Network Inverse Problems and High-dimensional Statistics

Sujay Sanghavi

Electrical and Computer Engg.

University of Texas, Austin

Joint w/ Y. Chen, and H. Xu

Network Inverse Problems

e.g. Scheduling & routing, consensus/gossip, mixing, epidemics and rumor mongering, peer-to-peer …

Forward Problems

Generative assumptions about networks

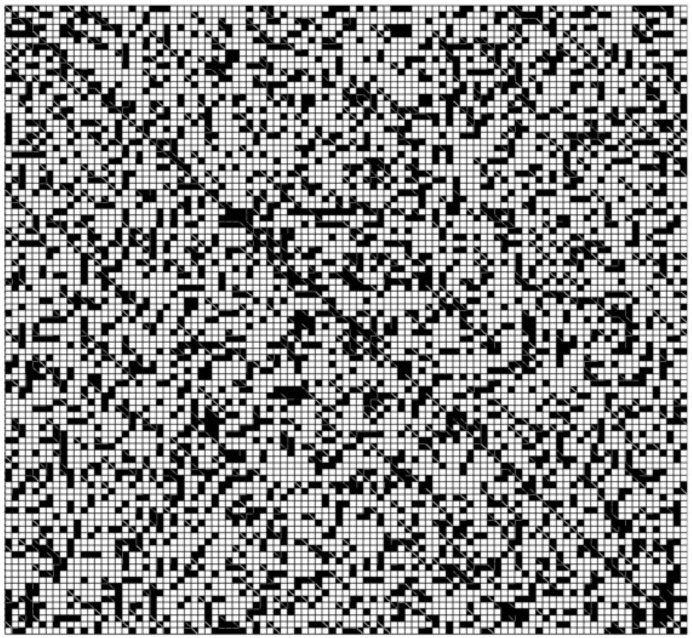
Network structure, node behavior

Inverse Problems

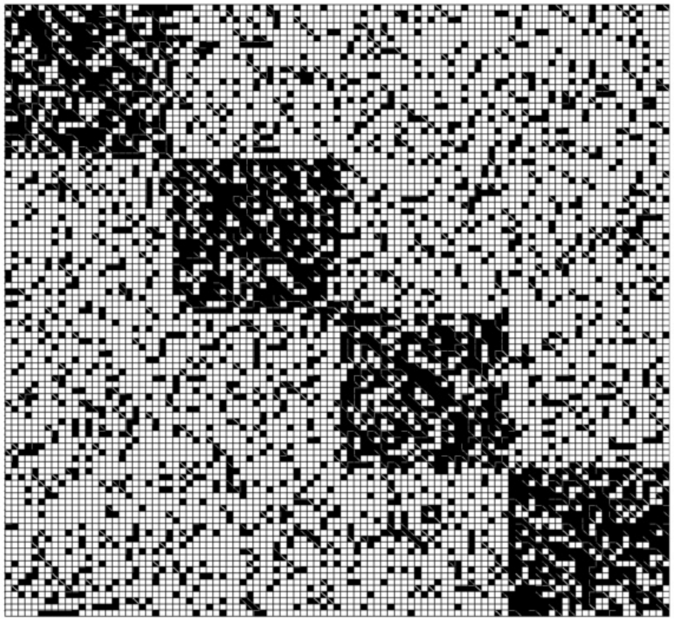
e.g. finding the network graph, finding influential nodes, …

