Matching Algorithms in the Stochastic Block Model

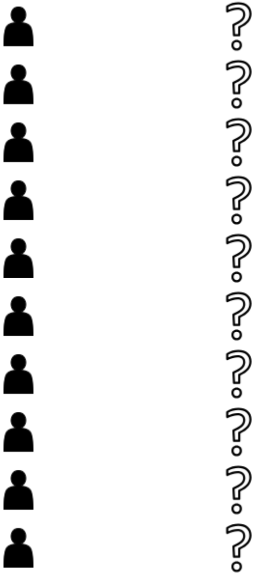
Nathan Sheffield, Divya Shyamal
Mentors: Anna Brandenberger, Byron Chin

MIT SPUR

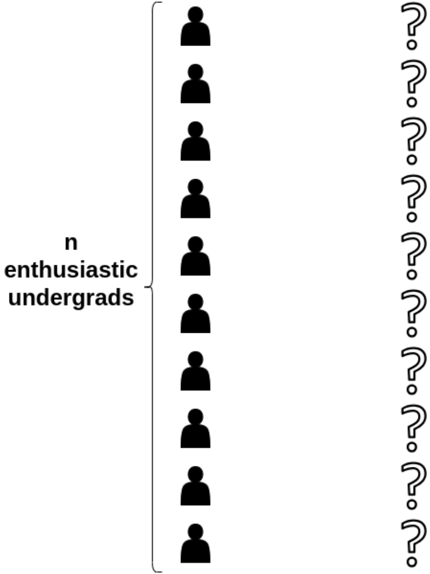
2023

A Story

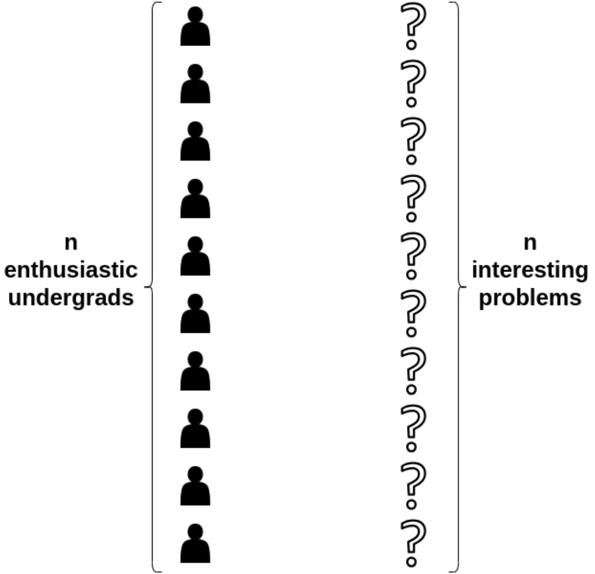
A STORY



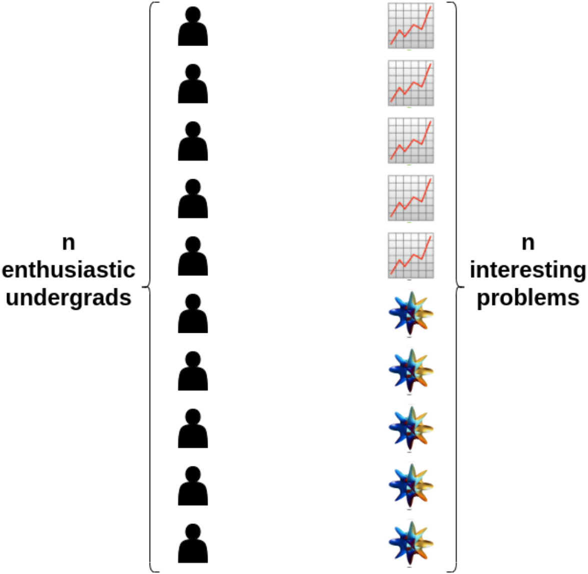
A STORY



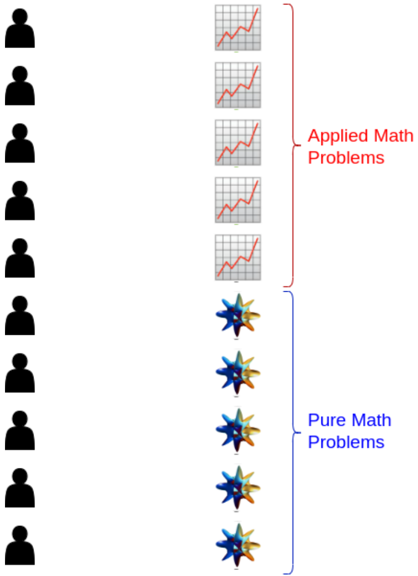
