Learning Probabilistic models for Graph Partitioning with Noise

Aravindan Vijayaraghavan

Northwestern University

Based on joint works with

Konstantin Makarychev

Microsoft Research

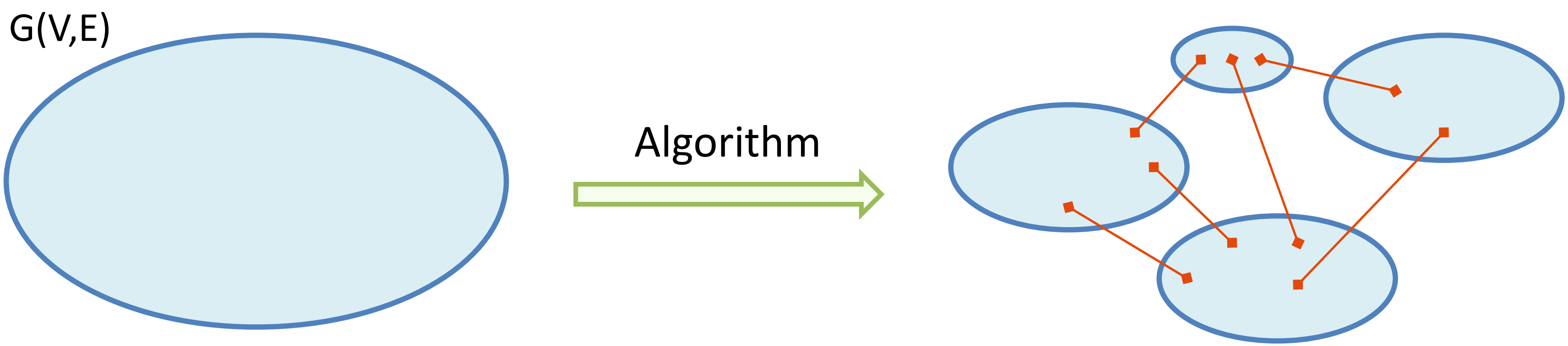
Yury Makarychev

Toyota Technological Institute at Chicago

Graph Partitioning problems

Goal: Divide V(G) into disjoint sets (clusters)

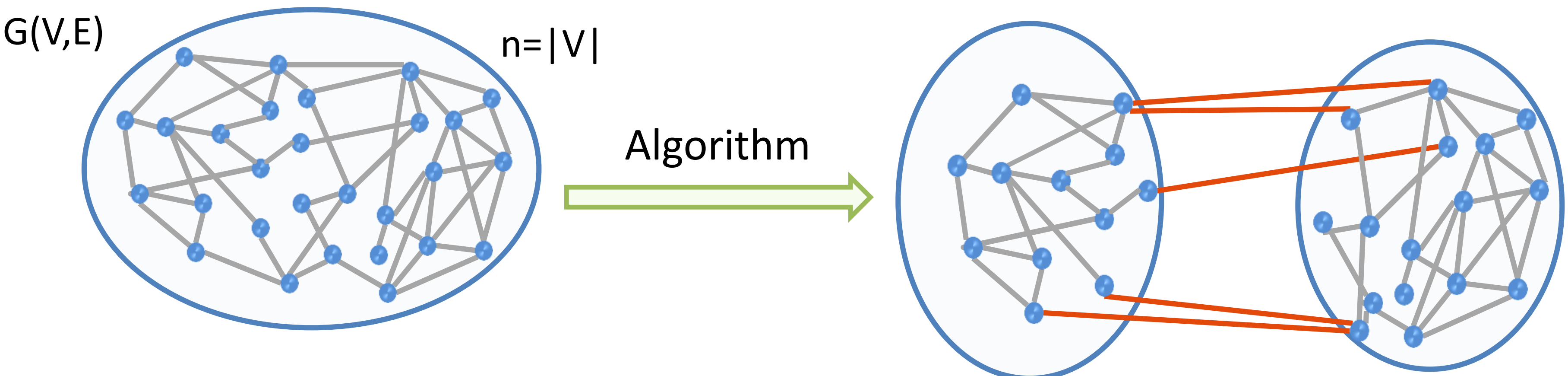
Minimize edges across clusters, subject to <constraints>



Graph Partitioning problems

Goal: Divide V(G) into disjoint sets (clusters)

Minimize edges across clusters, subject to <constraints>



- Balanced Cut
- Balanced K-way partitioning
- Sparsest cut
- Multicut
- Small set expansion

Divide V(G) into
two roughly equal pieces

Graph Partitioning problems

- NP-hard to solve exactly
- Central area of study in approximation algorithms

Algorithms

- Balanced Cut [LR88,ARV04] $O(\sqrt{\log n})$
- Multicut [GVY93] $O(\log n)$
- Sparsest cut [AR95,LLR95,ALN05] $O\left(\sqrt{\log n}\right)$
- Small set expansion [RST10,BFKMNNS11] $O(\log n)$

Hardness

- No PTAS [Khot02,GVY93,AMS07]
- No constant approximations assuming UGC and variants

  General Sparsest cut, Multicut [KV05] , Balanced Cut [RST11]

Only $\mathrm{poly}(\log n)$ *approximation algorithms known (worst-case)*

Can we do better using Average-case analysis?

Approximation ratio = $\max_{instances\ I} \frac{Alg(I)}{OPT(I)}$

Average-case Analysis

Average-case: Probability Distribution over instances

Average-case approximation ratio α w.r.t. distribution $\mathcal{D}$:

$$\operatorname*{Prob}_{I \leftarrow \mathcal{D}}\left[\frac{Cost_{Alg}(I)}{Cost_{OPT}(I)} \leq \alpha\right] = 1 - n^{-\omega(1)}$$

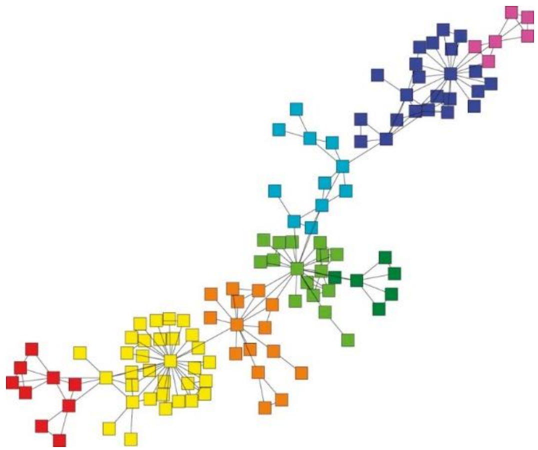
Main Challenges

- **Modeling Challenge:** Rich enough to capture real-world instances e.g. uniform distribution not usually realistic.

