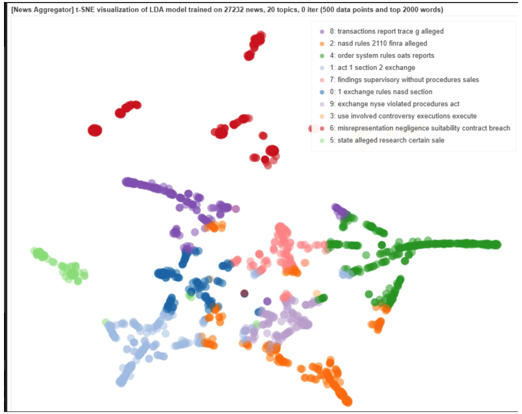
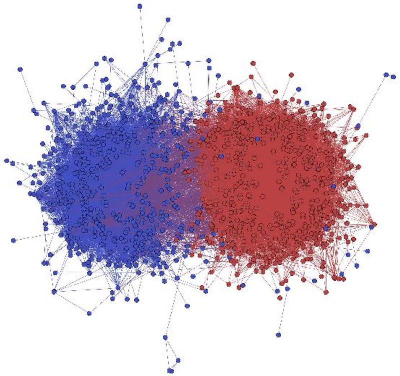
Social Networks and Image Recognition

By: Daniel Kraft and Steven Stuglia



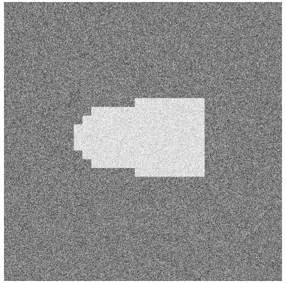
Practical Uses of Community Detection



Our Data

```
First line: n
Second line: W(1,1)  W(1,2)  W(1,3)  ...  W(1,n)
Third line:  W(2,1)  W(2,2)  W(2,3)  ...  W(2,n)
...
Line n+1: W(n,1)  W(n,2)  W(n,3)  ...  W(n,n)
```

where the entries $W(i,j)$ were computed based on the reference Phantom image. Specifically $W(i,j) = exp(-|I(i) - I(j)|/20 - \|i-j\|^2/10)$ where $I(i), I(j)$ are noisy intensities of pixels i and j of image Phantom*.bmp.



Erdos-Renyi Random Graph Model Testing

$$p_{MLE} = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}.$$

Estimated p Value
0.2768

$$\mathbb{E}[X_q | X_2 = m] \approx \binom{n}{q} \left(\frac{2m}{n(n-1)} \right)^{q(q-1)/2}$$

Estimated 3-Cliques	Actual 3-Cliques
3,786,100	9,552,971

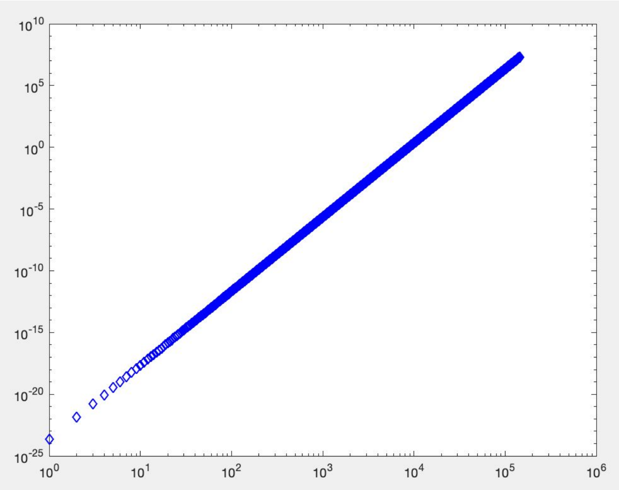
Estimated 4-Cliques	Actual 4-Cliques
20,505,000	417,952,847

