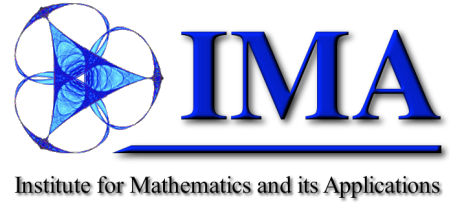
Finding overlapping communities

Brendan Ames

Institute for Mathematics and its Applications
University of Minnesota

IMA Postdoc Seminar
Thursday, July 5, 2012



Outline

1. Finding Overlapping Communities in Social Networks. Arora, Ge, Sachdeva, Schoenebeck. 2011

2. What I've been working on (joint with Steve Vavasis and Ting Kei Pong, C&O Dept, University of Waterloo)

3. What's left to do?

Arora et al.

- Traditional clustering techniques exploit the fact that the communities are nonoverlapping when looking for communities.
- Not representative of real-world networks where users can belong to several different communities.
- Present a planted case analysis: Propose a quasipolynomial time algorithm for finding overlapping communities in networks constructed in a particular way.

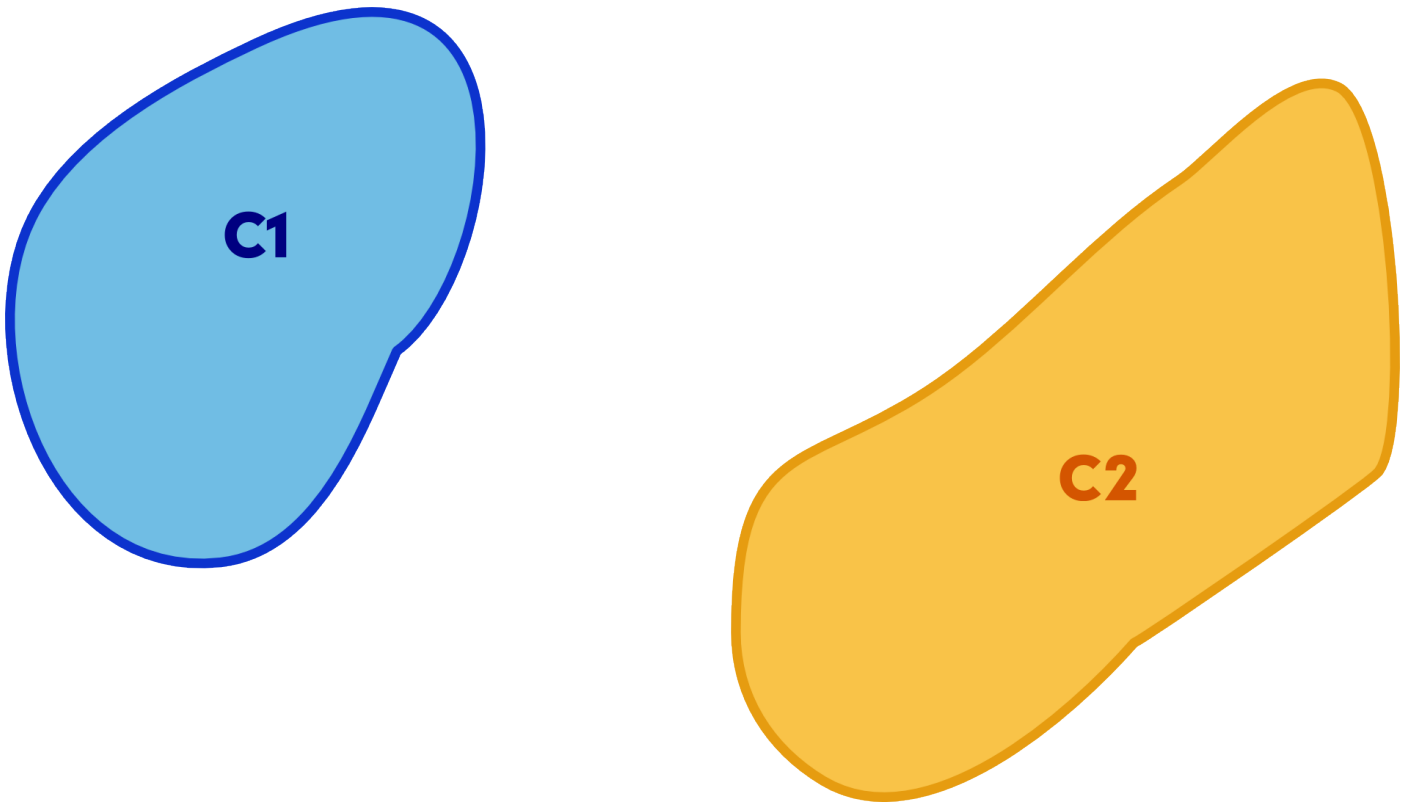
Communities in networks

- Social network may be represented as a graph.
- Nodes correspond to members of the network.
- Edges indicate “friendship”.
- Communities: groups of nodes connected with “lots of edges”.
- Cliques: groups of fully connected nodes (all users are friends).
- Not limited to social networks: internet search, communication/power networks, biological interactions, etc.