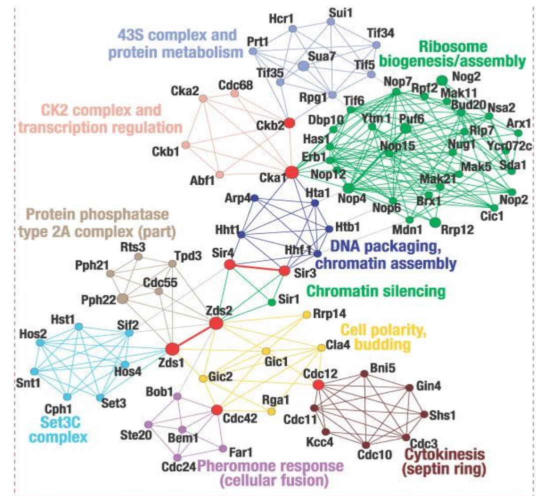
- **Algorithmic Challenge:** Want much better than worst-case

Models for Clustering Graphs

Graph represents similarity information between items (vertices)



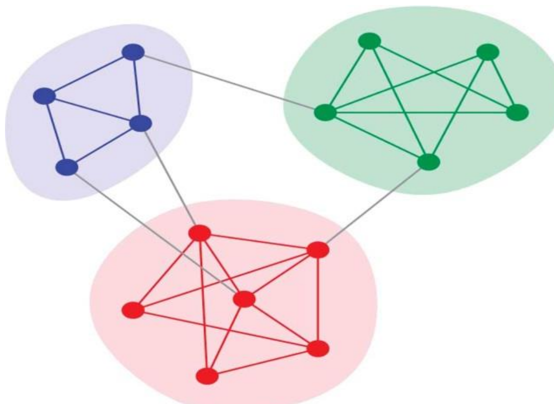
Collaboration network in a research lab
[Newman. Nature Physics'12]



Protein-protein interaction graph
[Palla, Derényi, Farkas and Vicsek. Nature' 05]

Nice clustering of vertices with:

- *Many edges inside clusters (related nodes)*
- *Few edges between clusters (unrelated nodes)*

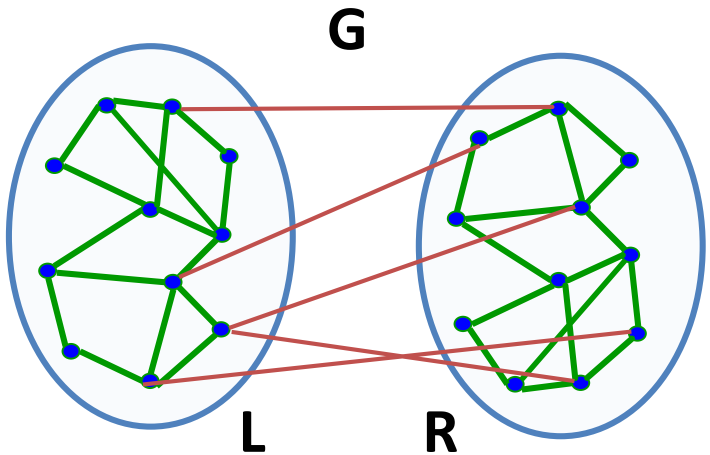


Distribution $\mathcal{D}$ generates such instances

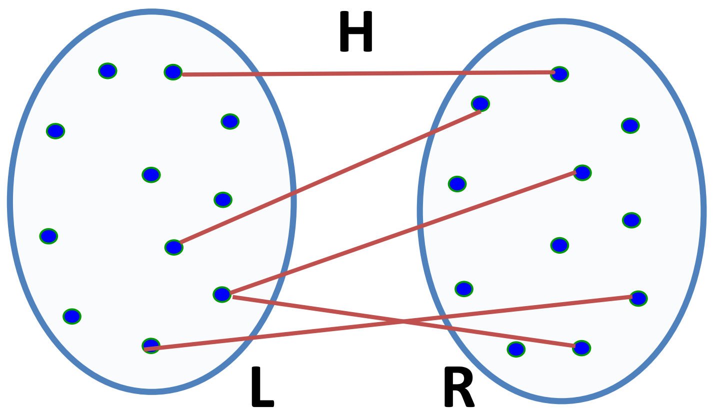
Goal 1: Approximating Objective

Distribution $\mathcal{D}$ generates instances

- *With ``nice'' partitioning*
- *Many edges inside clusters*
- *Few edges between clusters*



Planted Partition/Cut: $H(L, R)$



Given: Graph $G(V, E)$

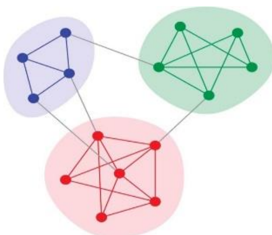
Goal: Find a (balanced) partition

Performance of algorithm:

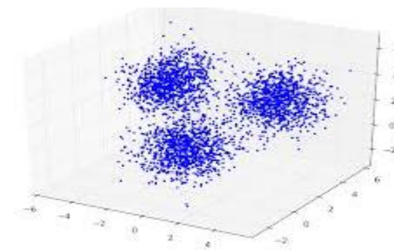
Cost of cut compared to planted cut $E_H = E_G(L, R)$

Goal 2: Learning Probabilistic model

Assumption: Ground truth probabilistic model generating data



Graph models for community detection

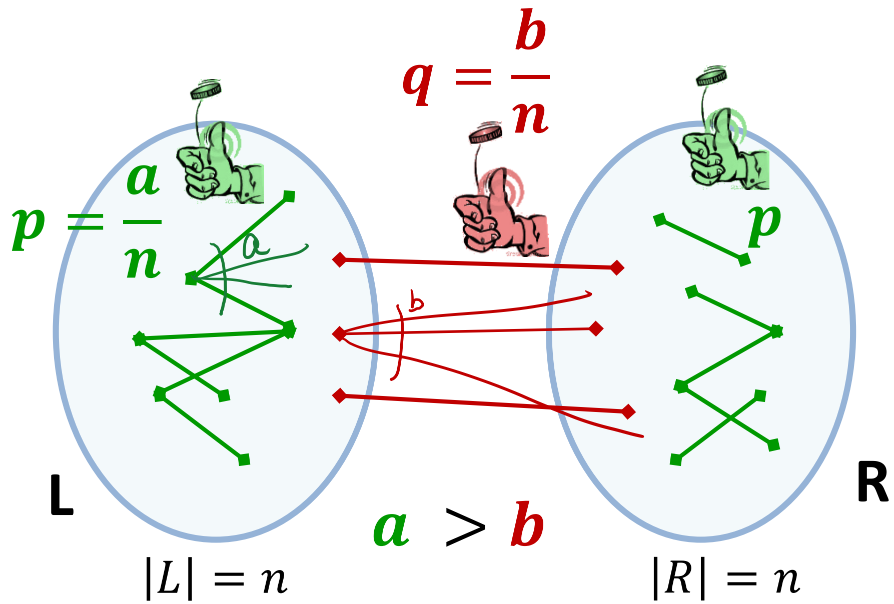


Analogous to Mixture of Gaussians for clustering points

Learning goal: Can we learn the probabilistic model i.e. recover the communities/planted partition from generated graph?

