Computational and Statistical Tradeoffs in Biclustering

Sivaraman Balakrishnan

Joint work with

Mladen Kolar

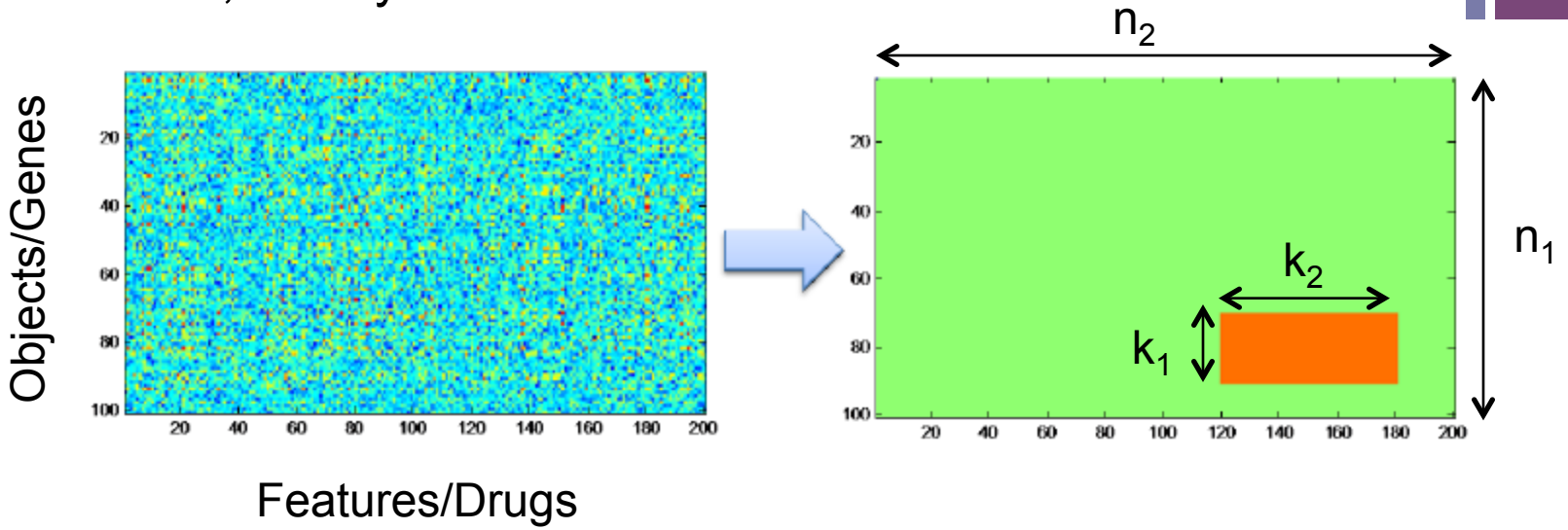
Alessandro Rinaldo

Aarti Singh

Larry Wasserman

Problem Setup

Small clusters, mostly irrelevant features



Single cluster model:

W = A + R

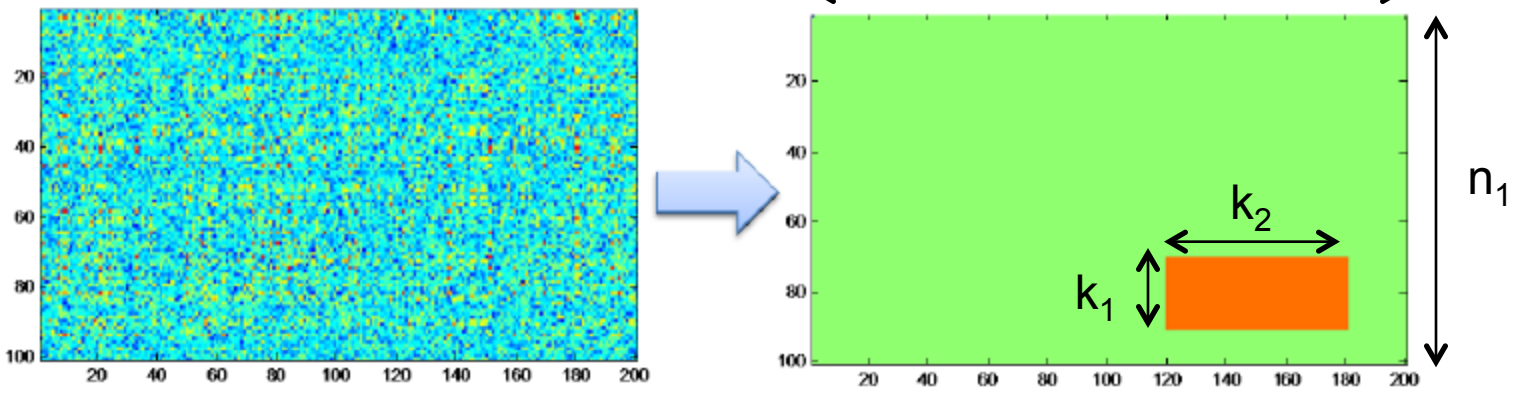
A has a single block of activation of strength μ