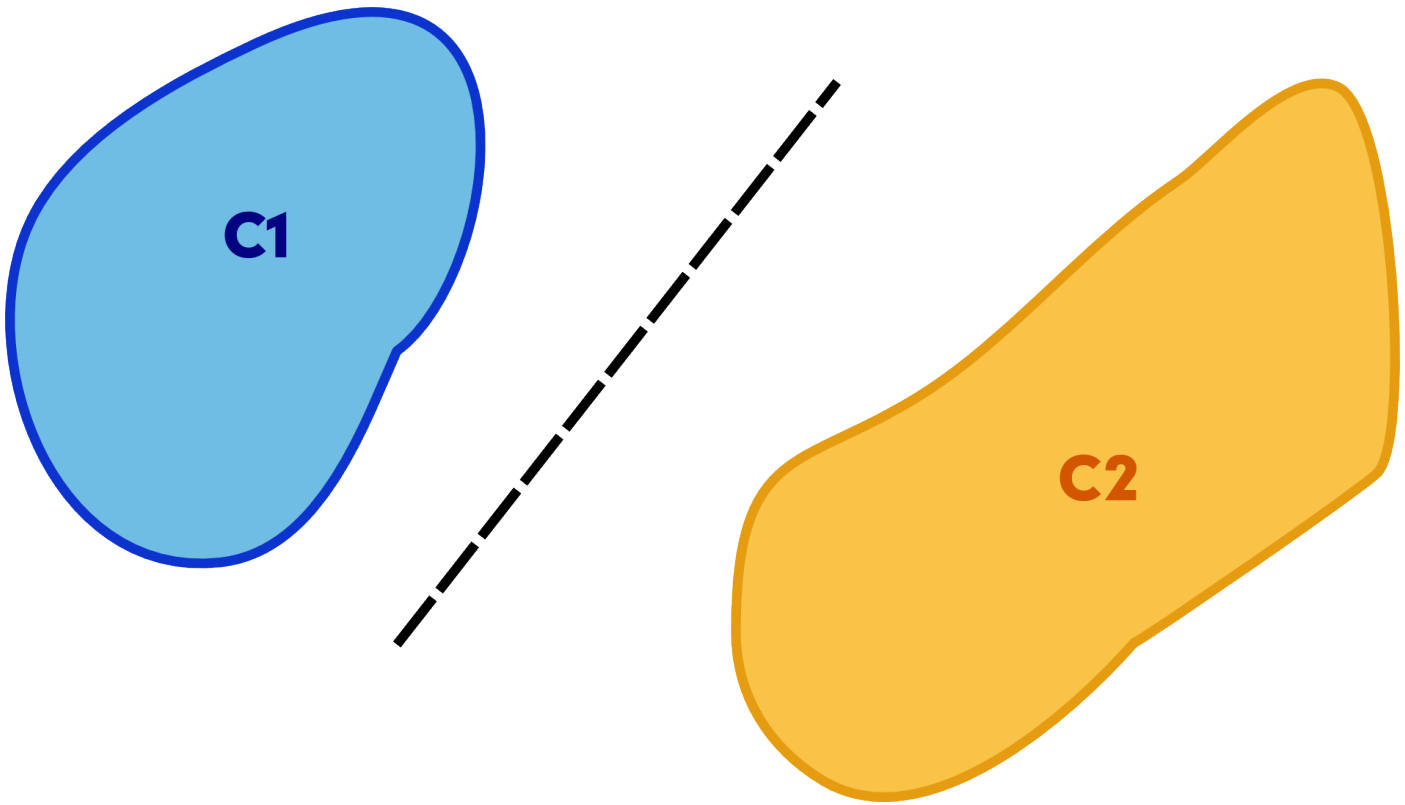
The nonoverlapping case

- Most heuristics are clustering based: partition the graph into clusters of similar users.
- Implicitly exploit the fact that the communities are separable (such a partitioning is appropriate):
  - i.e. communities are nonexpanding: densely connected with few edges leaving.
- Graph partitioning is intractable in general.
- In special cases can find nonoverlapping communities:
  - Ng et al. 2002, Ostrovsky et al. 2006, Ames & Vavasis 2010, Rohe et al 2010, Oymak & Hassibi 2011, Jalali et al 2011, Balakrishnan et al. 2011, Ames 2012