A simple random model : SBM

Two communities L, R of equal size n (L, R not known to us)



Each edge chosen independently at random with probability $p = \frac{a}{n}$ *or* $q = \frac{b}{n}$ *depending on inside cluster or between clusters.*

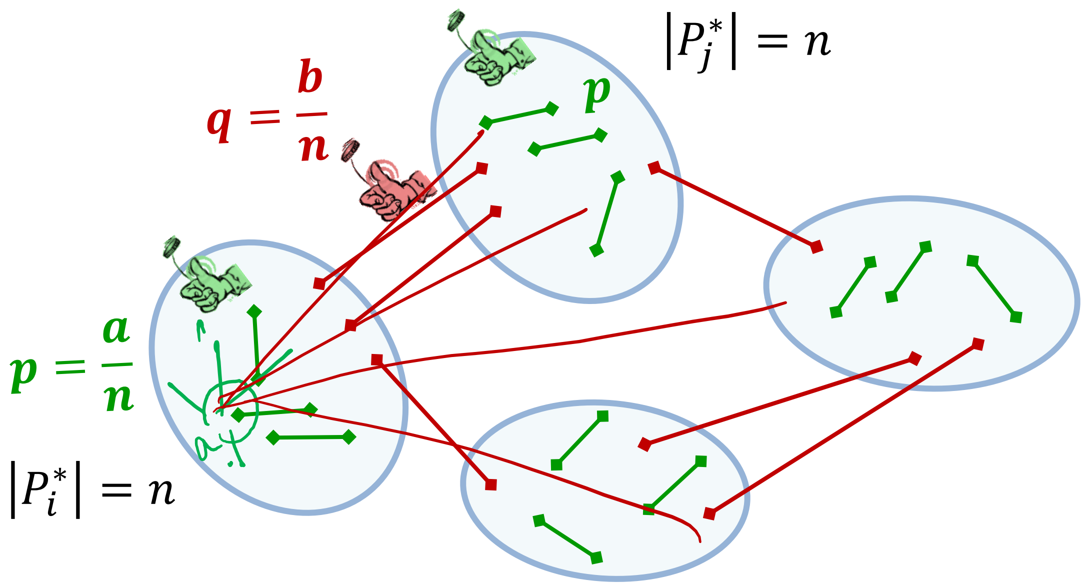
Stochastic Block Model (SBM): above model represented as SBM(n,2,a,b)

Learning Goal: Recover L, R i.e. the underlying community structure in poly(N) time.

Stochastic Block Model $SBM(n, k, a, b)$

Most commonly used probabilistic model for clustering graphs

k communities of equal size n. Number of vertices $N = nk$.



Number of edges

$$m \approx \frac{Na}{2} + \frac{N\,(k-1)b}{2}$$

a= $\mathbb{E}$[degree of a vertex

$a > b$

Every edge chosen independently at random.

Prior Work on Learning SBMs

- In Statistics, Social Networks, ML...
  [WBB'76], [HLL'83], [SL'90], [NJW'02], [ST'07], [L'07], [DKMZ'09]...
- Also called Planted Partitioning models in CS
  [BCLS 87],[Bop 88], [JS 92], [DI 98], [FK'99], [McS02], [Coj 04]...
- Also been generalized to handle different degrees, intercluster/ intra-cluster probabilities etc. [DHM'04, CL'09, CCT'12,AS'15]

Three broad classes of results:

1. Exact Recovery: Classify each vertex correctly. Need $a = \Omega(\log n)$.
2. Partial Recovery: Classify $1-\delta$ fraction of vertices correctly.
   Works in the sparse regime i.e., $a, b = O(1)$.
3. Weak Recovery: Classify better than a random partition.
   Sharp results [Mossel-Neeman-Sly , Massoulie].

Learning SBMs Exactly

Classify each vertex correctly. Need $a = \Omega(\log n)$.

[Bopanna88, McSherry 02,…] Spectral techniques w.h.p. find communities when

k=2: $a - b > C\sqrt{(a+b)\log n}$ i.e., $a = \Omega(\log n)$

general k: $a - b > C\sqrt{(a+(k-1)b)\log n}$

[MNS15, ABH14, AS15] Gives sharp characterization (in terms of a, b) for when exact recovery is possible.

Sparse Regime

[Coj06] Polytime algorithm that w.h.p. finds min. balanced cut if

$$(a-b) > C\sqrt{a+b}$$

Partial Recovery [MNS14, CRV 15, AS15]: Polynomial time algorithm that w.h.p. recovers communities with at most $(1-\delta)N$ misclassified vertices when $\frac{(a-b)}{\sqrt{a}} > C\sqrt{k\log(1/\delta)}$

Weak recovery [MNS'12,MNS'14, Mas'14]: Sharp phase transition for when we can find w.h.p. *a partition with non-trivial correlation* depending on whether $\frac{a-b}{\sqrt{a+b}} > 1$ (for $k = 2$)

[AS16] show weak recovery for k-communities if $\frac{a-b}{\sqrt{a+b(k-1)}} > 1$

Focus of this talk: Partial recovery

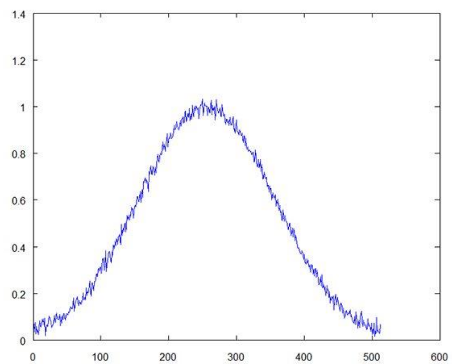
Drawbacks: Theory models vs Practice

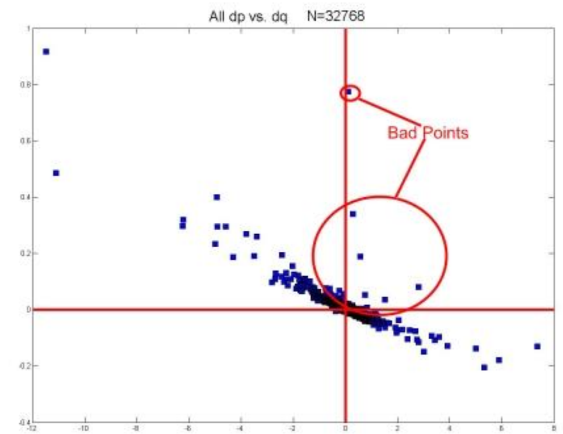
Theory vs Practice: Main criticism against theory (SBM)

Algorithms assume that data generated exactly from model (SBM)!