$\mathbf{R} \sim_{\text{i.i.d}} \mathcal{N}(0, \sigma^2)$

Problem Setup

Small clusters, mostly irrelevant features

- Objects/Genes



1. $n_1 = n_2 \equiv n, k_1 = k_2 \equiv k$

2. Activation μ constant across bicluster

3. k and σ known. w.l.o.g. $\sigma = 1$

Related work

The many normal means problem

- Given a vector Z in $\mathbb{R}^n$

$$Z_i \sim \theta_i + \sigma\epsilon_i$$
$$\epsilon_i \sim_{\text{i.i.d}} \mathcal{N}(0,1)$$

- θ is a k-sparse vector, with smallest non-zero entry μ
- **Detection**: Smallest μ as a function of (n, k, σ) such that θ is still distinguishable from the all zeros vector
- **Localization**: Smallest μ such that we can still find non-zero θ_i w.h.p

Related work

The many normal means problem

- Given a vector Z in $\mathbb{R}^n$

$$Z_i \sim \theta_i + \sigma\epsilon_i$$
$$\epsilon_i \sim_{\text{i.i.d}} \mathcal{N}(0,1)$$

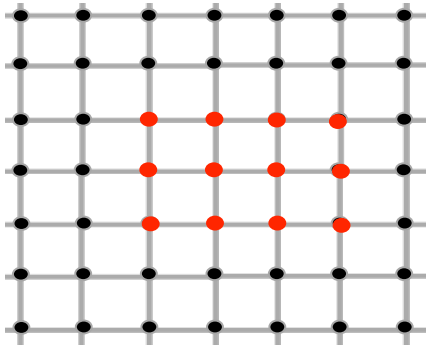
- θ is a k-sparse vector, with smallest non-zero entry μ

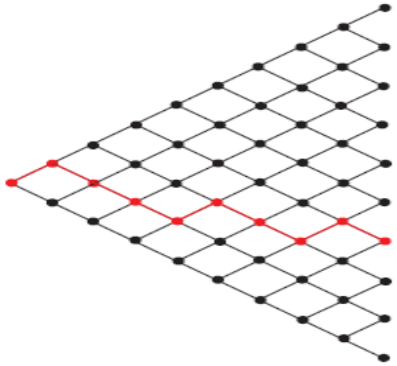
- The many normal means problem is a popular testbed for statisticians
  - Strong connections to nonparametric (orthogonal series) regression
  - Shrinkage estimation
  - Multiple hypothesis testing

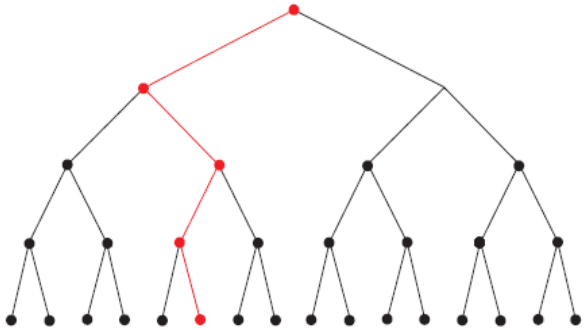
Structured normal means problems

Lattice topology (Arbitrary cluster of activation)
[Durand, Arias-Castro, Candes '09]

Lattice and tree topology (Path of activation)
[Arias-Castro, Candes, Helgason, Zeitouni'08]







Most relevant -- Complete graph (Activation pattern is spanning tree, perfect matching)
[A.-Berry, Broutin, Devroye, Lugosi'09]