A STORY



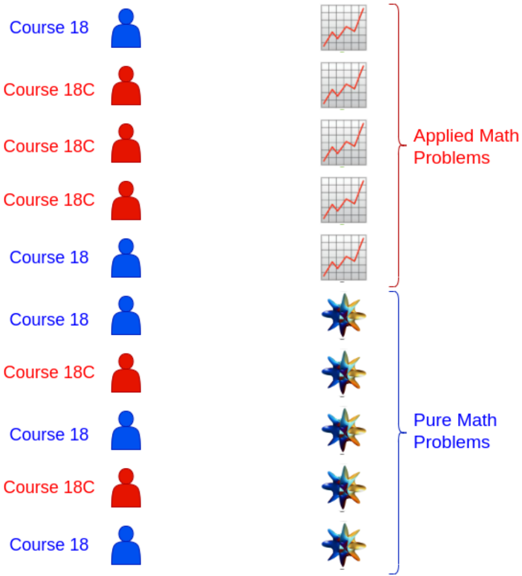
A STORY



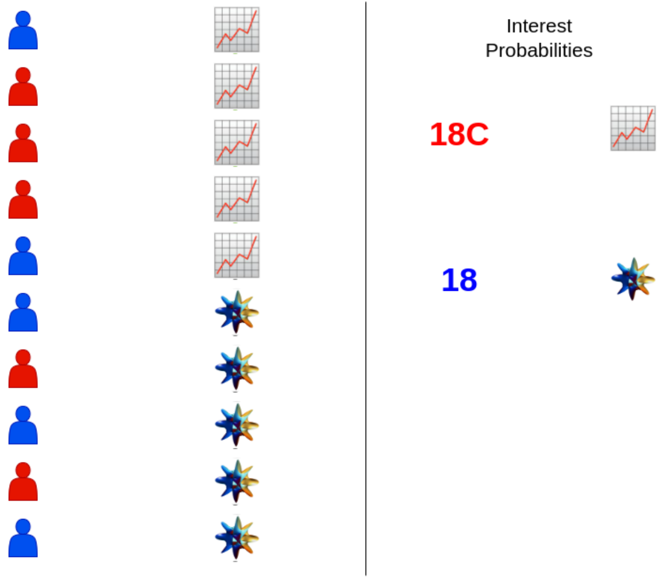
A STORY



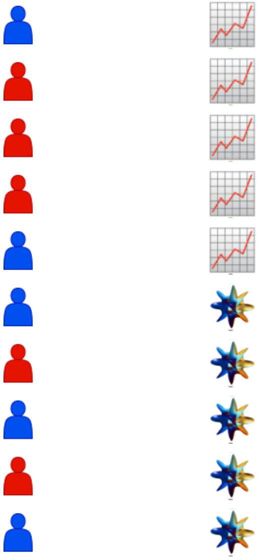
A STORY

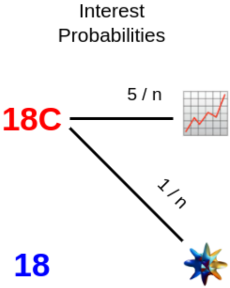

Course 18
Course 18C
Course 18C
Course 18C
Course 18
Course 18
Course 18C
Course 18
Course 18C
Course 18
Applied Math Problems
Pure Math Problems