Dealing with Errors: data is always noisy!

e.g. Input errors, Outliers, Mis-specification





Fundamental criterion for judging learning algorithms

- Can we measure robustness of algorithms to errors?
- Develop algorithmic tools that are more robust

How Robust are Usual Approaches?

Spectral clustering: Project and cluster in space spanned by top k-eigenvectors.

Drawback: Spectral methods are not very robust

Eigenvectors brittle to noise: can add & delete just O(1) edges.

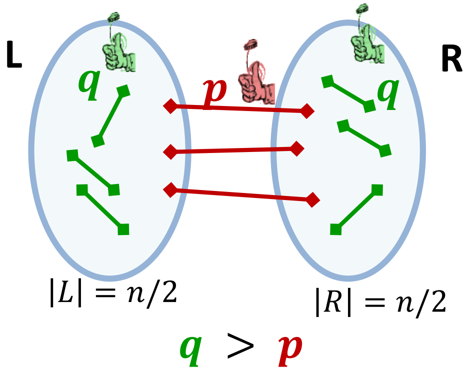
- Other algorithms based on counting paths, random walks, tensor methods are also not robust.

Maximum Likelihood (ML) estimation: Find the best fit model measured in KL divergence (measure of closeness for distributions)

Drawback: ML estimation is typically NP-hard!

- Heuristics like EM typically get stuck in local optima.

Drawbacks of Random Models



Unrealistic properties:

- Too much independence
- Does not have real-world graph properties
  - Small cliques, Concentrated degrees

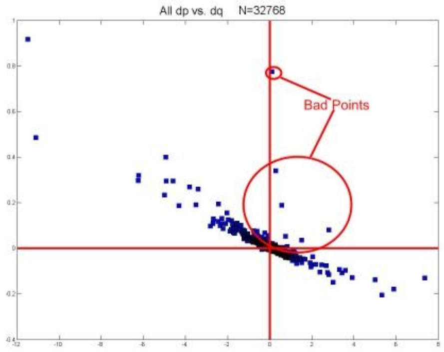
Properties of real-world graphs:
Heavy-tail degree distributions, dense subgraphs, high clustering coefficients [FFF’97, KRRT’00,NBW’06]

General enough average-case models capturing real-world instances?

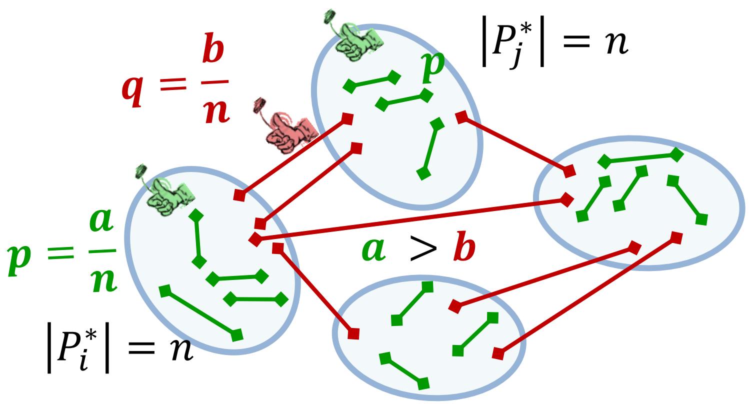
Beyond Simple Random models

1. **Realistic average-case models/semi-random models:**
   [Blum-Spencer, Feige-Kilian] Incorporate some random choices and some adversarial choices in generating input

2. **Handling Modeling errors:**
   Learning a probabilistic model like SBM, in the presence of various modeling errors



Monotone Adversaries [Feige-Kilian]



SBM(n, k, a, b)

<u>Monotone [Feige-Kilian'99]:</u>

Random model + Adversary can

1. delete edges between clusters
2. add edges inside clusters

Monotone: "Planted" solution is even better

SDPs used to make spectral arguments robust [FK99]:

recovers if $a - b > C\sqrt{(a+b)\log n}$ i.e. $a > \log n$

Extensions to k-way partitioning using convex relaxations [CSX'12,ABKK15]

Monotone Adversaries

Model: Random model + Adversary deletes edges between clusters & add edges inside clusters

Monotone: “Planted” solution is even better

Lower bounds for monotone adversaries [MPW 2016]:

Give first separation from simple random model (SBM)

Weak-recovery impossible when $(a-b) < c'\sqrt{a+b}$ where $c' > 1$

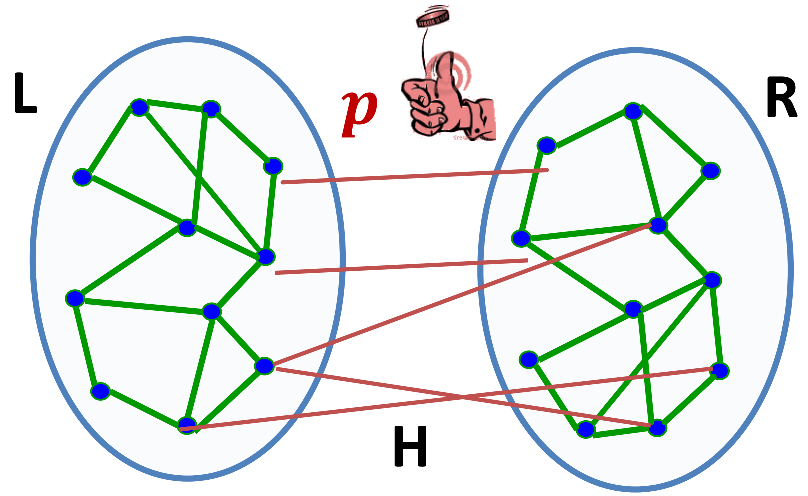
Open Question. Simple algorithm (non-SDP) e.g. spectral that are robust to monotone adversaries?