Especially: **network implications of high-dimensional statistics**

Graph Clustering



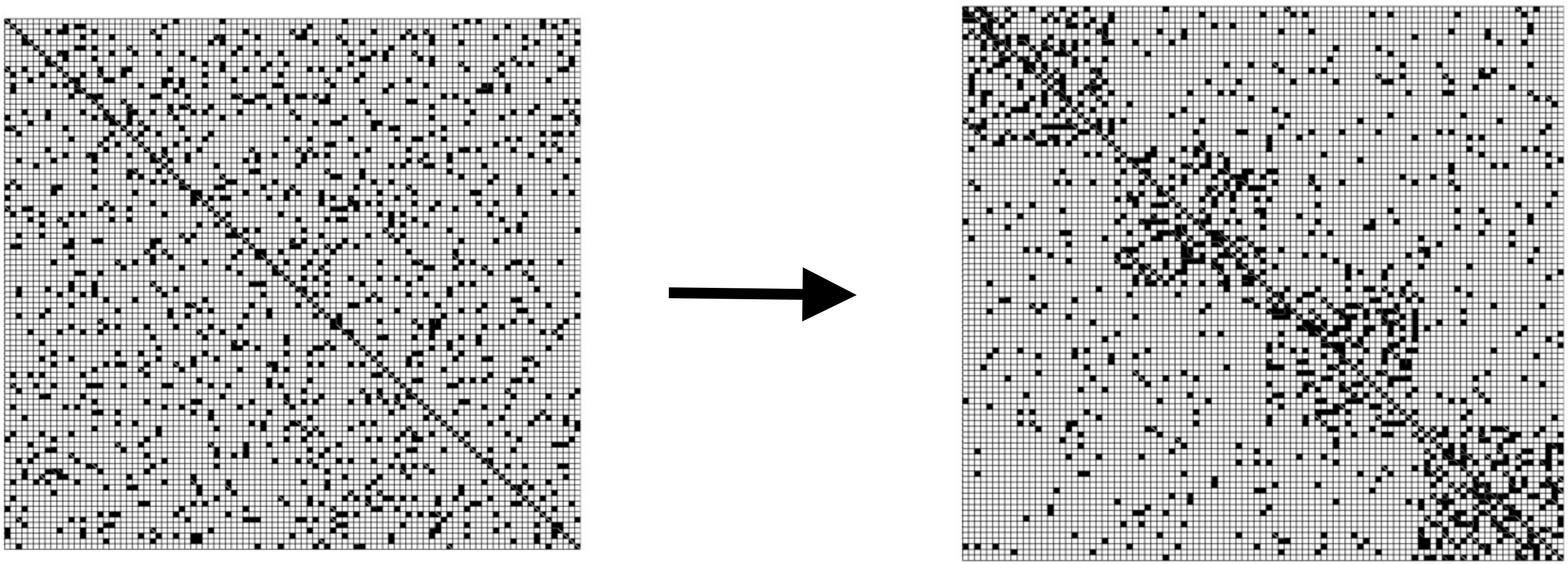
Given a graph



Partition the nodes so that there is higher density within partitions, and lower across

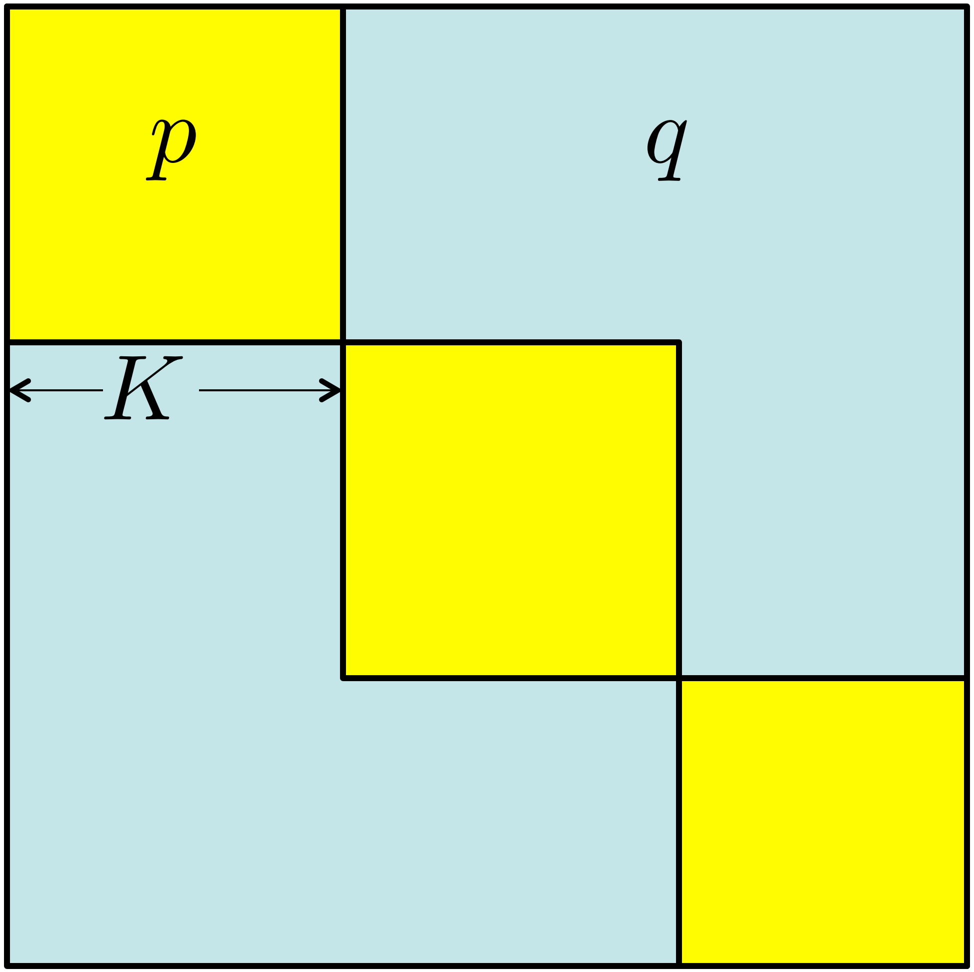
Applications: Community detection, recommendations, identifying bottlenecks / vulnerabilities …

Sparse Graph Clustering



Sparsity makes the problem harder
(because “SNR” is lower)

Planted Partition / Stochastic Block Model



A classic model for random graphs with clustering

Using an underlying partition of the nodes, make a random graph

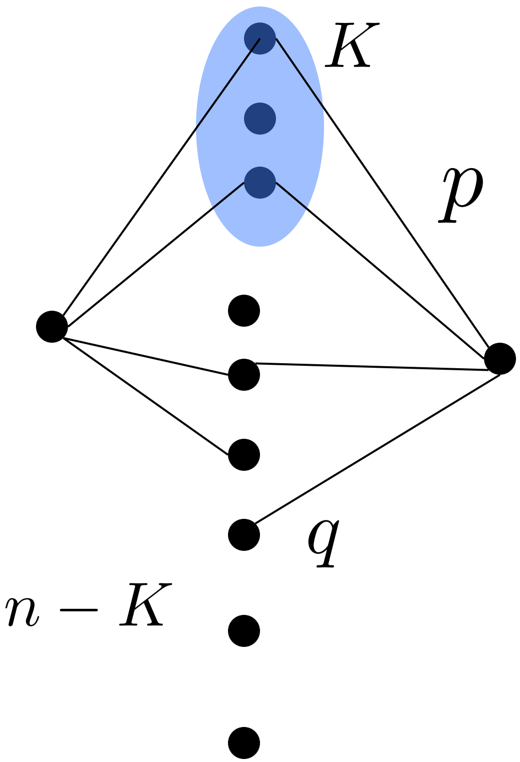
Clustering Task: given the graph, find the underlying Partition *(upto every last node)*

Quantities governing the difficulty: p, q, K

Min cluster size

Some intuition ...