A STORY



A STORY

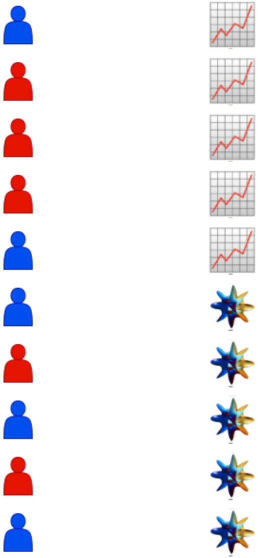


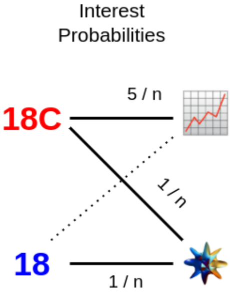


$c_{18C,\ applied} = 5$

$c_{18C,\ pure} = 1$

A STORY





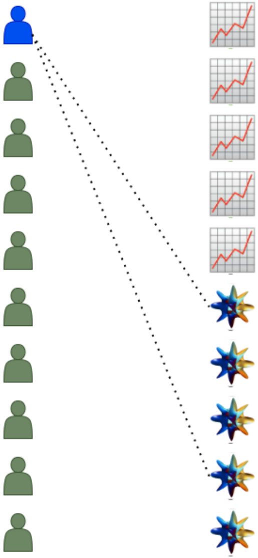
$c_{18C,\ applied} = 5$

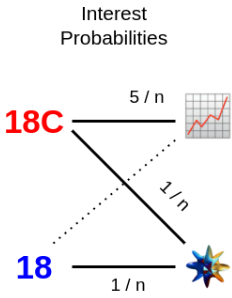
$c_{18C,\ pure} = 1$

$c_{18,\ applied} = 0$

$c_{18,\ pure} = 1$

A STORY





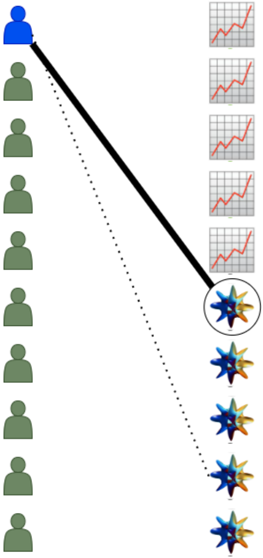
$c_{18C,\ applied} = 5$

$c_{18C,\ pure} = 1$

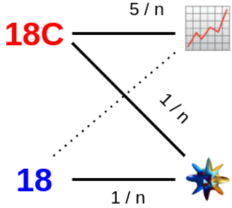
$c_{18,\ applied} = 0$

$c_{18,\ pure} = 1$

A STORY



Interest
Probabilities



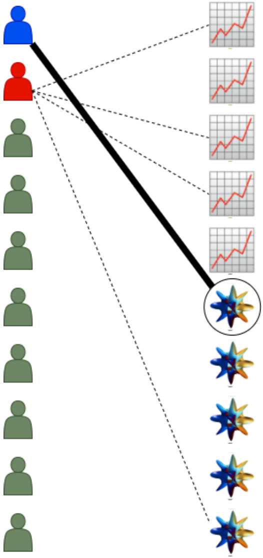
$c_{18C,\ applied} = 5$

$c_{18C,\ pure} = 1$

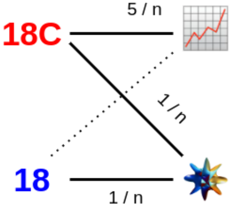
$c_{18,\ applied} = 0$

$c_{18,\ pure} = 1$

A STORY



Interest Probabilities



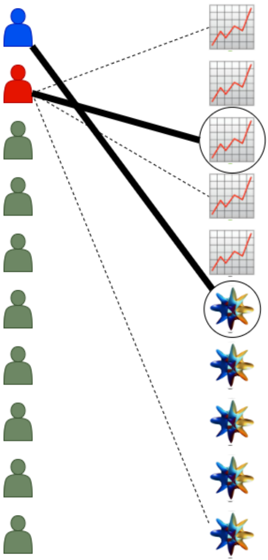
$c_{18C,\,applied} = 5$

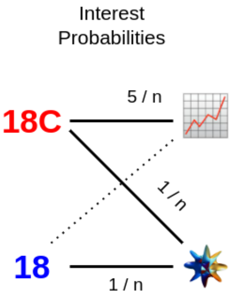
$c_{18C,\,pure} = 1$

$c_{18,\,applied} = 0$

$c_{18,\,pure} = 1$

A STORY





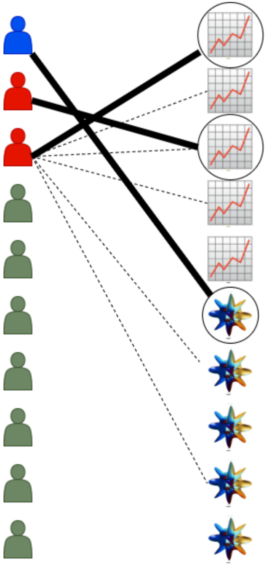
$c_{18C,\ applied} = 5$

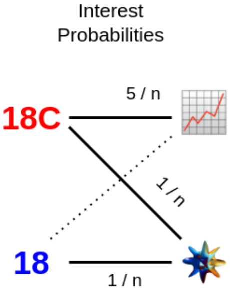
$c_{18C,\ pure} = 1$

$c_{18,\ applied} = 0$

$c_{18,\ pure} = 1$

A STORY





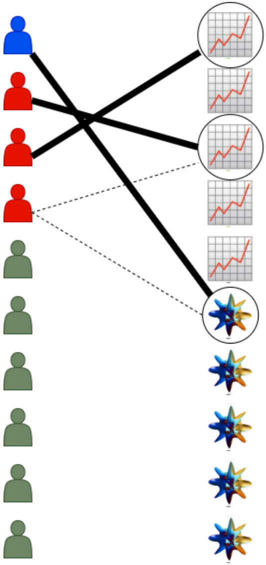
$c_{18C,\ applied} = 5$

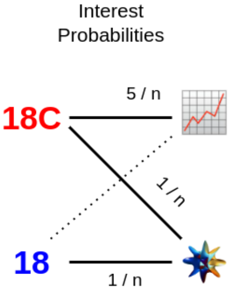
$c_{18C,\ pure} = 1$

$c_{18,\ applied} = 0$

$c_{18,\ pure} = 1$

A STORY