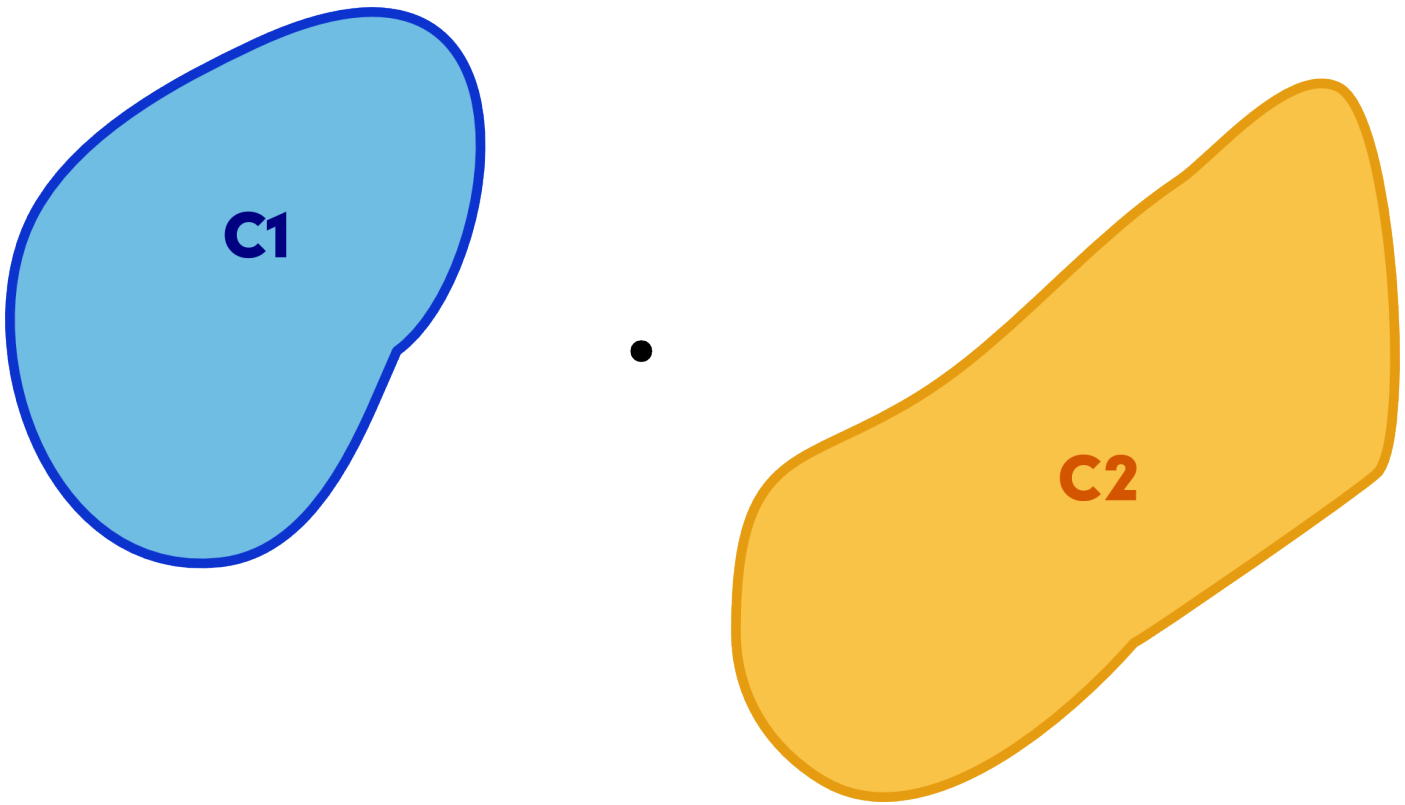
Example



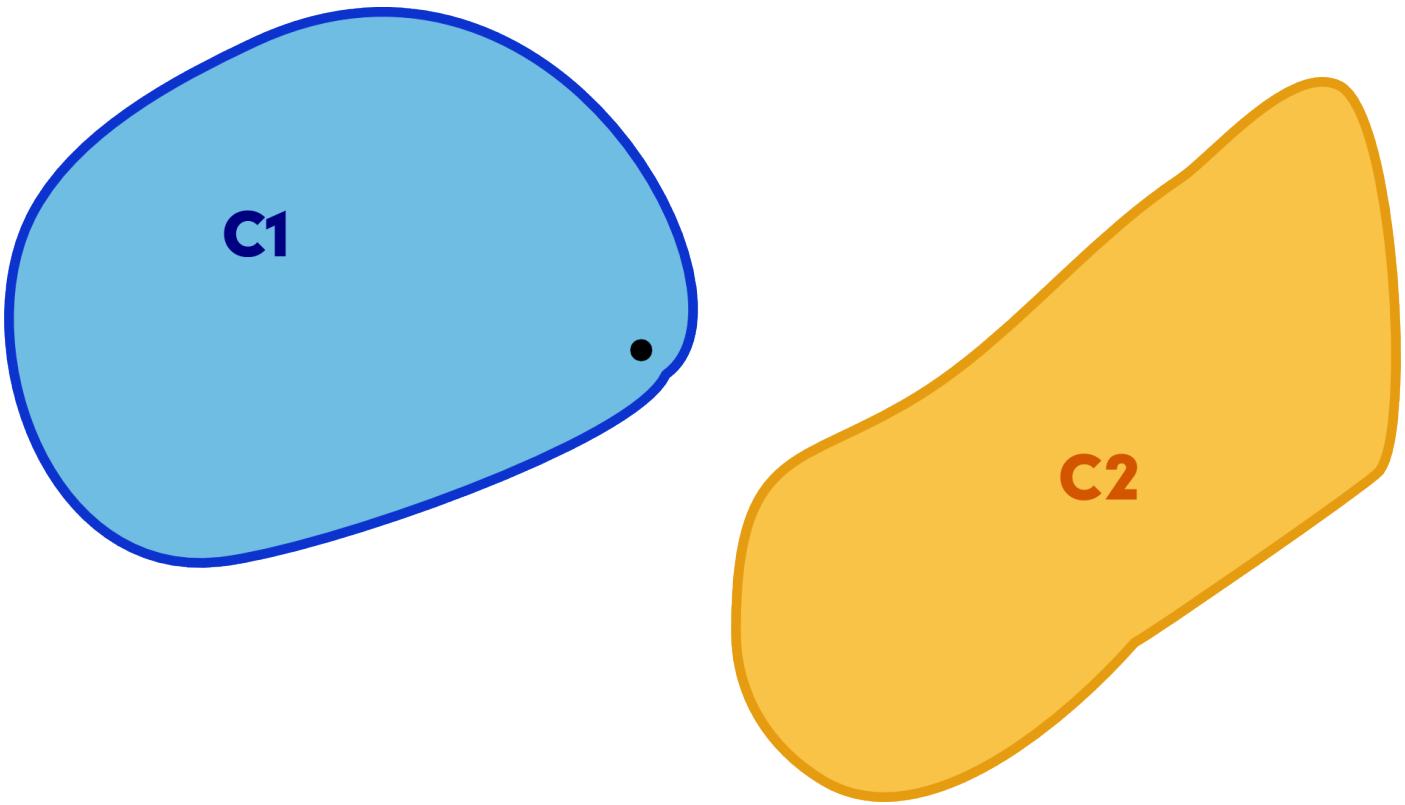
Example



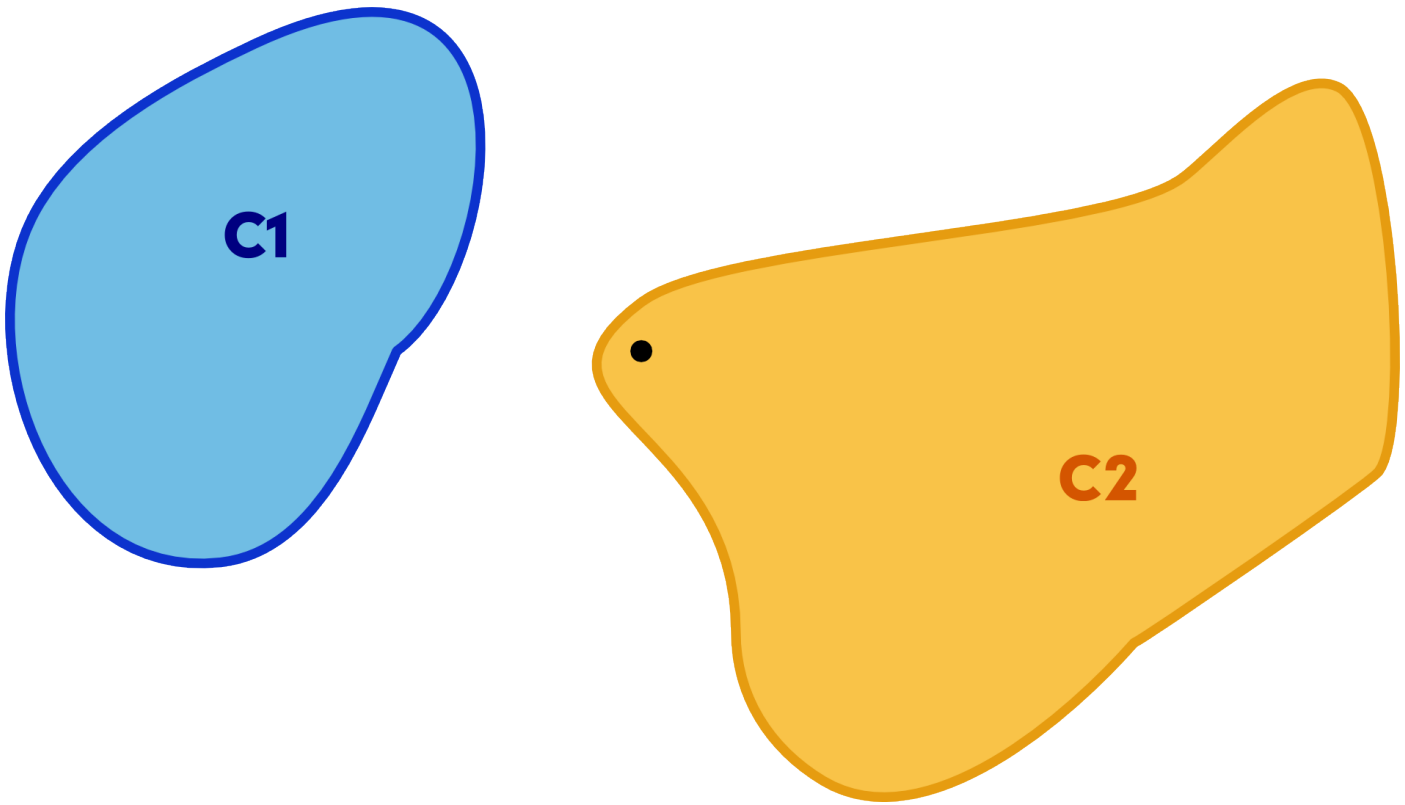
Example



Example



Example



Ego-centric networks

- Model of local behaviour.
- Consist of a single user (ego) and its neighbours.
- Thorougly studied in sociology.
- Individuals typically participate in only a small number of communities.
- Community membership based on homophily: people with shared interests will be friends.
- Have foci (e.g. family, workplace, social activities) around which joint activities are organized.

Notation

- Network is a graph $G = (V, E)$ on n users.
- Each user $u \in V$ belongs to exactly d communities.
- The largest community has cardinality k.

The Expected Degree Model

Assumption (Expected Degree Model)

- *Each node u in community C has affinity $p_u \in [\sqrt{\alpha}, 1]$ for some constant α.*
- *Have edge uv with probability $p_u p_v \in [\alpha, 1]$ if $u, v \in C$.*
- *If u and v are in the overlap of several communities, the probability of uv is the maximum of the $p_u p_v$'s.*