$$X_3 = \frac{1}{6} trace\{A(A^2 - D)\} = \frac{1}{6} trace(A^3).$$

Erdos Renyi Random Graph 4-Clique Prediction

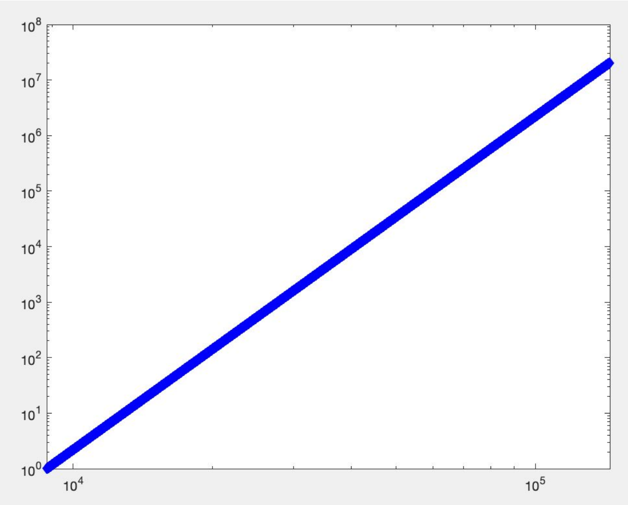
$$log(X_4) = a_{ER}log(m) + b_{ER}$$

Log-Log Plot (With All Edges)



$a_{ER} = 6.0$ $b_{ER} = -54.471$

Log-Log Plot (With Edges : $X_4 \geq 1$)



$a_{ER} = 6.0$ $b_{ER} = -54.471$

Stochastic Block Graph Model Testing

$$a_{MM} = \frac{1}{2}\left(c_1 + \sqrt[3]{2c_2 - c_1^3}\right) \quad , \quad b_{MM} = \frac{1}{2}\left(c_1 - \sqrt[3]{2c_2 - c_1^3}\right)$$

$$c_1 = \frac{4m}{n(n-1)} \qquad c_2 = \frac{24t}{n(n-1)(n-2)}.$$

m = # edges   n = # nodes   t = # 3 cliques = (1/6)trace(A^3)

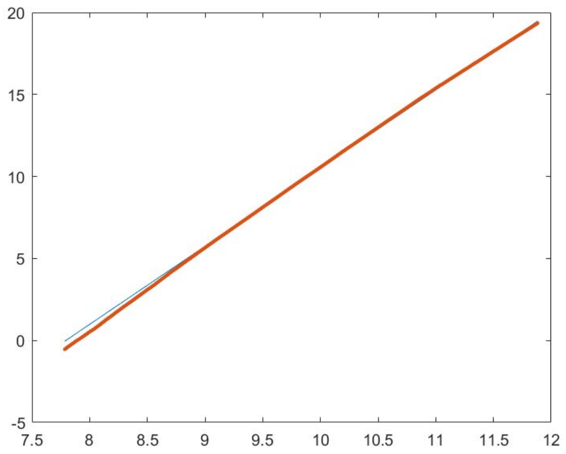
a = 0.5954   b = 0.0417

Expected $X_4 = 2.5325 \times 10^8$ Actual $X_4 = 4.1795 \times 10^8$.

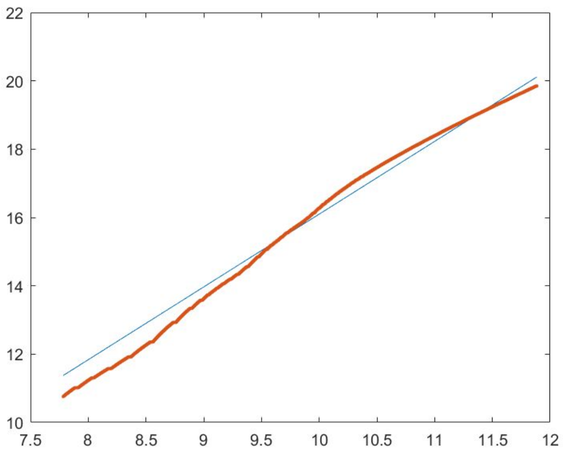
Stochastic Block Model Graph 4-Clique Prediction