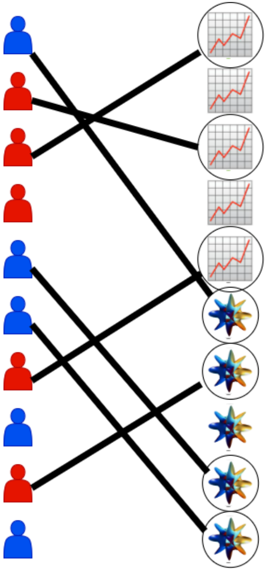
$c_{18C,\ applied} = 5$

$c_{18C,\ pure} = 1$

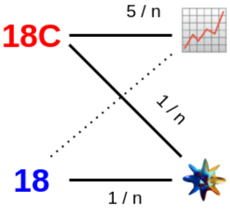
$c_{18,\ applied} = 0$

$c_{18,\ pure} = 1$

A STORY



Interest
Probabilities



$c_{18C,\ applied} = 5$

$c_{18C,\ pure} = 1$

$c_{18,\ applied} = 0$

$c_{18,\ pure} = 1$

Online Matching, Sparse Bipartite SBMs

- n left vertices and n right vertices; q classes on each side
- Edge probabilities depend on classes ($p_{ij} = \frac{c_{ij}}{n}$)
- Online setting: left vertices arrive sequentially. Must decide which right vertex to match it to (if any) before seeing the next left vertex.
- Class of each left vertex is drawn uniformly at random
- Interested in behavior as $n \to \infty$

Bipartite Erdős–Rényi Graph

Theorem (Mastin, Jaillet)

The expected matching size in the online setting on a bipartite Erdős–Rényi graph with $p = \frac{c}{n}$, $n \to \infty$ is

$$\left(1 - \frac{\ln(2 - e^{-c})}{c}\right) n + o(n)$$

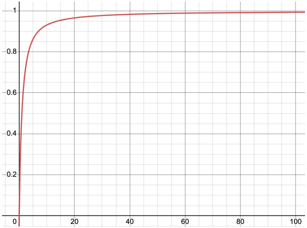


Figure: Expected size of matching vs. c

Algorithm 1: DUMB-GREEDY

- DUMB-GREEDY: choose an available edge uniformly at random to add to the matching
- Optimal in the **equitable** case; achieves the same expected matching size as in the bipartite Erdős–Rényi case with $c =$ avg. degree of each vertex
- **Equitable** case: all vertices have same expected degree