Models still assume lot of independence: essentially, each edge chosen independently at random

Semi-random model in

[Makarychev-Makarychev-V'12]

Aim: To capture arbitrary correlations inside clusters



Model

1. Inside cluster edges: arbitrary

2. Edges between clusters: random*

Perfect (arbitrary) partitioning + random noise

Theorem. Polytime algorithm finds a balanced cut $(S, \bar{S})$ which w.h.p. cuts $O(|E_H|) + n\sqrt{\log n}$ edges i.e.,

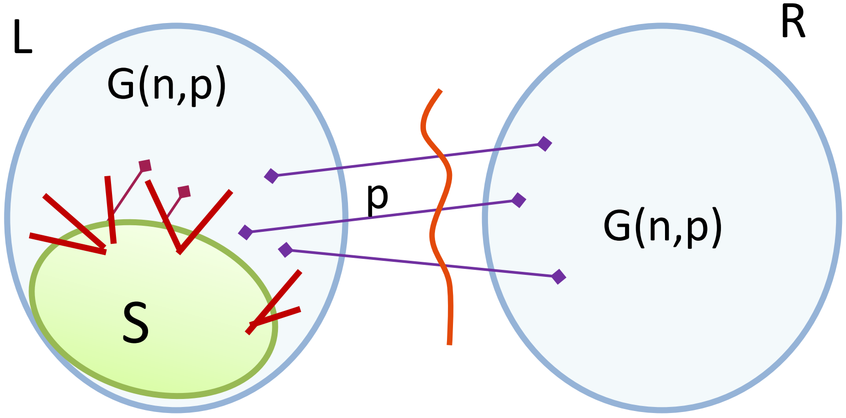
- O(1) approximation if $|E_H| = \Omega\left(n\sqrt{\log n}\right)$

*Like [FK99], adversary can also delete some between clusters edges E_H

Recovering the Planted Partition

Algorithms give balanced cut $(S, V\backslash S)$ with cost $\leq C \cdot |E_H|$

Recovery : How close is to ground truth $(\boldsymbol{L}, \boldsymbol{R})$?

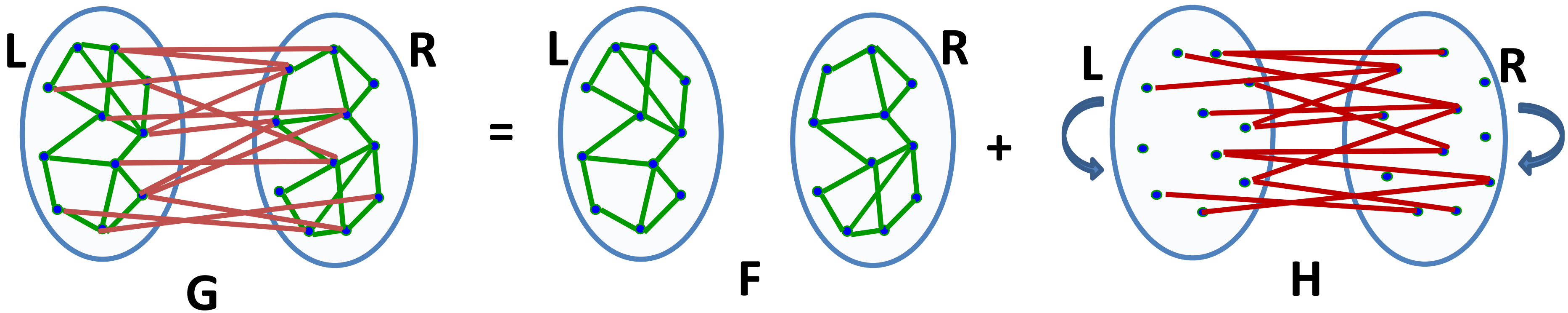


Can not recover in general !
Need assumptions about expansion inside the clusters

Partial Recovery [MMV12]. If expansion inside L > C. expansion(L,R), recover upto accuracy ρn vertices w.h.p. if $a > (\log n)^{1/2}/\rho$

Uses algorithm for semi-random Small Set expansion recursively

Random Permutation Invariant Edges (PIE) model [Makarychev-Makarychev-V'14]



Model:

1. Inside cluster edges F: arbitrary / worst-case

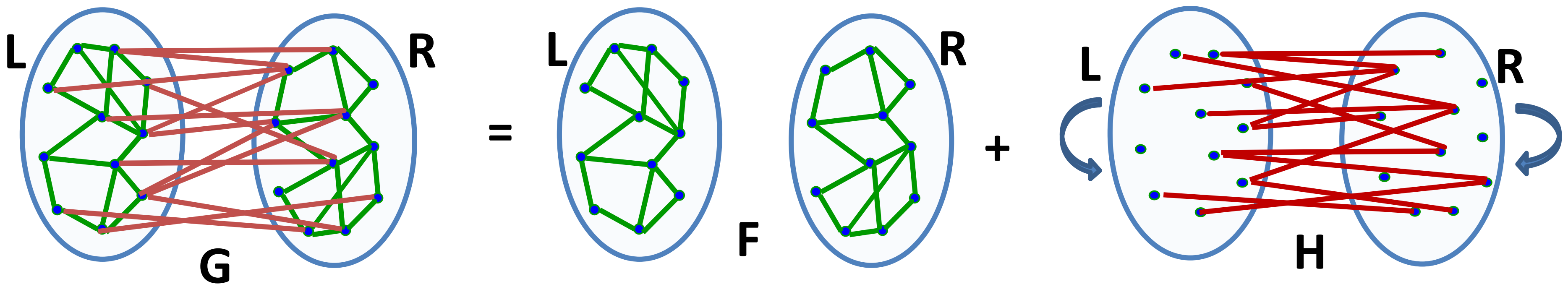
2. Between cluster edges H: ← $\mathcal{D}$; arbitrary/ worst-case

$\mathcal{D}$ **:** any distribution invariant to permutations of L and R

But this is worst-case instance !!

(or) $\mathcal{D}$ is symmetric w.r.t. vertices in L, and vertices in R

Advantages of Model



Capturing independence between F and H

- More general than all previous models
- Intra-cluster: worst-case. Inter-cluster: capture complex distributions
- Allows properties of real-world graphs like large cliques, dense subgraphs, clustering coefficient etc.

Result: Constant factor approximation algorithms in PIE model

Theorem [MMV'14]**.** Polytime algorithm that finds a balanced cut $(S, \bar{S})$ which w.h.p. cuts $O(|E_H|) + n\log^2 n$ edges

- O(1) approximation if $|E_H| = \Omega(n\log^2 n)$*

Interpretation: Min Balanced Cut is easy on any average-case model that satisfies the property of *permutation invariance.*

Open Questions.

1. Similar guarantees for k-way partitioning?
2. Conditions under which we can learn the model (recover planted partition)?