$$log(X_4) = a_{SSBM} log(m) + b_{SSBM}$$

Log-Log Plot of Estimated 4-Cliques



Log-Log Plot of 4-Cliques



a_{SSBM} =4.7631 b_{SSBM} = -37.1215

a_0 = 2.130547 b_0 = -5.2078

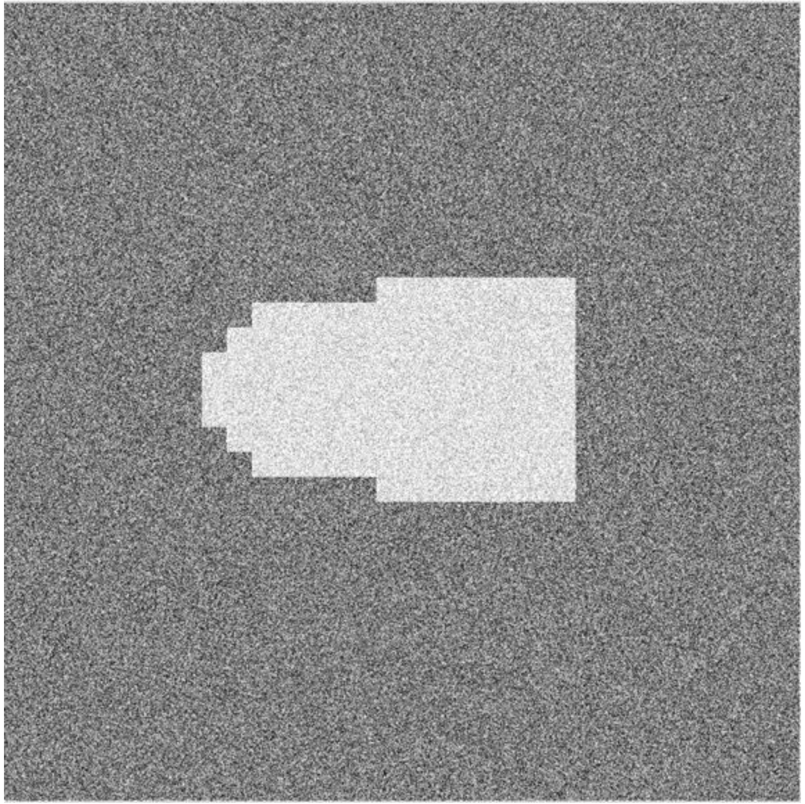
Spectral and SDP Partitions

$$\Delta = D - A \ , \ \ \Delta_{ij} = \begin{cases} d_i & \textit{if} \quad i = j \\ -1 & \textit{if} \quad (i,j) \in \mathcal{E} \\ 0 & \quad \textit{otherwise} \end{cases}$$

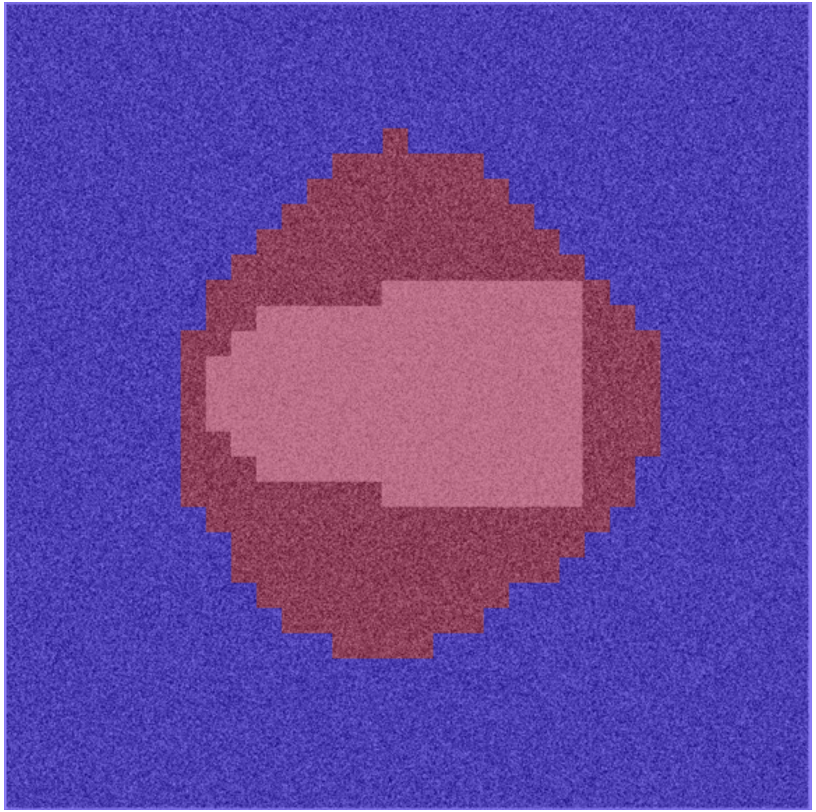
$$L = D^{-1}\Delta \ , \ \ L_{i,j} = \begin{cases} 1 & \textit{if} \quad i = j \text{ and } d_i > 0 \\ -\frac{1}{d(i)} & \textit{if} \quad (i,j) \in \mathcal{E} \\ 0 & \quad \textit{otherwise} \end{cases}$$

$$\tilde{\Delta} = D^{-1/2}\Delta D^{-1/2} \ , \ \ \tilde{\Delta}_{i,j} = \begin{cases} 1 & \textit{if} \quad i = j \text{ and } d_i > 0 \\ -\frac{1}{\sqrt{d(i)d(j)}} & \textit{if} \quad (i,j) \in \mathcal{E} \\ 0 & \quad \textit{otherwise} \end{cases}$$