Two more assumptions

Assumption (The Gap Assumption)

A $w \notin C$ is adjacent to at most $(\alpha - \epsilon)|C|$ members of C.

- *i.e. nodes are better attached to the community than nodes outside the community.*
- *Should not be able to add nodes to a community.*

Assumption (Membership Assumption)

For any node, community membership accounts for fraction $\gamma > 0$ of the edges incident with it.

Special case: Cliques of same size

Theorem

- *Suppose that every node has community affinity* $\alpha = 1$ *(every community is a clique).*
- *Suppose further that all communities have size* $|C| = k$*.*
- *Then we can find every community with probability at least* $2/3$ *in time* $O(nk)2^{O(\log^2 d)}$*.*

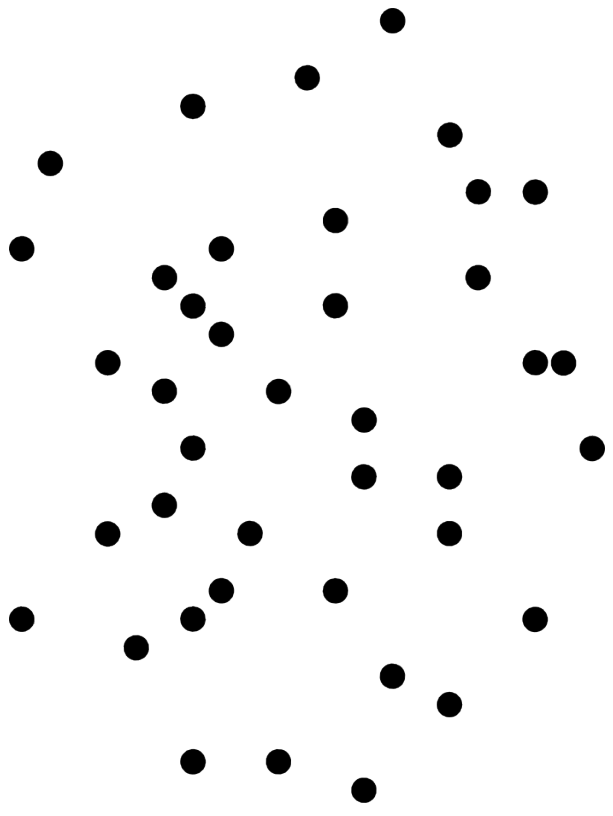
Step 1: Random sampling

Step (Random sampling)

Choose $9n/k$ nodes uniformly at random.

Claim

- *A node is chosen from any community with probability k/n.*
- *Choose ≈ 9 nodes from each community.*



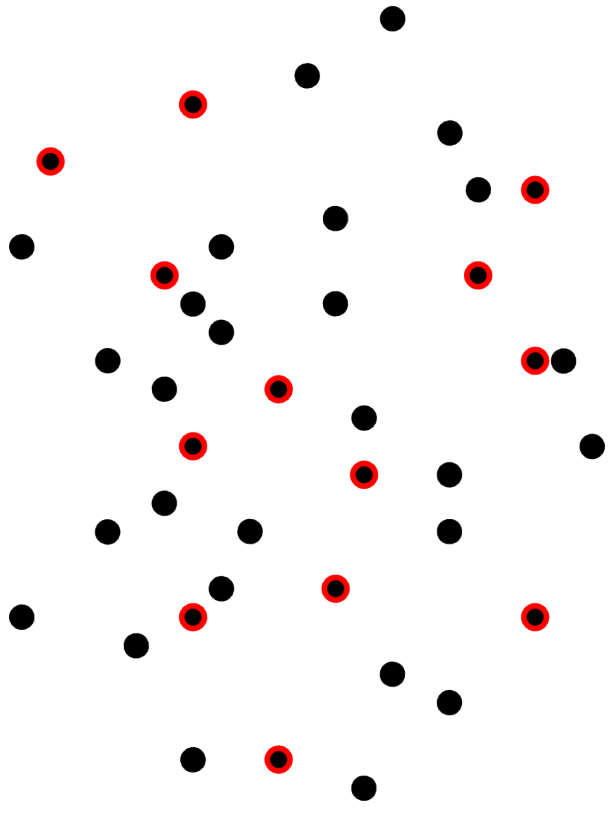
Step 1: Random sampling

Step (Random sampling)

Choose $9n/k$ nodes uniformly at random.