LEARNING WITH MODELING ERRORS

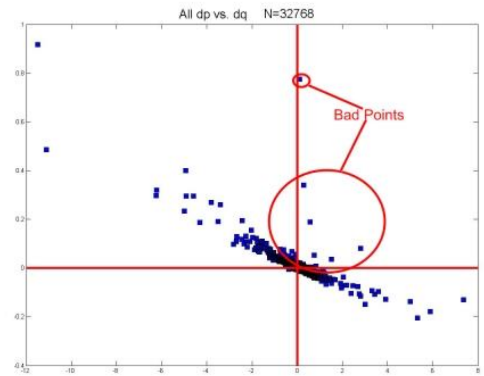
Learning with Modeling Errors

Dealing with Errors: data is always noisy!

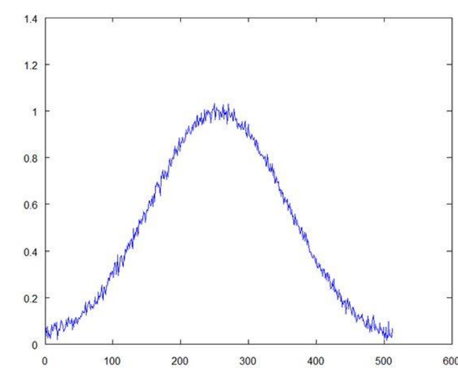
e.g. Input errors, Outliers, Mis-specification

Want to capture the following errors:

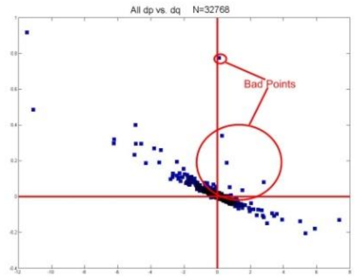
- Outliers or corruptions



- Model misspecification



Outliers or Input Errors

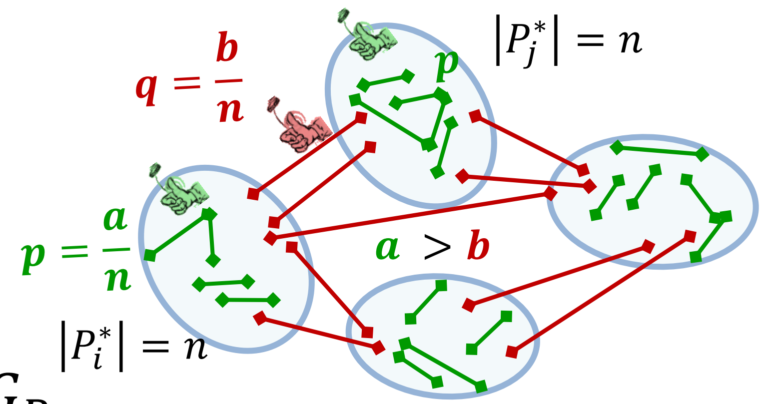


Captures up to ϵ fraction of the edges have errors/ corrupted.

Graph G generated as follows:

1. G_R generated from $SBM(n,k,a,b)$
2. Adversary picks $\epsilon_1, \epsilon_2 \geq 0$
   such that $\epsilon_1 + \epsilon_2 = \epsilon$
3. Adversary deletes $\epsilon_2 m$ edges from G_R
4. Adversary adds $\epsilon_1 m$ edges to the
   remaining graph to get G.

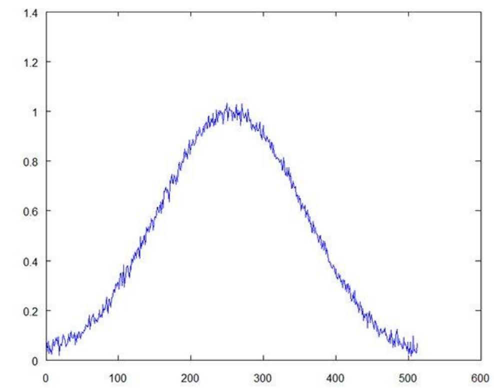
- Corruptions can be very correlated.



$\mathbf{SBM(n,k,a,b)}$

Model Misspecification in KL divergence

- Assumption of Data Analyst: Graph $G(V,E)$ drawn from model i.e. $G \sim SBM(n,k,a,b)$
- What if graph G is drawn from Q,
  a distribution that is close to $SBM(n,k,a,b)$?



KL divergence between probability distributions P, Q:

$$d_{KL}(Q,P) = \sum_{\sigma \in events} Q(\sigma) \log\left(\frac{P(\sigma)}{Q(\sigma)}\right)$$

- Graph is drawn from any distribution Q that is ηm close in KL to SBM, where m= number of edges.
- Captures upto $O(\eta m)$ adversarial edge additions.
- Edge draws can be dependent.

Robustness Learning Guarantees

$SBM(n,k,a,b)$: $N = nk$ vertices with k clusters of equal size. No. of edges = m

- Algorithms tolerates outlier errors up to ϵm, model specification up to ηm (think of $\epsilon, \eta \sim 0.01$).

Theorem[MMV16]**.** Given instance drawn from any distribution that is (i) ηm close to $SBM(n,k,a,b)$ in KL-divergence with (ii) ϵm outlier edges (iii) *any* monotone errors polytime algorithm to recover communities up to δN vertices where

$$\delta \leq O\left(\frac{(\sqrt{\eta}+\epsilon)(a+(k-1)b)}{a-b} + \underbrace{\frac{\sqrt{a+(k-1)b}}{a-b}}_{\text{no noise}}\right)$$

- Good partial recovery for $\eta, \epsilon = \Omega(1)$:

$$\text{if } (a-b) > C\sqrt{a+(k-1)b},\ \epsilon, \eta \ll \frac{a-b}{a+b(k-1)}$$

Near Optimal for Edge Outliers (only)

- Can amplify accuracy to match bounds of [Chin-Rao-Vu] for δ-recovery even in noiseless case.

Theorem. Given instance of $SBM(n,k,a,b)$ having m edges with ϵm outlier edges (adversarial) , recovery up to δN vertices if

$$\underbrace{\frac{(a-b)}{\sqrt{a}} > C\sqrt{k\log(1/\delta)}}_{\text{Condition in [CRV15] (zero noise)}}, \qquad \frac{(a-b)}{\epsilon(a+(k-1)b)} > \frac{C}{\delta}$$

Lower bound for $k=2$ communities: indicates this is correct dependence for both the terms, up to constants. For δ-recovery, need

$$\frac{(a-b)}{\sqrt{a+b}} > c\sqrt{\log(1/\delta)}. \qquad \frac{(a-b)}{\epsilon(a+b)} > \frac{c}{\delta}$$

Related Work

Deterministic Assumptions about data [Kumar Kannan 10]:

Noise needs to be structured i.e. strong bound on spectral radius

Vertex Outliers: [Cai and Li, Annals of Statistics 2015]

- Consider t vertex outliers. Design algorithms based on SDPs.
- For $a, b = C \log n$, they handle $\mathrm{O}(\log n)$ vertex outliers.
- To handle $t = \epsilon n$ outliers, they need $a = \Omega(n)$ i.e., dense graph.