SLINK: i, j in same cluster $\Leftrightarrow N(i) \cap N(j) > \tau$



For two nodes in the same partition

$$\mathbb{E}[N(i) \cap N(j)] = Kp^2 + (n-K)q^2$$

$$\mathbb{V}ar[N(i) \cap N(j)] \approx Kp^2 + (n-K)q^2$$

For two nodes in different partitions

$$\mathbb{E}[N(i) \cap N(j)] = 2Kpq + (n-2K)q^2$$

Some intuition ...

$$\mathbb{E}[\text{same} - \text{different}] \; = \; K(p-q)^2$$

Assuming $K << n$

And $p \approx q$

$$\mathbb{V}ar[same] \; \approx \; Kp^2 + (n-K)q^2$$

$$\approx \; np^2$$

For there to exist a threshold with a high likelihood of success, need

$$K(p-q)^2 \; \gtrsim \; \sqrt{n}p$$

i.e.

$$\frac{(p-q)^2}{p} \; \gtrsim \; \frac{\sqrt{n}}{K}$$

Some intuition ...

$$\mathbb{E}[\mathrm{same} - \mathrm{different}] \ = \ K(p-q)^2$$

Assuming $K << n$

And $p \approx q$

$$\mathbb{V}ar[same] \ \approx \ Kp^2 + (n-K)q^2$$

$$\approx \ np^2$$

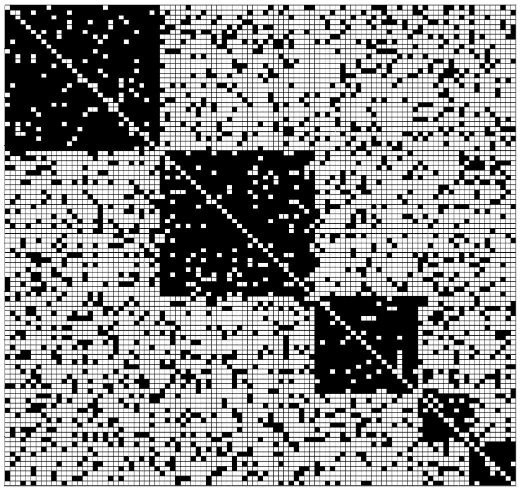
For there to exist a threshold with a high likelihood of success, need

$$K(p-q)^2 \ \gtrsim \ \sqrt{n}p$$

i.e.