Related Work

Structured normal means problems

Many of these consider fairly general problems but…

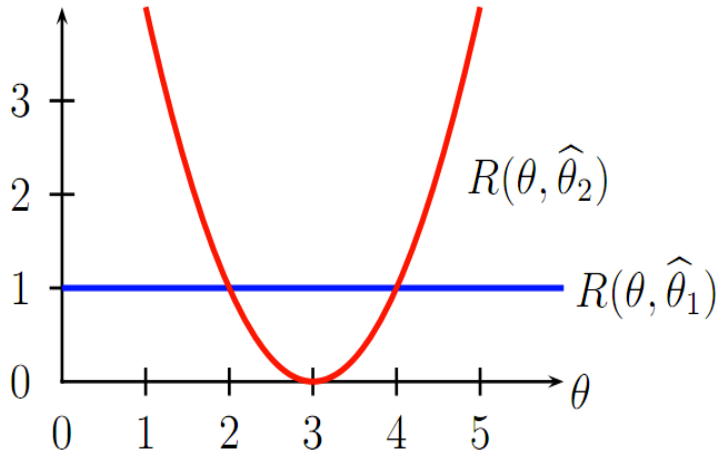
- Focus on detection
- Ignore computation !!
  - Often involve computationally inefficient estimators

Minimax analysis

- Approach the problem via information theoretic limits
- Classical problem in statistics: How to compare two estimators?
- Define: The risk of an estimator $\hat{\theta}$

$$R(\theta, \hat{\theta}) = \mathbb{E}(L(\theta, \hat{\theta}))$$

  - Example: classification risk - probability of misclassifying a new randomly drawn point



Minimax analysis (ii)

- Minimax estimator
  - An estimator δ that satisfies

$$\sup_{\theta} \mathbb{E}(L(\theta, \delta)) = \inf_{\hat{\theta}} \sup_{\theta} \mathbb{E}(L(\theta, \hat{\theta}))$$

- Ignore constants (*rate* minimax estimators)

$$\sup_{\theta} \mathbb{E}(L(\theta, \delta)) \asymp \inf_{\hat{\theta}} \sup_{\theta} \mathbb{E}(L(\theta, \hat{\theta}))$$

Minimax lower bound for biclustering

- Theorem: There exists a constant c, such that if

$$\mu \leq c\sqrt{\frac{\log n}{k}}$$

"Structure" gain

the success probability of any procedure remains bounded away from 1 as (n,k) grow.

- Proof is an application of Fano's lemma with 0/1 loss
- Compare this to $c\sqrt{\log n}$ lower bound for many normal means problem

Combinatorial upper bound

Establishing tightness of lower bound and the minimax rate

Theorem: For C large enough, if

$$\mu \geq C\sqrt{\frac{\log n}{k}}$$

the largest average $(k \times k)$ sub-matrix recovers the true bicluster.

- Establishes minimax rate (matches lower bound up to constants)
- Appears computationally difficult to find largest average sub-matrix (search over all submatrices of size (k x k)?)

Tractable algorithms

Algorithm	Rates
Thresholding	$\mu \asymp \sqrt{\log n}$
Row/column averaging For clusters of size $\Theta\left(n^{1/2+\alpha}\right)$	$\mu \asymp \sqrt{\frac{\log n}{n^{2\alpha}}}$
Sparse SVD	$\mu \asymp \sqrt{\log n}$

See NIPS paper for more details

Summary so far

	Minimax rate	Computation
Normal means	$\mu \asymp \sqrt{\log n}$	Trivial
Biclustering	$\mu \asymp \sqrt{\frac{\log n}{k}}$	Seems hard to achieve lower bound

- Looked at some computationally efficient procedures but they don't achieve the information theoretic limits

How hard is the biclustering problem?

- Special case of many NP-hard problems !!!
- Densest k-subgraph
  - Given a graph $\mathcal{G}$ = (V,E,w) (w edge weights, possibly negative), and a number k, find the subgraph of $\mathcal{G}$ of size k which has largest density ρ
  - Density is just sum of edge weights divided by k
  - Easy to see that this recovers the bicluster in our problem on the bipartite graph induced by the matrix
  - Densest (unrestricted) subgraph problem is computationally easy
  - W.h.p densest (unrestricted) subgraph is the true bicluster only when size is O(n)

A few more NP-hard problems

- Quadratic assignment problem
  - Given W, A. Permute rows and columns of W to maximize tr(WA)
  - Using A to be a $(k \times k)$ block of 1s, padded with 0s, we can recover the bicluster
- Many more examples like this
  - L0 constrained sparse SVD

Two questions

1. Will approximation algorithms work for biclustering?
2. Previous connections can be unsatisfying. Is a random instance *actually* hard?
   - The worst-case might be hard, but why should we suspect this for a random instance?