Claim

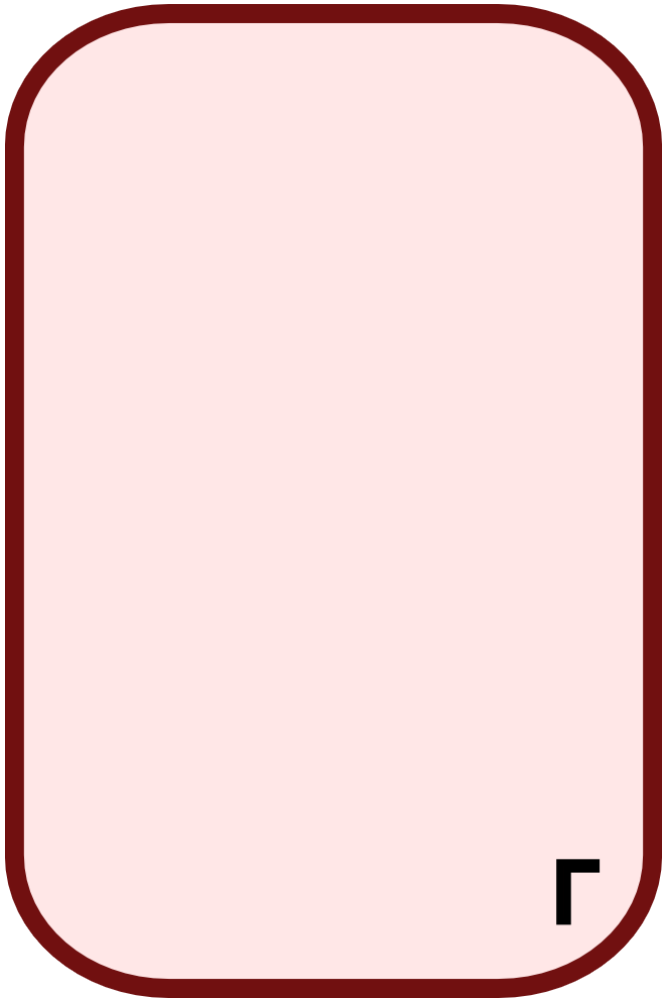
- *A node is chosen from any community with probability k/n.*
- *Choose ≈ 9 nodes from each community.*



Step 2: Neighbourhood subsampling

Step (Subsampling)

- *Pick sampled node v and find neighbourhood Γ of v.*
- *Randomly sample $S \subseteq \Gamma$ by including each $u \in \Gamma$ independently with fixed probability $p = c/k$ (c scalar depending on d, ϵ, γ).*



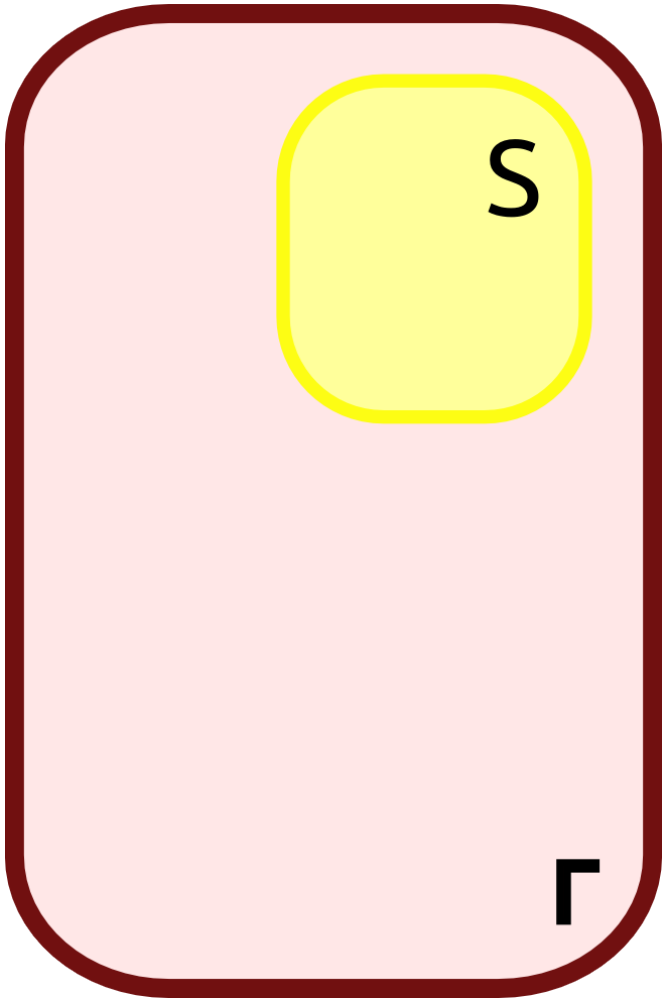
Claim

- $|S| \leq 3E[|S|] = O(d \log d)$ *w.h.p.*

Step 2: Neighbourhood subsampling

Step (Subsampling)

- *Pick sampled node v and find neighbourhood Γ of v.*
- *Randomly sample $S \subseteq \Gamma$ by including each $u \in \Gamma$ independently with fixed probability $p = c/k$ (c scalar depending on d, ϵ, γ).*