$$\frac{(p-q)^2}{p} \ \gtrsim \ \frac{\sqrt{n}}{K}$$

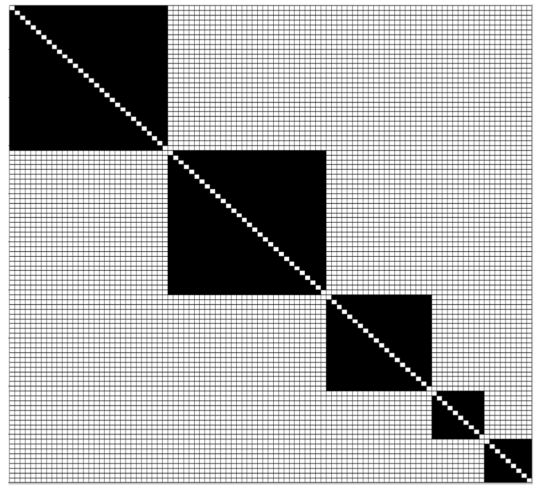
A classic observation



Given adjacency matrix

A

→

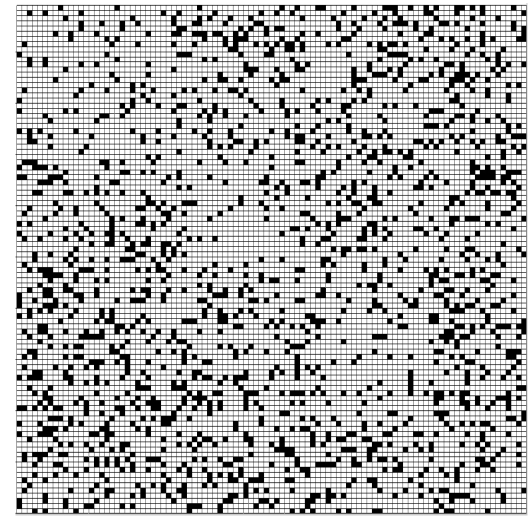


cluster matrix

Y

Note: low-rank

+



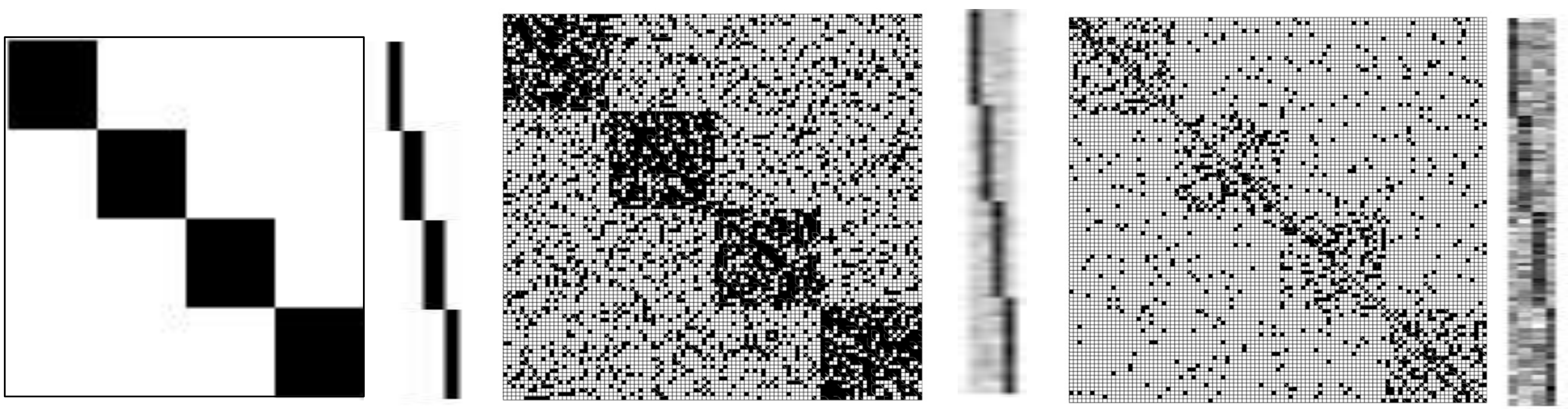
Perturbation/Error matrix

S

“Generic” Spectral Algorithm

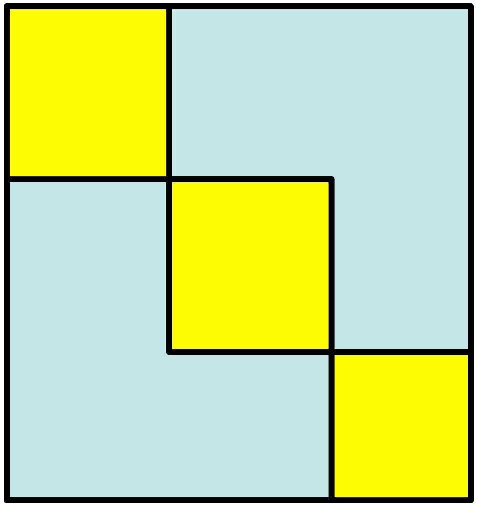
(1) Find top r eigenvectors of A via SVD

(2) Represent each node as a point in eigenvalue space, and do “simple, local” clustering + rounding



The eigenspace becomes noisier as graph parameters become harder

The Spectral SNR



Leading eigenvector is $\mathbf{1}$, so lets center the matrix

$$\widehat{a}_{ij} = a_{ij} - (q + K(p-q)/n)$$

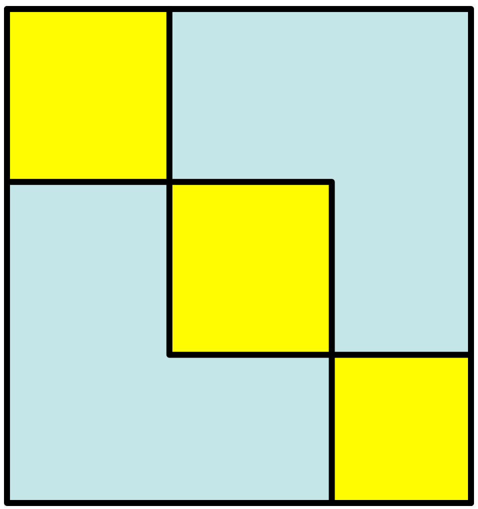
"Signal" = $E\left[\frac{1}{K}\mathbf{1}_C' \widehat{A}\mathbf{1}_C\right]$

$$= K(p-q)(1-K/n) \approx K(p-q)$$

"Noise" = largest eigenvalue of "iid" random matrix where every element has variance $\Theta(p)$

$$\approx \sqrt{np}$$

The Spectral SNR



Leading eigenvector is $\mathbf{1}$, so lets center the matrix

$$\widehat{a}_{ij} = a_{ij} - (q + K(p-q)/n)$$

"Signal" = $E\left[\frac{1}{K}\mathbf{1}_C' \widehat{A}\mathbf{1}_C\right]$

$$= K(p-q)(1-K/n) \approx K(p-q)$$

"Noise" = largest eigenvalue of "iid" random matrix where every element has variance $\Theta(p)$

$\approx \sqrt{np}$

So, spectral algorithms need $\frac{p-q}{\sqrt{p}} \gtrsim \frac{\sqrt{n}}{K}$ vs $\frac{(p-q)^2}{p} \gtrsim \frac{\sqrt{n}}{K}$

However, no spectral algorithm has been demonstrated to achieve this.
Main result of our paper: we do !