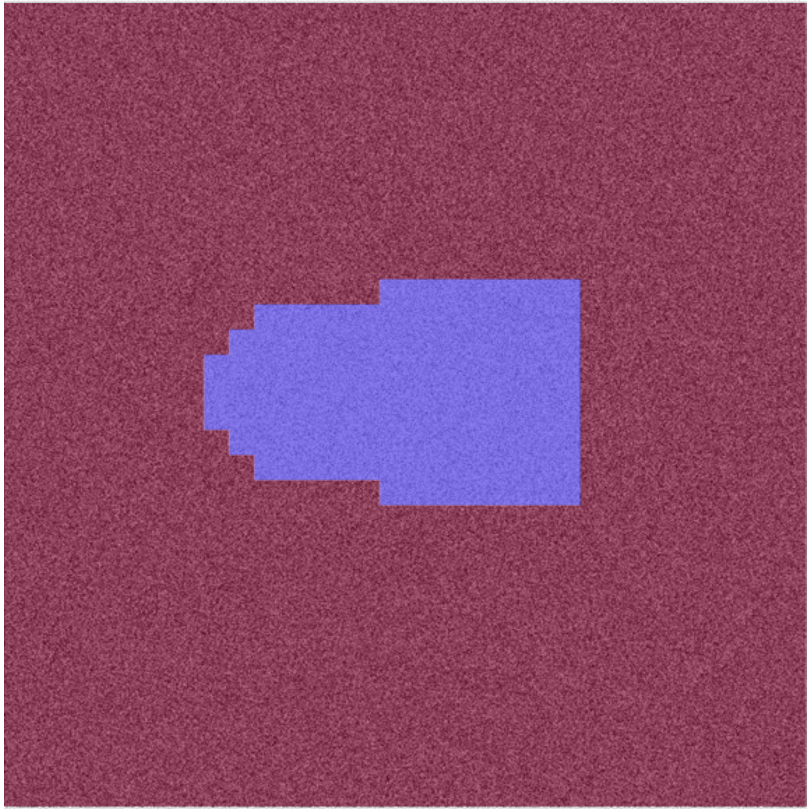
High Resolution Noisy Phantom



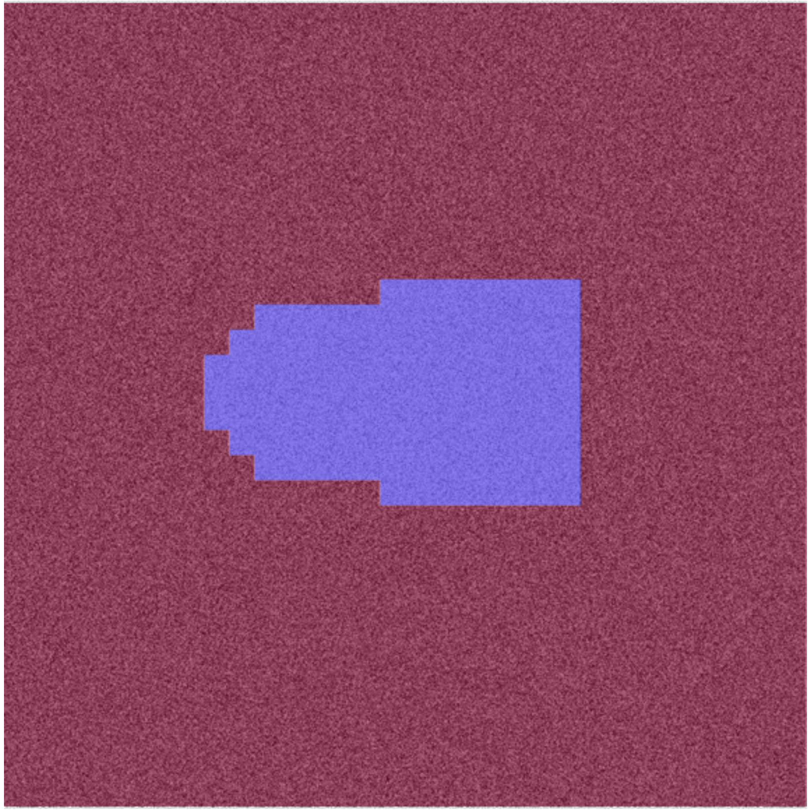
Spectral Algorithm using the Weight Matrix