Claim

- $|S| \leq 3E[|S|] = O(d \log d)$ *w.h.p.*

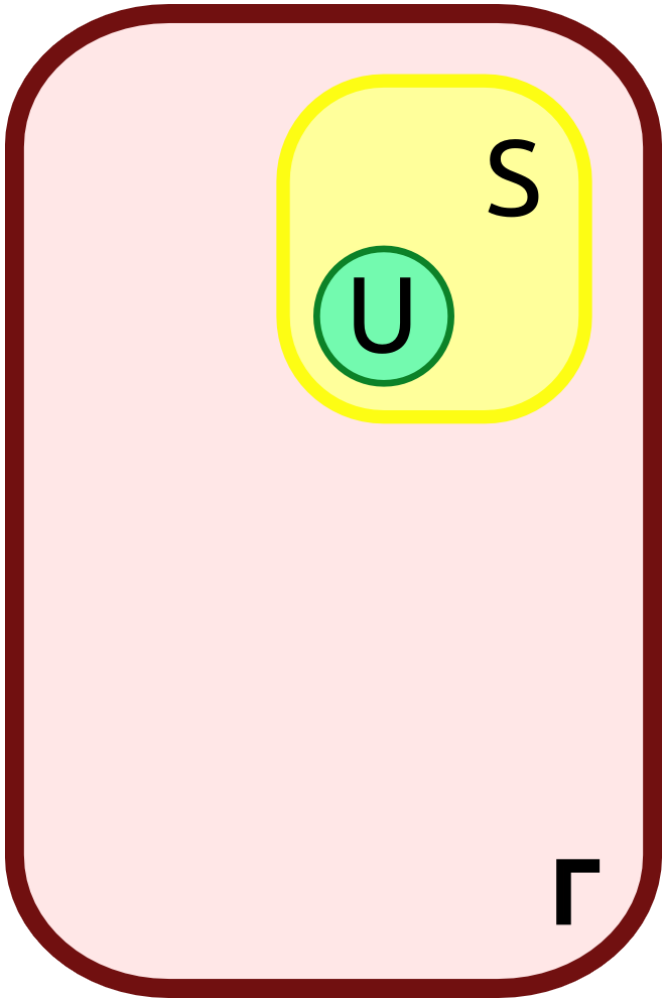
Step 3a: Clique finding

Step (Clique finding)

Find all cliques U of $G(S)$ of size at most $2pk$.

Claim

- *The set of such U includes $S \cap C$ for every community C containing v w.h.p.*



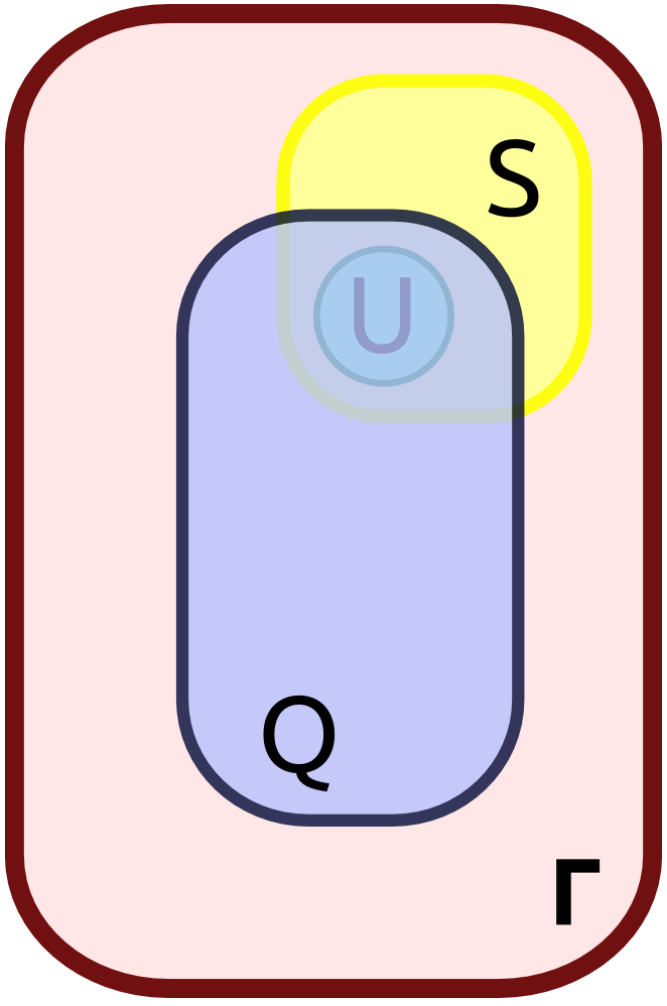
Step 3b: Expand cliques

Step (Expand U)

For all U: find set $Q \subseteq \Gamma$ connected to all nodes of U.

Claim

- *If $U = S \cap C$ then $C \subseteq Q$.*
- $|Q| \leq (1 + \epsilon/4)k$ *w.h.p.*
- $|E(Q)| \leq (1 + \epsilon/12)k$.



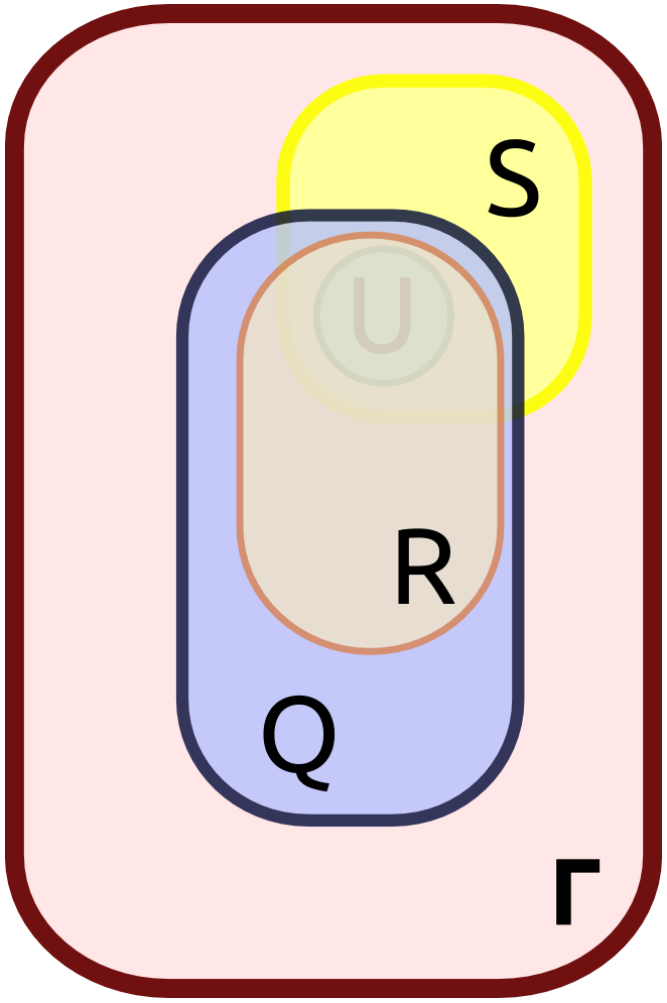
Step 3c: Output communities

Step (Output C)