Existing Work in this Model

Paper	Min. cluster size K	Density difference $p-q$
Boppana (1987)	$n/2$	$\frac{\sqrt{p}}{\sqrt{n}}$
Jerrum & Sorkin (1998)	$n/2$	$\frac{1}{n^{1/6-\epsilon}}$
Condon & Karp (2001)	n	$\frac{1}{n^{1/2-\epsilon}}$
Carson & Impaglizzo (2001)	$n/2$	$\frac{\sqrt{p}}{\sqrt{n}}$
Feige & Kilian (2001)	$n/2$	$\frac{1}{n})$
McSherry (2001)	$n^{2/3}$	$\sqrt{\frac{pn^2}{K^3}}$
Bollobas (2004)	n	$\max\{\sqrt{\frac{q}{n}}, \frac{1}{n}\}$
Giesen & Mitsche (2005)	$\sqrt{n}$	$\frac{\sqrt{n}}{K}$
Shamir (2007)	$\sqrt{n}$	$\frac{\sqrt{n}}{K}$
Rohe et al (2010)	$n^{3/4}$	$\frac{n^{3/4}}{K}$
Oymak & Hassibi (2011)	$\sqrt{n}$	$\max\{\frac{\sqrt{n}}{K}, \sqrt{\frac{1}{K}}\}$
Sussman et al (2011)	$n^{3/4}$	
Our result	$\sqrt{n}$	$\frac{\sqrt{pn}}{K}$

Maximum Likelihood

$$\max_{Y} \quad \log \Pr(A|Y)$$

(assume for now p, q known)

Y is a cluster matrix

$y_{ij} = 1 \Leftrightarrow i, j$ in same cluster

Simplifying …

$$\Pr(A|Y) = \prod_{(i,j):y_{ij}=1} p^{a_{ij}}(1-p)^{1-a_{ij}} \prod_{(i,j):y_{ij}=0} q^{a_{ij}}(1-q)^{1-a_{ij}}$$

In-cluster edges

Across-cluster edges

Key Step

Rewriting the maximum likelihood via regrouping terms

$$\Pr(A|Y) = \prod_{(i,j):y_{ij}=1} p^{a_{ij}}(1-p)^{1-a_{ij}} \prod_{(i,j):y_{ij}=0} q^{a_{ij}}(1-q)^{1-a_{ij}}$$

$$= \prod_{(i,j):a_{ij}=1} p^{y_{ij}} q^{1-y_{ij}} \prod_{(i,j):a_{ij}=0} (1-p)^{y_{ij}}(1-q)^{1-y_{ij}}$$

$$\equiv \prod_{(i,j):a_{ij}=1} \left(\frac{p}{q}\right)^{y_{ij}} \prod_{(i,j):a_{ij}=0} \left(\frac{1-p}{1-q}\right)^{y_{ij}}$$

Optimization problem (still combinatorial)

Thus maximum (log) likelihood becomes

$$\max_{Y} \quad c_1 \left(\sum_{a_{ij}=1} y_{ij} \right) - c_2 \left(\sum_{a_{ij}=0} y_{ij} \right)$$

Y is a cluster matrix

Where

$$c_1 = \log \frac{p}{q} \qquad c_2 = \log \frac{1-q}{1-p}$$

Our Algorithm

Replacing the “cluster constraint” with a penalty, and relaxing integrality

$$\max_{Y} \quad c_1 \left(\sum_{a_{ij}=1} y_{ij} \right) - c_2 \left(\sum_{a_{ij}=0} y_{ij} \right) - \lambda \|Y\|_*$$

Nuclear/trace Norm
= sum of singular values

Convex !

$$0 \leq y_{ij} \leq 1$$

Note: cluster matrices also satisfy $Y \succeq 0$. However, adding this

(a) Makes the convex program harder to solve

(b)< and We do not know how to use this to get better performance results ..>

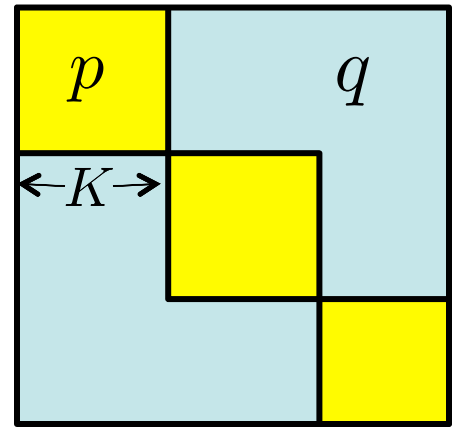
Performance Analysis

Under what conditions on p, q, K

will the (unrounded, un post-processed)

optimum of the convex program recover

the true cluster matrix **exactly** ?



Parameters: linearize using $\log x \approx x - 1$

$$c_1 = \log \frac{p}{q} \approx \frac{p-q}{q}$$

$$c_2 = \log \frac{1-q}{1-p} \approx \frac{p-q}{1-p}$$

Also: $\lambda = 48\sqrt{n \log n}$

Main Result

Theorem:

The true cluster matrix is the unique optimum of our convex program, provided

$$p - q \;\geq\; \alpha \, \frac{\sqrt{p(1-q)n}}{K} \, \log^2 n$$

In the paper: a way to estimate p, q from the graph itself ….