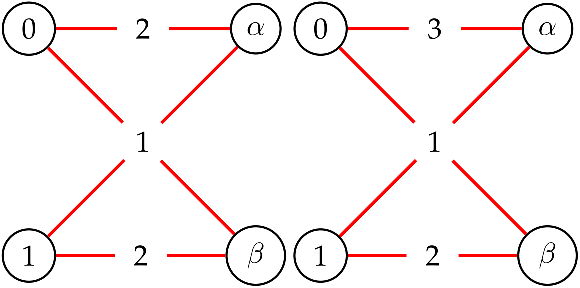


Figure: Equitable and non-equitable examples, $|\alpha| = |\beta| = \frac{n}{2}$

WHERE DUMB-GREEDY FAILS

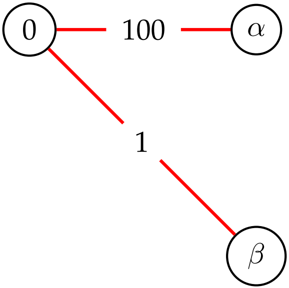


Figure: $|\alpha| = |\beta| = \frac{n}{2}$

A better strategy in this case is to match to β vertices whenever there is one available.

ALGORITHM 2: DEGREEDY

- DEGREEDY: of available right vertices, consider those with lowest expected degree. Choose one uniformly at random to match with.
- "Prefers" classes with lowest expected degree

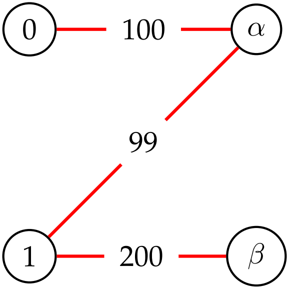


Figure: DEGREEDY suboptimal. $|\alpha| = |\beta| = \frac{n}{2}$

ALGORITHM 3: SHORTSIGHTED

- SHORTSIGHTED: minimize the probability of not matching the next vertex
- $u_\alpha =$ # unmatched α vertices, $u_\beta =$ # unmatched β vertices. Prefer to match to class

$$\operatorname*{argmin}_{c \in \{\alpha,\beta\}} \sum_{i=0}^{1} \frac{1}{2} (1 - p_{i\alpha})^{u_\alpha - \mathbb{1}\{c=\alpha\}} \left(1 - p_{i\beta}\right)^{u_\beta - \mathbb{1}\{c=\beta\}}$$

- SHORTSIGHTED performs very well in practice, but is not optimal

ALGORITHM 4: BRUTE-FORCE

- BRUTE-FORCE:
  - Precompute expected matching size found by algorithm at each "state"
  - At each step, move to the state with highest expected matching size
- BRUTE-FORCE is the optimal online algorithm
- Computationally very expensive, $\Theta(n^{q+1})$

WHERE SHORTSIGHTED FAILS

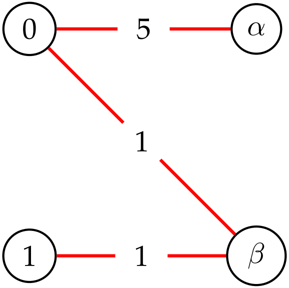


Figure: $|\alpha| = |\beta| = \frac{n}{2}$

SHORTSIGHTED achieves $\approx 0.574946n$, while BRUTE-FORCE achieves at least $\approx 0.575597n$ in expectation (gap of $\geq .0006n$)

Comparison Between SHORTSIGHTED and BRUTE-FORCE

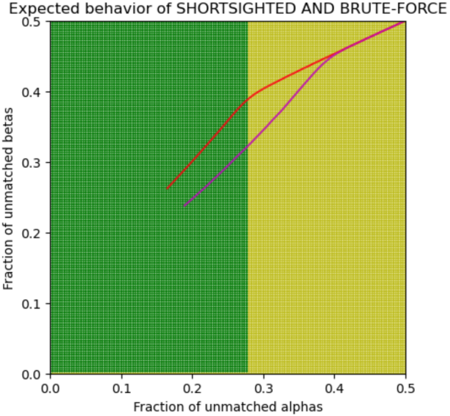
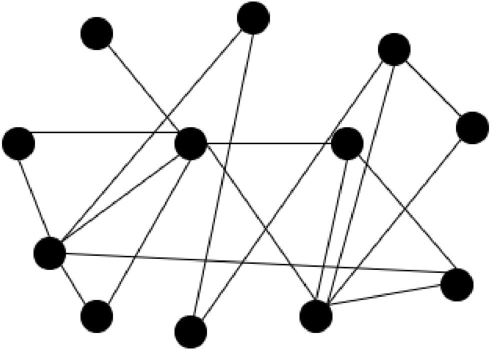


Figure: SHORTSIGHTED represented by red curve, BRUTE-FORCE by magenta. Yellow indicates a preference for class α, green for β in SHORTSIGHTED.