- *Find set R of nodes with degree at least $(1 - \epsilon/2)|Q|$ in $G(Q)$.*
- *Output $R = C$ if it is a k-clique and satisfies the Gap assumption.*

Claim

- *Nodes in C have degree at least k.*
- *All others $\leq (1 - 3\epsilon/4)k$.*



Complexity of the algorithm

- $pk = O(d \log d)$ and d is small relative to n and k, we can find all cliques of S by exhaustive search quickly.
- Finding the set Q of neighbours of U requires

$$O(2pk|\Gamma|) = O\left(2pk^2 d/\gamma\right)$$

  operations since $|\Gamma| \leq kd/\gamma$.
- Finding R requires checking the degrees of each node in $G(Q)$, which can be done in time $O(k^2)$ since $|Q| = O(k)$.
- Putting everything together, we have a total runtime of at most

$$O(nk/\gamma) \times \text{max number of cliques} = O(nk/\gamma)2^{O(polylog(d))}.$$

The General Case

Theorem

- *Suppose that*
  - *each community satisfies the Expected Degree Model Assumption with* $p_u \geq \sqrt{\alpha_{\min}}$,
  - *the Gap Assumption is satisfied w.r.t* ϵ, *and*
  - *the Membership Assumption is satisfied w.r.t* γ.
- *Then all communities can be recovered in*

$$O\left(n^{C\log(kd/\gamma)/\alpha_{\min}\epsilon^2}\right)$$

time.

The General Case: Intuition

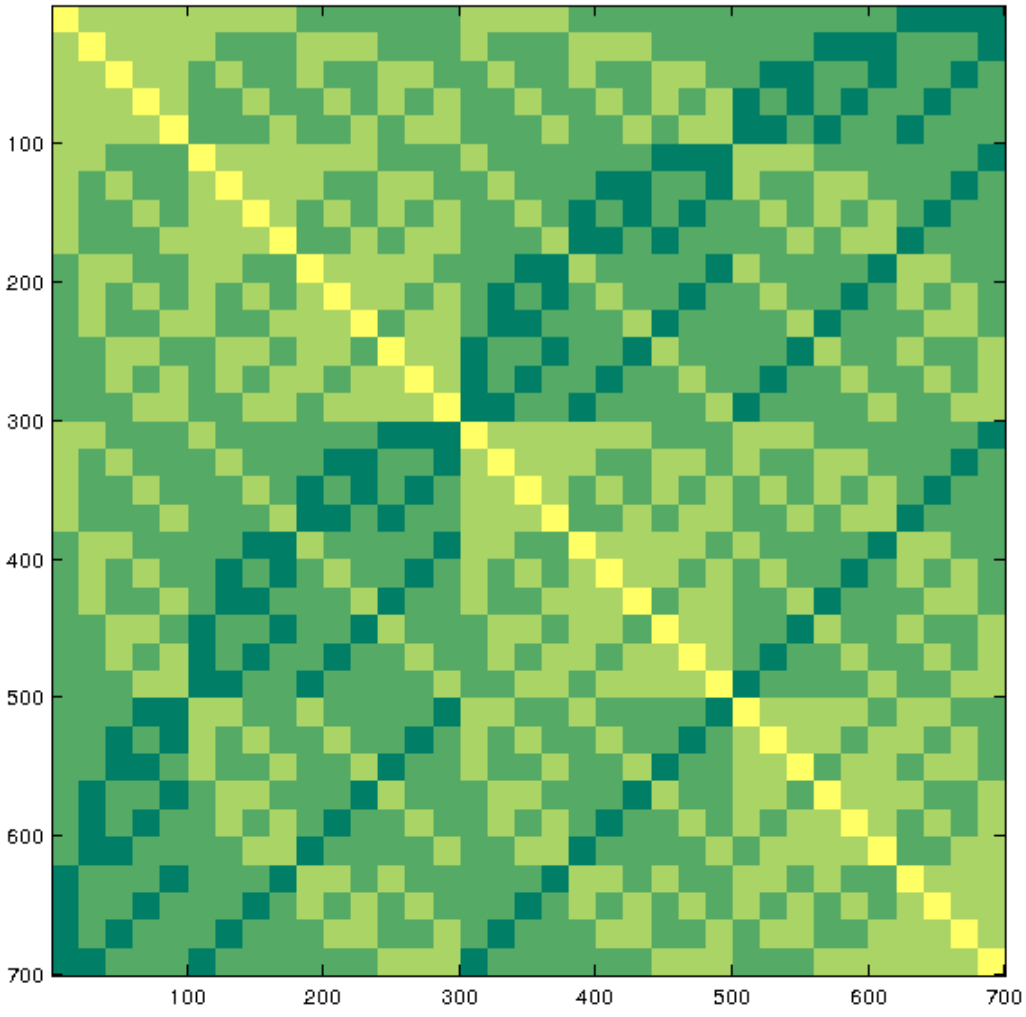
- If communities are very different sizes, random sampling will not find small communities.
- Instead sample ALL subsets of size $T = 100 \log(kd/\gamma)/\alpha_{\min}\epsilon^2$.
- Treat these subsets as the random subsamples in the previous algorithm.
- Since we have to check all subsets of size T, run time is $O(n^{T+C})$.

The Overlapping Community Model

- Assume that network has k communities and n users.
- Every user belongs to exactly d of them.
- The $\binom{k}{d}$ intersections of d communities are all nonempty, with cardinality at least $\hat{n}$.
- If u and v belong to ℓ of the same communities, they are attached by ℓ edges.
- Noise is introduced as up to m random edge additions and deletions between each pair of users.
- Rationale: probability of adding an edge between two users is proportional to the number of communities they share.