- Comparison: Edge outliers more general than vertex outliers when $a, b \geq \log n$.
- Our algorithms handle ϵm outliers even in sparse regime $a, b = O(1)$

ALGORITHM OVERVIEW:
LEARNING SBM WITH ERRORS

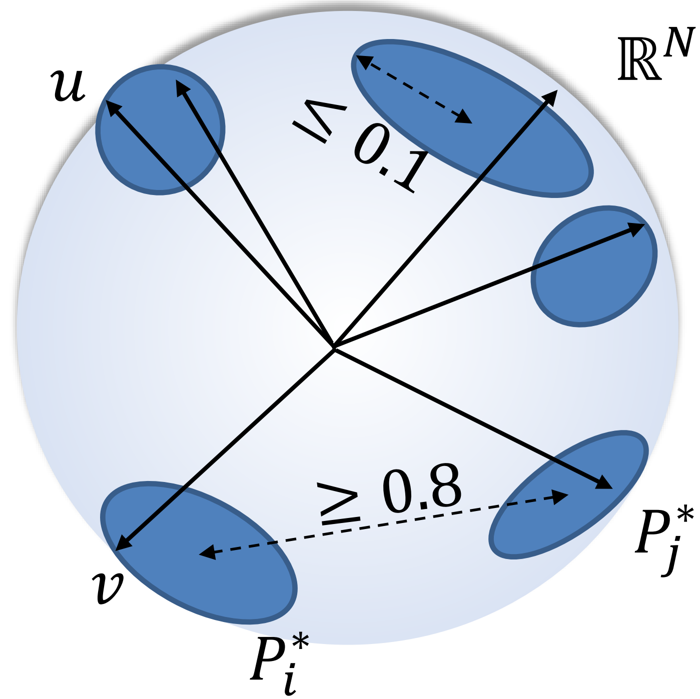
Algorithm Overview: Relax and Cluster

- Write down a SDP Relaxation for Balanced k-way partitioning (this is the ML estimator)

- Treat the SDP vectors as points in $\mathbb{R}^N$ for representing vertices.

- Use a simple greedy clustering algorithm to partition the vertices

Vectors given by SDP solution



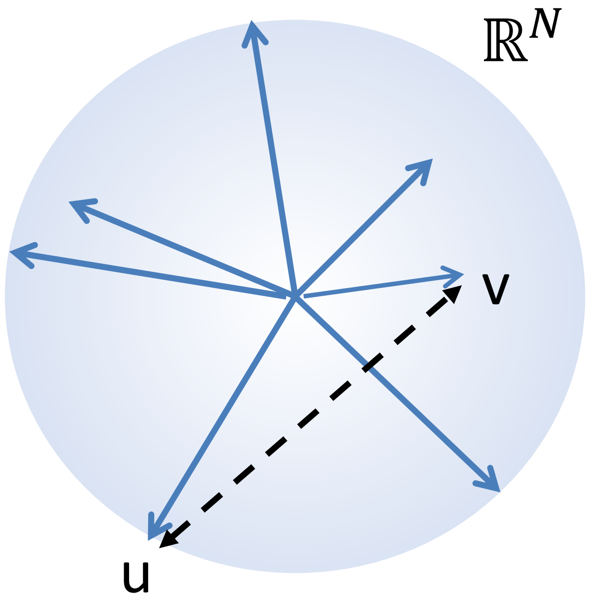
SDP Relaxations

SDP: $$\boldsymbol{min} \sum_{(u,v)\in E} \frac{1}{2}\|\bar{u}-\bar{v}\|^2$$

s.t. $\forall u \in V,\ \ \|\bar{u}\|_2 = 1, \forall u, v \in V\ \ \langle \bar{u}, \bar{v}\rangle \geq 0$

$$\textstyle\sum_{u,v\in V} \frac{1}{2}\|\bar{u}-\bar{v}\|^2 \ \geq n^2 k(k-1)/2$$

$$\boxed{d_{SDP}(u,v) = \frac{1}{2}\|\bar{u}-\bar{v}\|^2 \ \in [0,1]}$$



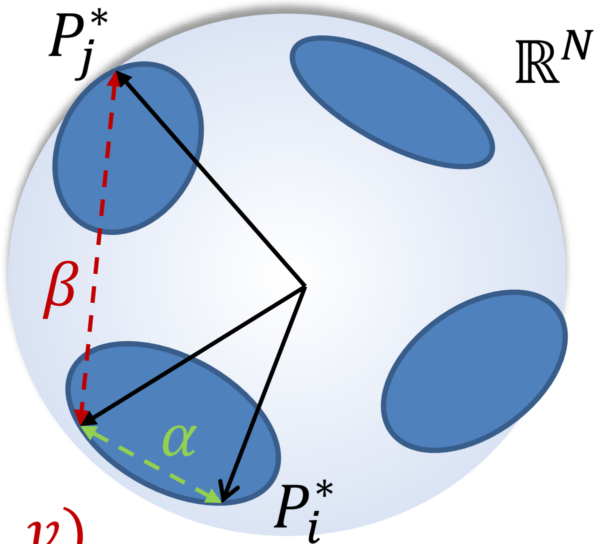
- **Intended solution:** $d_{SDP}(u,v) = 0$ if u, v in same cluster
  $= 1$ if u, v in different clusters
- $d_{SDP}(u,v)$ intuitive notion of “distance” (no triangle inequalities)

Intracluster & Intercluster Distances

Intracluster distance $\alpha = \text{Avg}_{u,v \in (V \times V)_{in}} d_{SDP}(u,v)$

- $(V \times \mathrm{V})_{in}$: pairs of vertices inside the communities $P_1^*, P_2^*, .., P_k^*$

SDP solution

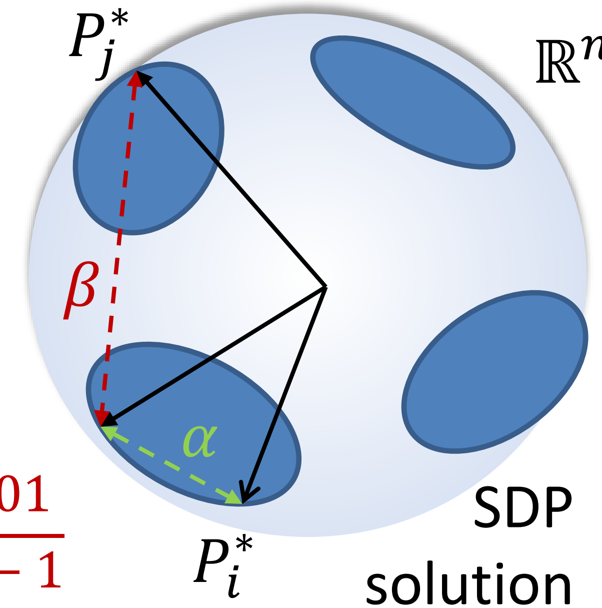


Intercluster distance $\beta = \text{Avg}_{u,v \in (V \times V)_{out}} d_{SDP}(u,v)$