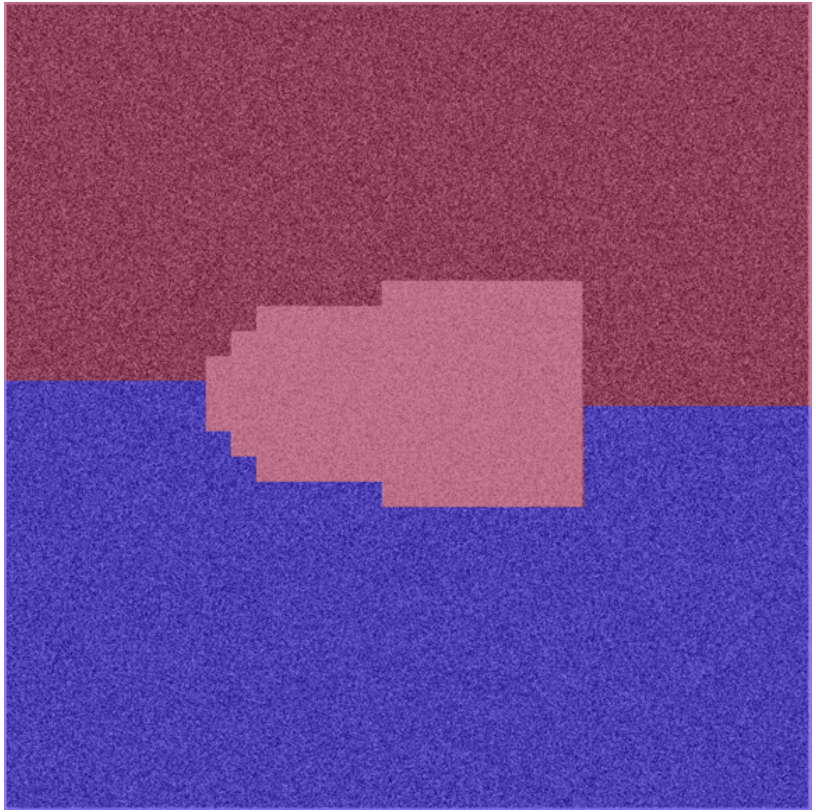
Spectral Algorithm using the Graph Laplacian



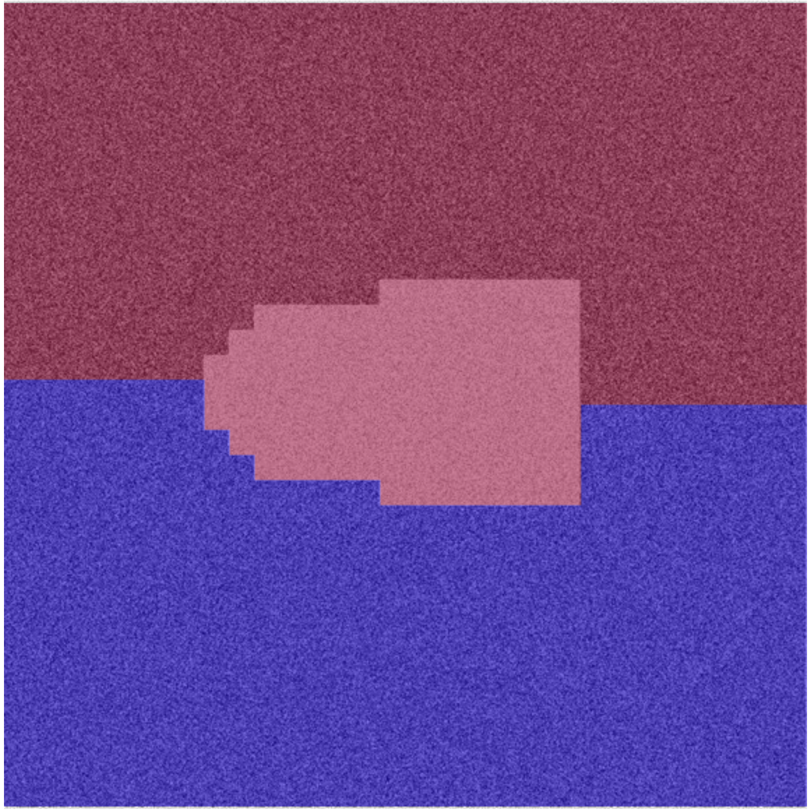
Spectral Algorithm using the Symmetric Normalized Weighted Graph Laplacian



Weight Matrix SDP



The Graph Laplacian SDP



Weighted Graph Laplacian SDP