A note on approximation algorithms

- Constant factor approximation is good enough
  - Remember we're only shooting for a *rate* optimal procedure
  - Need to ensure we can translate from approximation ratio of the objective back to statement about which rows and columns are recovered
- Even these seem unlikely – example densest k-subgraph best known approximation ratio is $O(n^{1/4})$, even worse for quadratic assignment problem
  - Negative result for densest k-subgraph – no constant factor approximation unless NP has sub-exponential algorithms

Why do we think even a random instance might be hard?

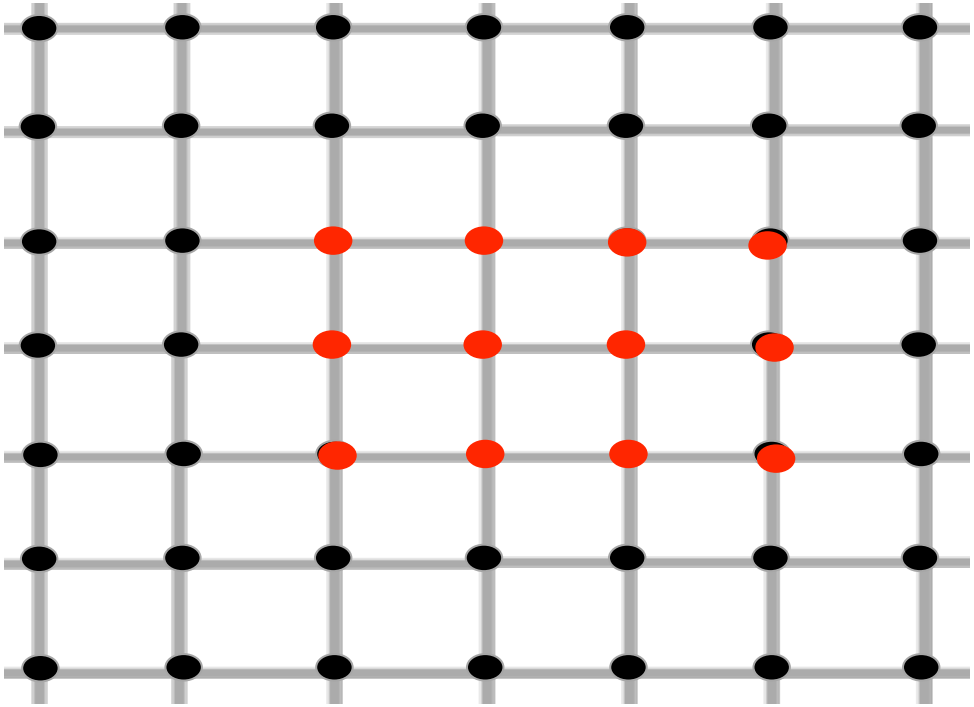
- Planted clique problem
  - Given an Erdos Renyi graph, $\mathcal{G}(n, 1/2)$ with a clique of size k planted in it, find the clique w.h.p
  - Impossible if $k \leq c \log n$, and "possible" if $k \geq C \log n$
  - Efficient algorithms only known if $k \geq C\sqrt{n}$
  - Large gap, conjectured to be hard for most parameter ranges
- There is even a cryptosystem based on the presumed hardness for the planted clique problem
  - A. Juels and M. Peinado. Hiding cliques for cryptographic security.

Why do we think even a random instance might be hard? – (ii)

- "Reduce" biclustering to planted clique problem (by thresholding, for instance at 0)
- Typically get even harder planted "near" clique problem