- $(V \times V)_{out}$: pairs of vertices in different communities

Geometrical Clustering of SDP

Theorem. In $SBM(n,k,a,b)$, suppose $a+(k-1)b \geq C_1$, then with probability at least $1-\exp(-2N)$



(1) Average Intra-cluster distance

$$\alpha \leq \frac{c_2\sqrt{a+(k-1)b}}{a-b} + \frac{\epsilon(a+(k-1)b)}{a-b} \sim 0.01$$

(2) Average Inter-cluster distance

$$\beta \geq 1 - \frac{c_2\sqrt{a+(k-1)b}}{(a-b)(k-1)} - \frac{\epsilon(a+(k-1)b)}{(a-b)(k-1)} \sim 1 - \frac{0.01}{k-1}$$

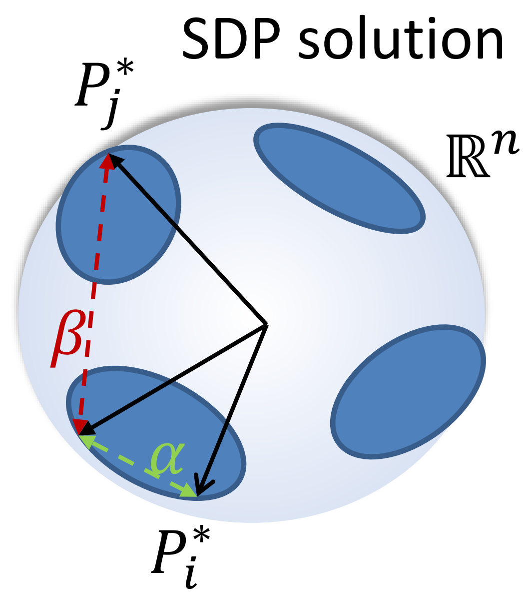
SDP vectors geometrically clustered acc. to communities:

- Points in same cluster are very close i.e. $\alpha \approx o(1)$
- Points in different clusters are far i.e. $\beta \approx 1 - \frac{o(1)}{k-1}$

The Algorithm

SDP vectors geometrically clustered acc. to communities:

- Points in same cluster are very close $\sim o(1)$
- Points in different clusters are far $\sim 1 - \frac{o(1)}{k-1}$

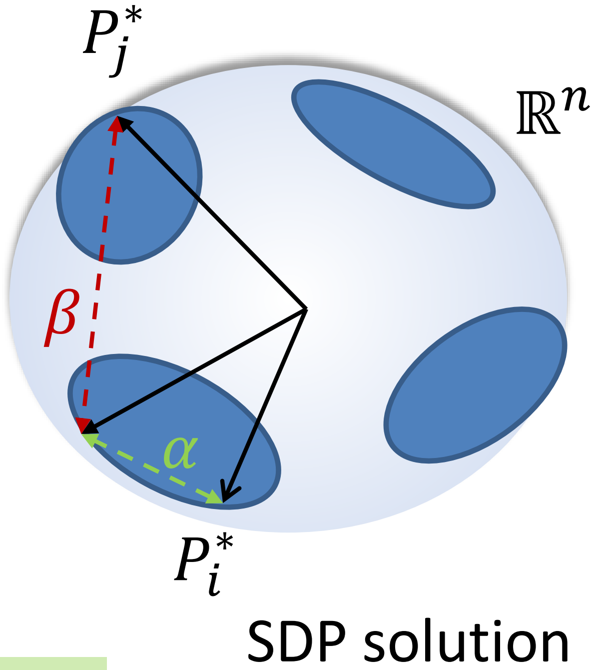


Simple Algorithm for $k = 2$ communities:

1. Pick a random vertex (or guess).
2. Cut out a ball of radius ½
3. Geometric clustering of points → o(n) vertices misclassified.

Clustering Algorithm for k communities

- Can't guess centers for k clusters
- Since k is large, random centers also doesn't quite work



Simple, greedy geometric clustering:

while (exist active vertices $A \subset V(G)$)
- $u = argmax_{v \in A} |Ball(v, 0.1) \cap A|$
- $Cluster\ C = Ball(u, 0.1) \cap A; \quad A = A \setminus C$

Distance concentration

- $\alpha = \text{Avg}_{u,v \in (V \times V)_{in}} d_{SDP}(u,v)$, $\beta = \text{Avg}_{u,v \in (V \times V)_{out}} d_{SDP}(u,v)$

Average # of edges inside communities $= a\frac{nk}{2}$

between communities = $b\,\frac{nk(k-1)}{2}$

Lemma: In $SBM(n,k,a,b)$, with m edges and with ϵm edge outliers, then with probability at least $1 - \exp(-2nk)$

$$sdp \geq \alpha\frac{ank}{2} + \beta\frac{bnk(k-1)}{2} - \underbrace{\mathrm{c}_2 nk\sqrt{a+(k-1)b} - \epsilon m}_{\text{error}}$$

- Uses Grothendieck inequality for sparse graphs: uses ideas from [Guedon-Vershynin 14]
- For $\mathrm{m} = \Omega(n\log n)$, spectral expansion/ JL+ ϵ-net suffice [KMM11,MMV12]

Takeaways and Future Directions

- More realistic average-case models for Graph Partitioning that are more general than simple random models
- Algorithms for learning in the presence of various modeling errors e.g. outlier errors or corruptions, monotone errors, model misspecification (in KL divergence).

Future Directions

- Other natural properties of average-case models (like permutation invariance) that enables tractability?
- Simpler algorithms e.g. spectral algorithms with similar guarantees?
- Unsupervised learning of other probabilistic models with errors (similar to [Lai et al, Diakonikolas et al. 16])?

Thank you!

Questions?

Drawbacks of Worst-Case Analysis

Limited by Worst-case analysis ?

Real-world instances are not worst-case instances !!

Capturing Smart Heuristics

- Differentiating smart vs trivial heuristics
- Systematically comparing heuristics

The Realistic Average-Case

Main Challenges

- **Modeling Challenge:** Rich enough to capture real-world instances e.g. uniform distribution not usually realistic.

- **Algorithmic Challenge:** Want good guarantees e.g. constant factor approximations

This talk: More Realistic Average-Case models

Examples: Semi-random models [Blum-Spencer, Feige-Kilian]