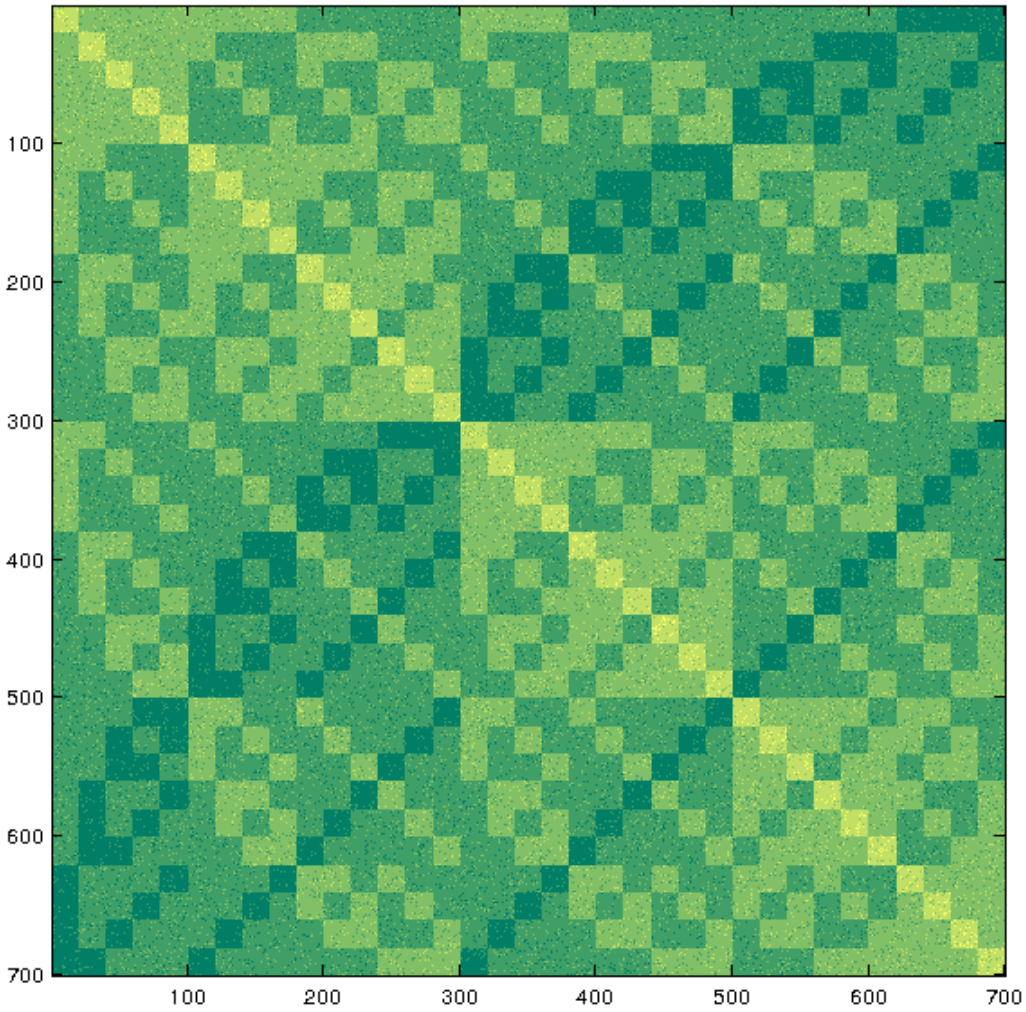
An example

- $k = 7, d = 3, \hat{n} = 20, p = 0.25, q = 0.25$



An example

- $k = 7, d = 3, \hat{n} = 20, p = 0.25, q = 0.25$



What if we knew the intersections of communities?

- The adjacency matrix of the multigraph is a $\binom{k}{d} \times \binom{k}{d}$ block matrix.
- Diagonal blocks correspond to the intersection d communities.
- These should be more denser than other blocks.
- The blocks are disjoint.
- Should be able to find by looking for $\binom{k}{d}$ clusters.

Guaranteed intersection recovery

Theorem

- *Suppose that the multigraph G is sampled from the overlapping community model.*
- *Then the set of intersections of d communities can be recovered w.h.p. from the optimal solution of the semidefinite program*

$$\max \left\{ Tr(AX) : X\mathbf{e} \leq \mathbf{e},\ TrX = \binom{k}{d},\ X \geq 0,\ X \in \Sigma_+^n \right\}$$

in the case that

$$\hat{n} = \Omega(\log n \sqrt{mn}).$$

Algorithm blue print: Contraction

Step (Initial clustering)

Find all intersections of d communities.

Step (Contraction)

- *Obtain the intersection graph G_I as follows.*
- *Contract each intersections of d communities to a single node.*
- *Each pair of nodes is adjacent by an edge with weight equal to the average number edges between the two intersections of communities.*
- *Edge weight is the number of shared communities in the two intersections w.h.p.*

Algorithm blue print: Union step

Step (Union of clusters)

- *Suppose that user u is in communities $S = \{1, 2, ..., d\}$.*
- *Delete any node in G_I with less than $d-1$ edges to $I_{\{1,2,...,d\}}$.*
  - *These nodes do not share $d-1$ communities with S w.h.p..*
  - *Cannot be in the intersection of $d-1$ communities containing u.*
- *Delete $I_{\{1,2,...,d\}}$ as well.*
- *Blocks of nodes sharing the same $d-1$ elements of S form d disjoint clusters in this reduced graph:*

$$I_{\{2,...,d\}}, I_{\{1,3,...,d\}}, \ldots, I_{\{1,2,...,d-1\}}.$$

- *Apply a clustering algorithm to find them.*

Algorithm blue print: Induction

Step

Repeat recursively

- *The union of $I_{\{1,\ldots,d\}}$ and each of the clusters found gives the set of intersections of $d-1$ communities containing u.*
- *Repeat the process to find the sets of intersections of $d-2, d-3, \ldots, 2$ communities containing u.*
- *Repeat one last time to find the k communities containing u.*

Complexity: Generate at most $d!$ graphs to cluster. If d is very small compared to n then the whole algorithm runs in polynomial time (in n, k).

How to cluster?

- How do we find the intermediate clusters/intersections at each step?
- Can our SDP find these clusters?
- Is there an inexpensive way to find these clusters?
- Do we need to solve an SDP? When would an LP suffice?
- Do we only have to compare degrees and common neighbours as in Arora et al.?

Are all branches necessary?

- The inductive step generates $O(d!)$ graphs to cluster (or SDPs to solve).
- Do we need to solve all of them to find all of the communities?
- How to choose which ones? Random sampling?

Different generative models?

- In the overlapping communities model, we assume that it is known how many communities a pair of users share.
- This is unlikely to be true in practice.
- What if we only required the intersections of communities to be more dense than communities and communities to be more dense than noncommunities?
- Can we recover the intersections of communities in this case?
- What if some intersections are empty?