Can we say anything more?
More tradeoffs

- Consider the ``spatially localized'' biclustering problem



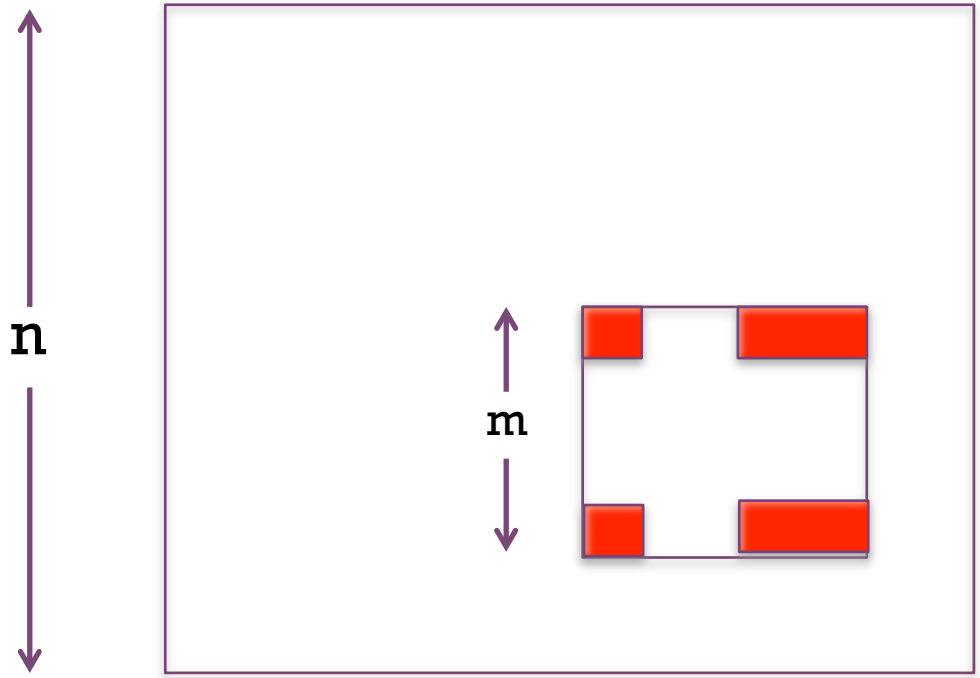
More tradeoffs (ii)

- Easier computation – "scan statistic" only costs $O(n^2)$

- Minimax rate gets "better"
  - Can detect smaller signals
  - $\mu \geq C \frac{\sqrt{\log n}}{k}$ suffices (this is tight)

Approximate clustering

- Motivation: Practitioner gives you an approximate clustering from "prior" knowledge
  - Note – does not use matrix
- Approximate clustering partially localizes the bicluster



Approximate clustering

- Minimax rate

$$\mu \geq C \max \left(\frac{\sqrt{\log n}}{k}, \sqrt{\frac{\log m}{k}} \right)$$

- Computation is naively $O(n^2 m^k)$
- Minimax rate shows two regimes depending on what the dominant "cost" is (approximately localizing the signal or exactly localizing it)
  - If m is O(k)
  - If m is O(n)

Active sensing approaches

Active measurements

Distilled sensing (Haupt et. al.)

- Lets return to the k-sparse length n vector normal means problem, with a slightly different setting
  - Allowed to make repeated measurements of some locations
    - For simplicity, assume 2n measurements
  - Want to compete with a passive learner who sees the entire vector twice
- Sequential thresholding algorithm
- Passive learner still needs $\mu \geq C_1 \sqrt{\log n}$
- A very simple argument shows the active learner only needs

$$\mu \geq C_2 \sqrt{\log k + \log \log n}$$

What does active learning buy us in biclustering?

- An easy result for biclustering
  - If we want to achieve the passive learning lower bound, but using an active strategy
  - Essentially possible if bicluster is smaller than $\log n$ (important case)
  - Most importantly the algorithm is computationally efficient
- Ignoring issue of active lower bounds and achieving them
  - Might require us to exploit the structure better?

Active learning summary

- Sometimes active learning can help in one of two ways
  - Can let us detect weaker signals
  - Can let us detect signals computationally efficiently
  - Also, allows us to tradeoff these two
  - Important to characterize this better

Summary (ii)

- Connections to some hard problems and some reasons to believe that biclustering is generally computationally hard
- Side information can make the problem computationally easier and let us detect even weaker signals
- Active learning can sometimes let us tractably detect weak signals

Conclusions and future work

- Structured normal means – test bed for computational and statistical tradeoffs?
- Showing hardness for random structured problem instances
- But really, can we improve minimax analysis?
  - *Computationally efficient* minimax estimators

$$\sup_{\theta} \mathbb{E}(L(\theta, \delta)) \asymp \inf_{\hat{\theta} \text{ efficiently computable}} \sup_{\theta} \mathbb{E}(L(\theta, \hat{\theta}))$$

- Active learning – gains in computational/sensing efficiency? Lower bounds?