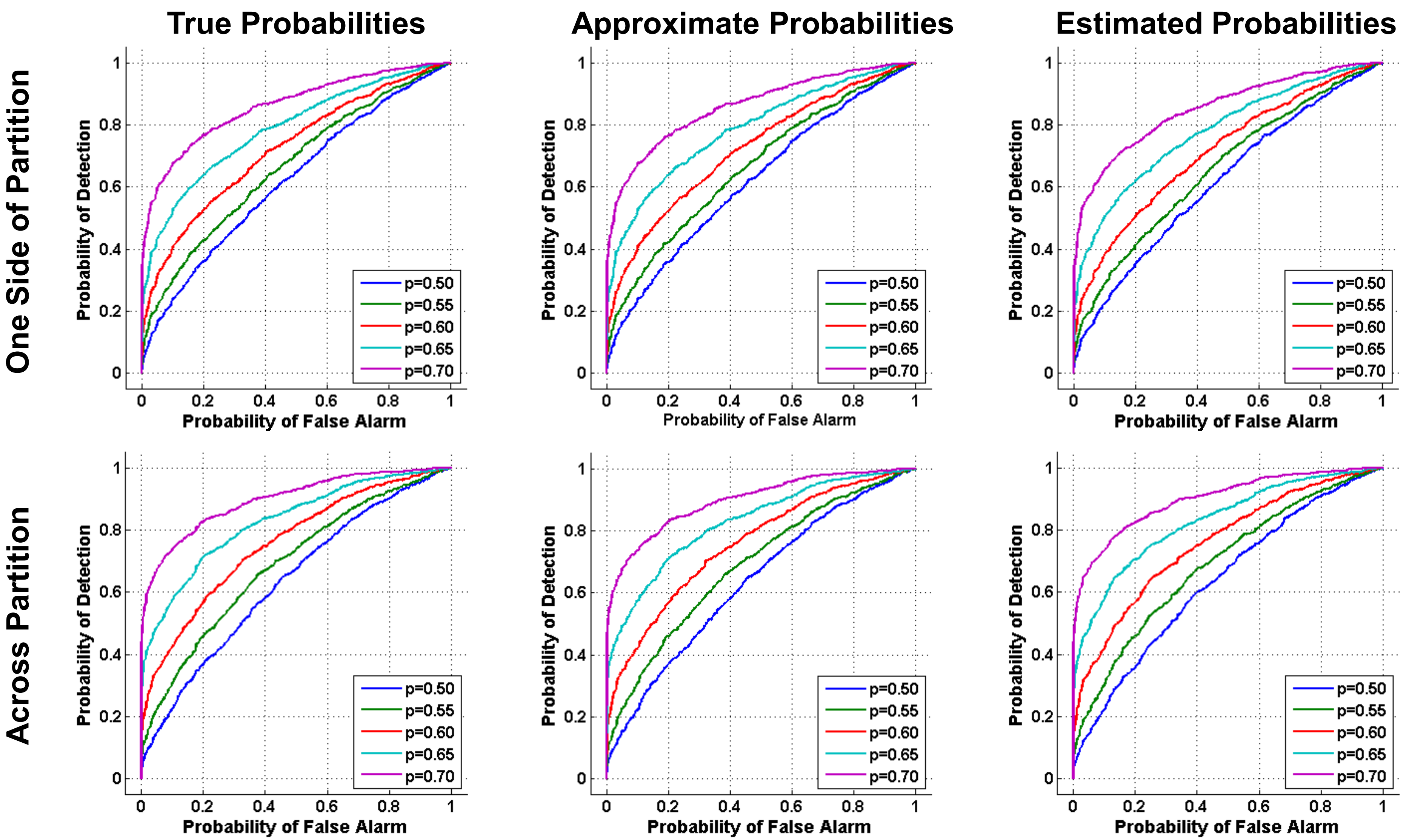
- Approximate probabilities are log-linear in popularity-based parameters
- Demonstrates the impact of using a computationally exploitable model

Estimated Approximate Probabilities

- Probability matrix is estimated using a very simple estimator based on observed densities and degrees
- Demonstrates the loss in performance when not given model parameters