… and an overall theorem guaranteeing that using estimated parameters also works

Remarks

- If $K \in \Theta(n)$ then algorithm can cluster even when $p, q \approx \frac{\log^4 n}{n}$

  - close to the connectivity threshold, matches previous results

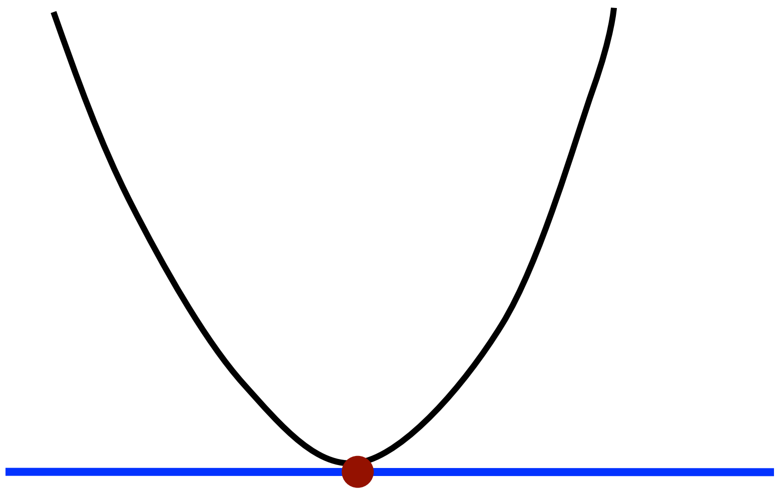
- If $K \in \Omega(\sqrt{n}\log^2 n)$, our method works with $p - q \in \Theta\left(\frac{n \log^4 n}{K^2}\right)$

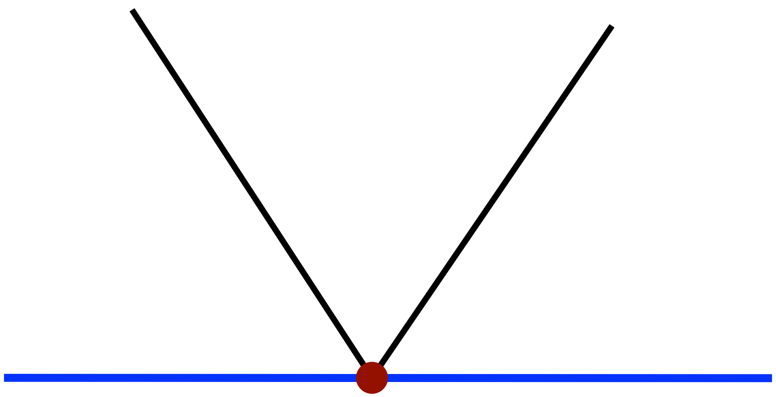
  - previous best result needed $p - q \in \Theta\left(\frac{n^2}{K^3}\right)$

- Ours is the first result on weighted sparse + low-rank (in any setting)

  - shows order-wise better performance than unweighted.

Proof Technique





A point x is the optimum of a convex function f $\Longleftrightarrow$ Zero lies in the (sub) gradient $\partial f(x)$ of f at x

Proof Technique

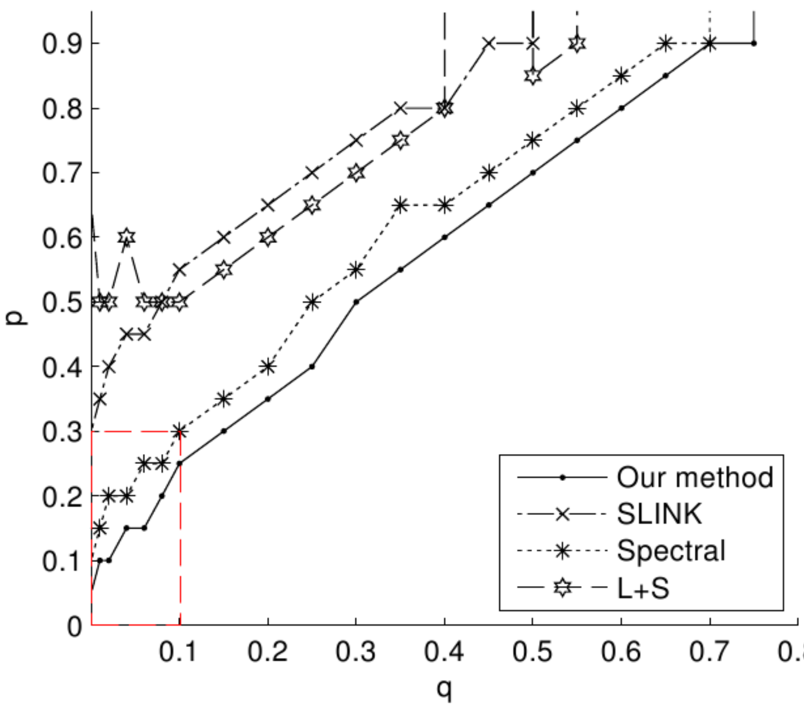
So, idea 0: show that, w.h.p., $0 \in \partial f(Y^*)$