True (Offline) Matching Numbers

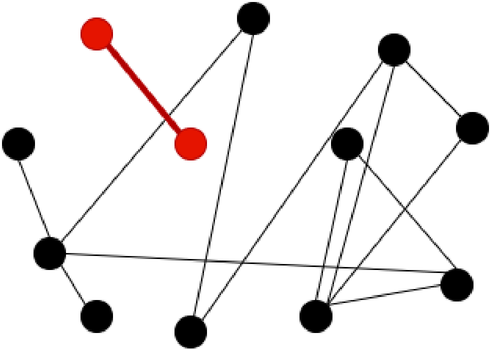
Karp–Sipser Algorithm

- If ∃ vertex of degree 1, add its edge to the matching.



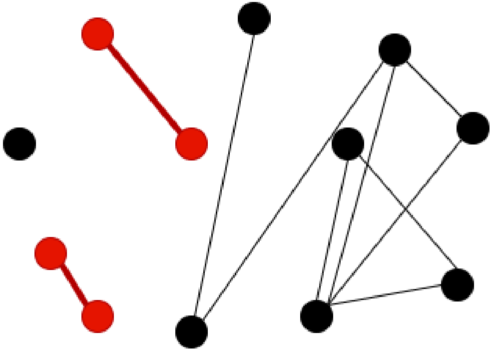
Karp–Sipser Algorithm

- If ∃ vertex of degree 1, choose one at random and add its edge to the matching.



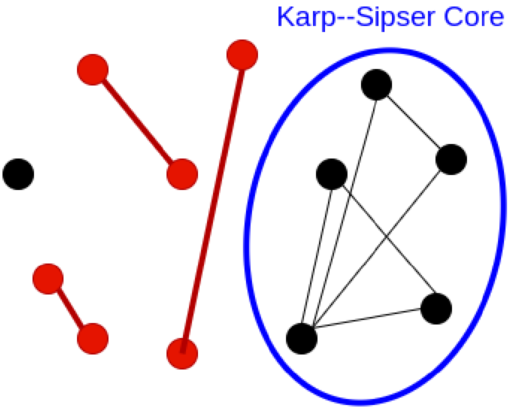
Karp–Sipser Algorithm

- If ∃ vertex of degree 1, choose one at random and add its edge to the matching.



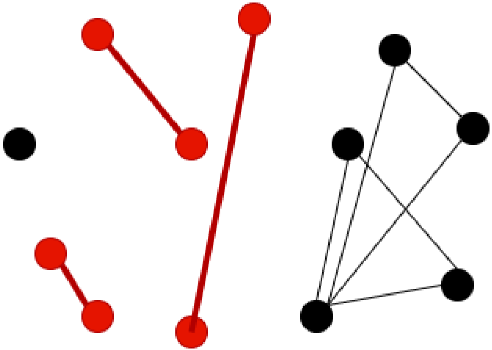
Karp–Sipser Algorithm

- If ∃ vertex of degree 1, choose one at random and add its edge to the matching.



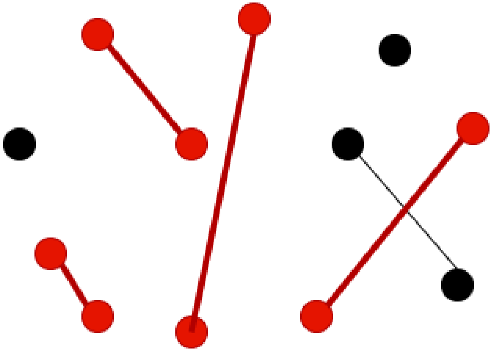
Karp–Sipser Algorithm

- If ∃ vertex of degree 1, choose one at random and add its edge to the matching.
- Otherwise, add any random edge