Use different residuals matrices to capture the effects of approximation and estimation

Detection Performance in Simulation

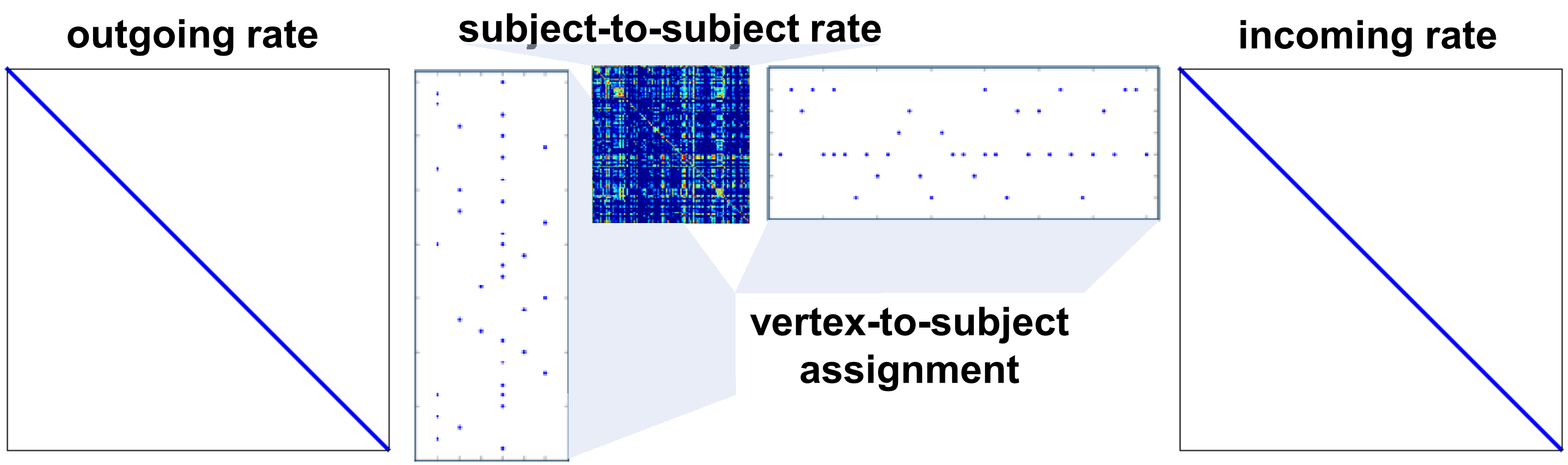


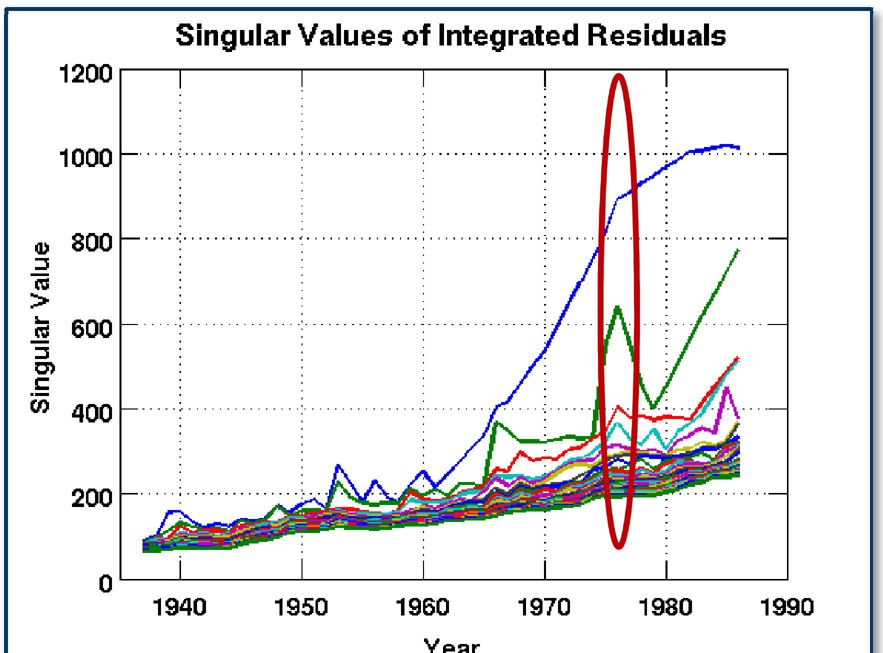
Computationally exploitable model yields nearly the same performance as true model

N. Arcolano and B. A. Miller, "Statistical models and methods for anomaly detection in large graphs," *SIAM Ann. Meeting*, 2012.