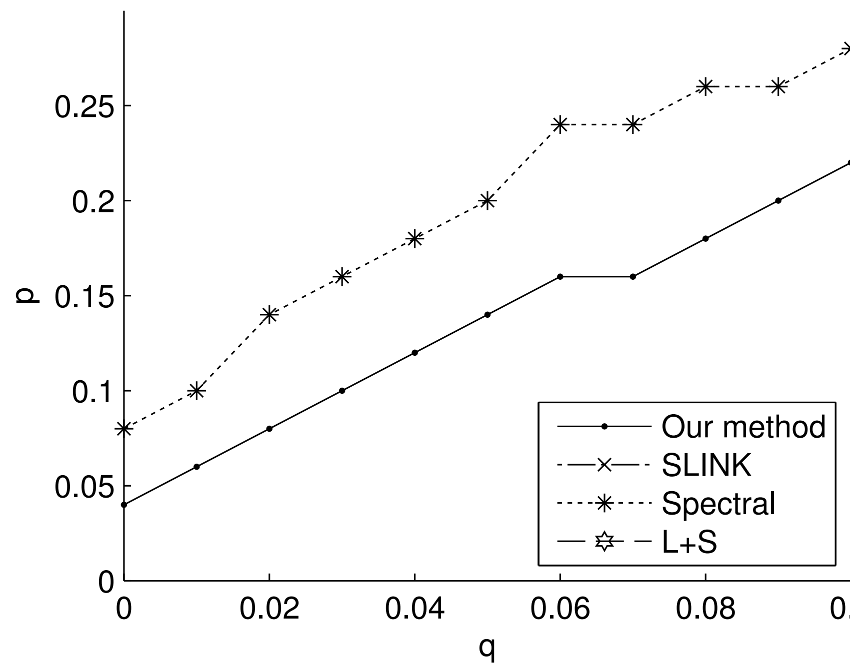
$$\text{when} \quad p - q \;\geq\; \alpha \frac{\sqrt{p(1-q)n}}{K} \log^2 n$$

This is hard to do !

Our approach: make a new, different **sufficient** (not necessary) condition for optimality

Empirical Performance

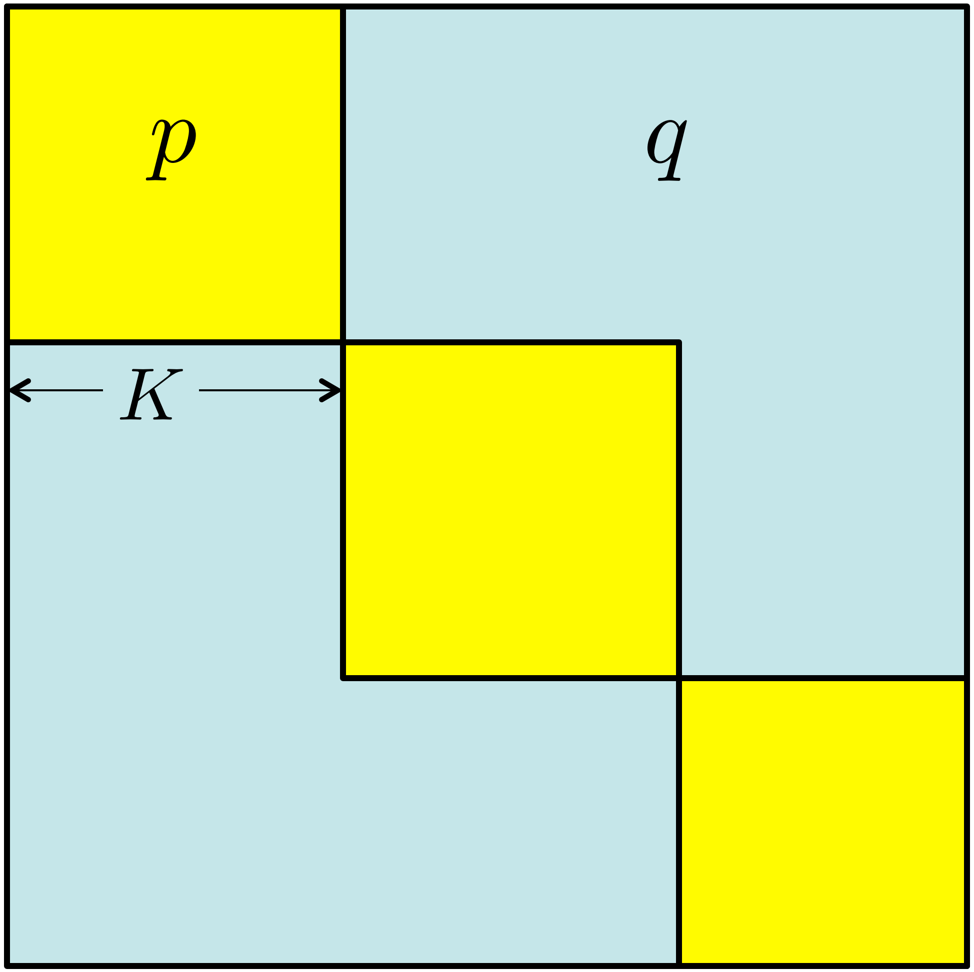




$$n = 1000$$

$$K = 200$$

Algorithm: Estimating Parameters



$\mathbb{E}[A]$ has following e-values:

$$\lambda_1 = K(p-q) + nq$$

Extensions

Lemma: (monotonicity)

Consider a realization A , and let $\widehat{Y}$ be the optimum of the algorithm.

Then, consider an **arbitrary** perturbation $\widetilde{A}$ of A, obtained as follows:

(a) Choose some pairs i, j for which $a_{ij} = 0$ but $\widehat{y}_{ij} = 1$,

and set $\widetilde{a}_{ij} = 1$

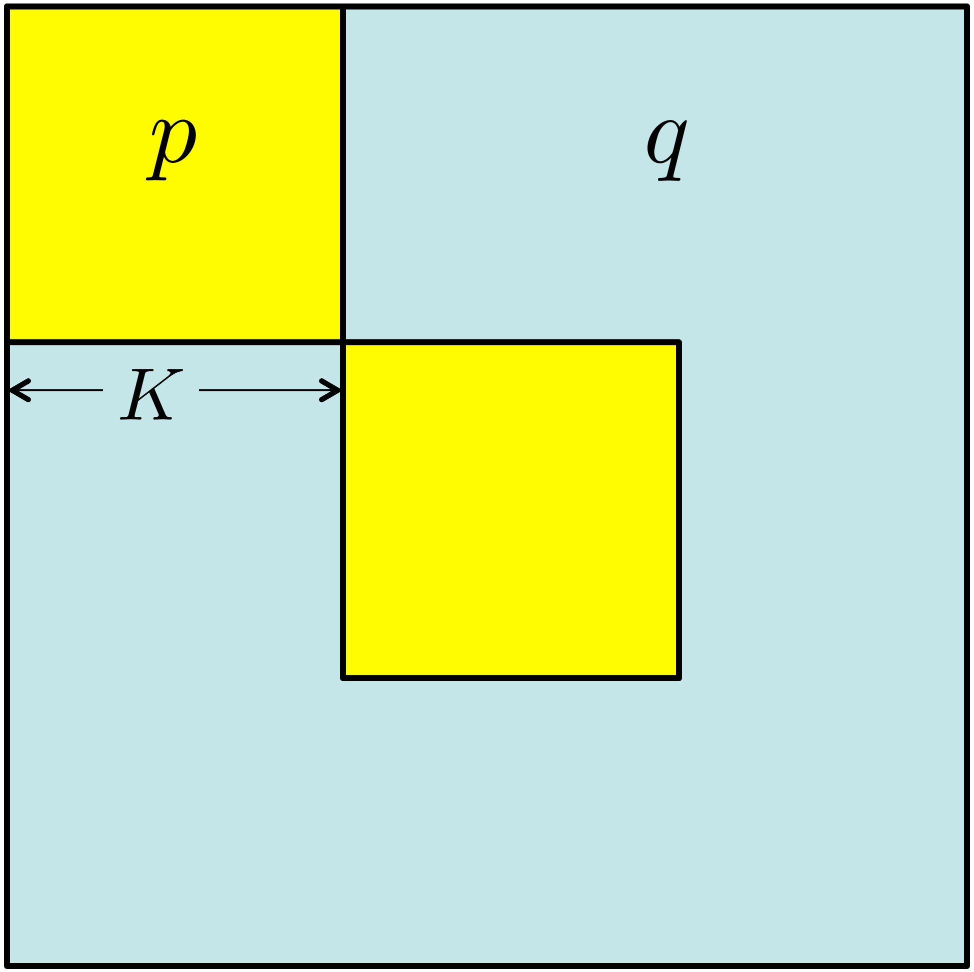
(b) Choose some pairs i, j for which $a_{ij} = 1$ but $\widehat{y}_{ij} = 0$,

and set $\widetilde{a}_{ij} = 0$

Then, if the algorithm is run with $\widetilde{A}$, the optimum will still be $\widehat{Y}$

Direct implication: Heterogenous edge probabilities allowed

Extensions



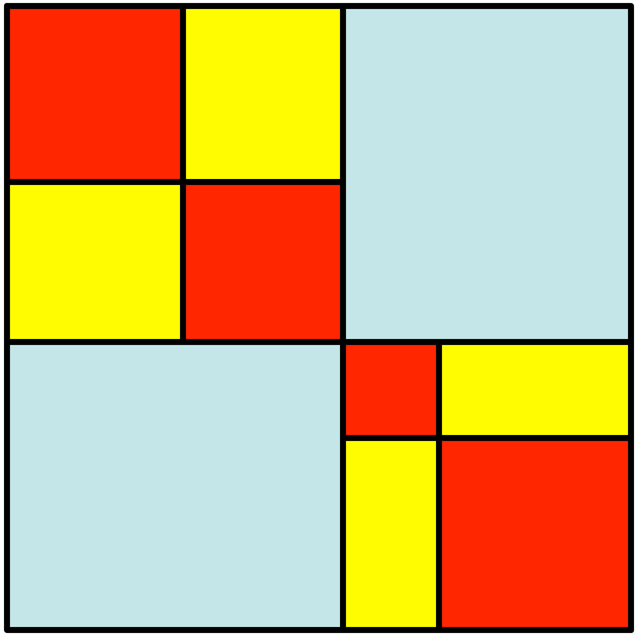
Outliers:

Nodes that are not part of any cluster.

If every edge out of such a node has probability upper bounded by q,

Then algorithm will still find the clusters.

Implications: Hierarchical Clustering



If we run algorithm with

p = lower bound on top-level cluster's probability

q = upper bound on every other level's probability

then will find all top level clusters.

… and can repeat hierarchically.

Summary

- **New algorithm** for clustering sparse graphs
  - maximum-likelihood, with regularization replacing combinatorial
  - convex program, with fast specialized algorithms
- **Beats all previous performance bounds**
  Close to “fundamental spectral limit” (?)
- Extends to hierarchical clustering
- Similar results can be shown for dense graph clustering, planted coloring etc.

Open problem:

Lower bounds – none known for case of more than two clusters.

Thanks + Questions