Web of Science Data Analysis

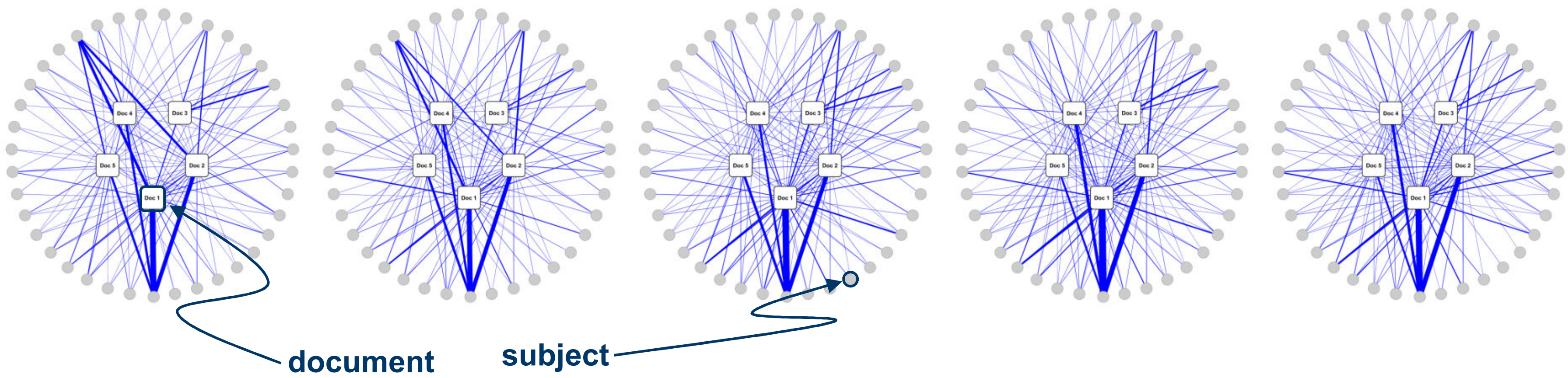
$E[A]=$





- **Model probability of connection as the product of two rate parameters and a parameter based on subjects**
  - **Approximation to logistic regression for small probabilities**
  - **290 subjects, thus, a rank-290 matrix**
- **Compute top 30 singular vectors and values of integrated residuals**
  - **Integrated with a ramp filter (to emphasize emergence) over 6 years**

Emerging Cross-Subject Influence: Citation Graph



- **Approximation of GLM used for residuals in citation graph**
  - **Each vertex has a unique weight**
  - **Ordered pairs of vertices have a weight based on document subject**
- **5 analytical chemistry papers stand out in 1976**
  - **High degree, but not as high as many other documents**
  - **Thousands of citations, some quite recent**
- **Documents stand out over higher-degree vertices due to much higher *cross-subject* citation**

Outlier subgraph demonstrates the impact of using metadata for graph residuals

Summary

- Analytics for very large graphs are a key component of addressing numerous big data challenges
- MIT Lincoln Laboratory has developed an analytic framework for uncued anomaly detection in graphs
  - Based on analysis of graph residuals
- Several new approaches for modeling data in this framework were investigated under the current effort, all informed by real, diverse datasets
  - Preferential attachment with memory
  - Moving average adjacency filter
  - Generalized linear model for attribute-based modeling
- Demonstrated computation on a 1-billion vertex graph using a commodity computing cluster
- Approximation for GLM enables detection of subsets with anomalously high cross-subject citation
- Ongoing work includes incorporating data corruption and obfuscation mechanisms into the modeling process