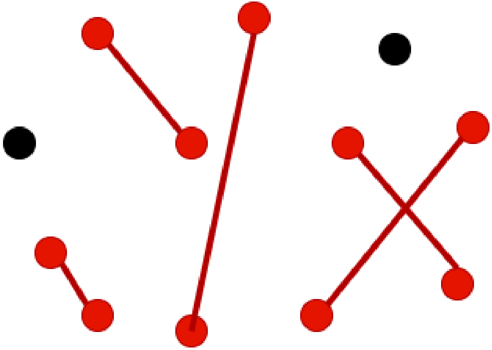
Karp–Sipser Algorithm

- If ∃ vertex of degree 1, choose one at random and add its edge to the matching.
- Otherwise, add any random edge



Karp–Sipser Algorithm

- If ∃ vertex of degree 1, choose one at random and add its edge to the matching.
- Otherwise, add any random edge



Known Results in Erdős–Rényi Case

Optimality

Theorem (Karp, Sipser)

On an Erdős–Rényi graph with edge probability $\frac{c}{n}$, whp Karp–Sipser constructs a matching within $o(n)$ of optimal.

Phase Transition

Theorem (Karp, Sipser)

On an Erdős–Rényi graph with edge probability $\frac{c}{n}$, whp the Karp–Sipser core has size

- $o(n)$ *if* $c < e$
- $\Theta(n)$ *if* $c > e$

Comparisons to SBM Case

Optimality

- The Karp–Sipser is optimal in some specific cases, including the equitable case, which has the same matching number as Erdős–Rényi .
- It is **not** optimal on general SBM graphs

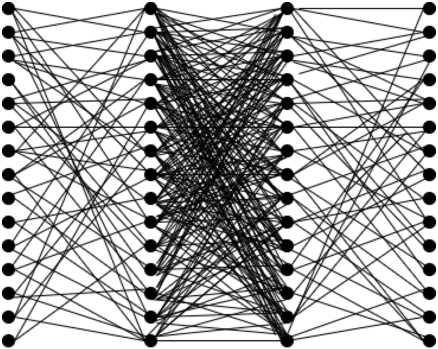
Phase Transition

- The Karp–Sipser core is $o(n)$ whenever average degree is $< e$ in all classes
- It is also $o(n)$ for more cases

Where Karp–Sipser Fails

$$c_{ab} = c_{cd} = 10$$

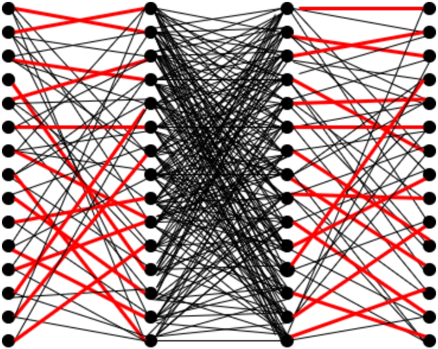
$$c_{bc} = 100$$



Where Karp–Sipser Fails

$$c_{ab} = c_{cd} = 10$$

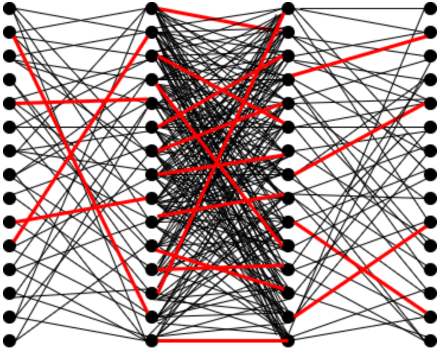
$$c_{bc} = 100$$



Where Karp–Sipser Fails

$$c_{ab} = c_{cd} = 10$$

$$c_{bc} = 100$$



EQUITABLE CASE

Theorem

Karp–Sipser is asympotically optimal on equitable SBM graphs, and finds whp a matching of size

$$\left(1 - \frac{x + ce^{-x} + xce^{-x}}{2c}\right) n + o(n)$$

where c is the average degree, and x is the smallest soln to $x = ce^{-ce^{-x}}$

Corollary

DUMB-GREEDY achieves tight competitive ratio $\frac{c-\ln(2-e^{-c})}{(2c-x+ce^{-x}+xce^{-x})}$ on equitable SBM graphs (minimum over $c \approx .837$)

Critical Threshold

Claim: The Karp–Sipser core is $o(n)$ whenever the expected degree is less than e for every class.

Critical Threshold

Claim: The Karp–Sipser core is $o(n)$ whenever the expected degree is less than e for every class.

Proof Outline:

- Neighbourhood of a vertex in sparse SBM looks like a type of random tree

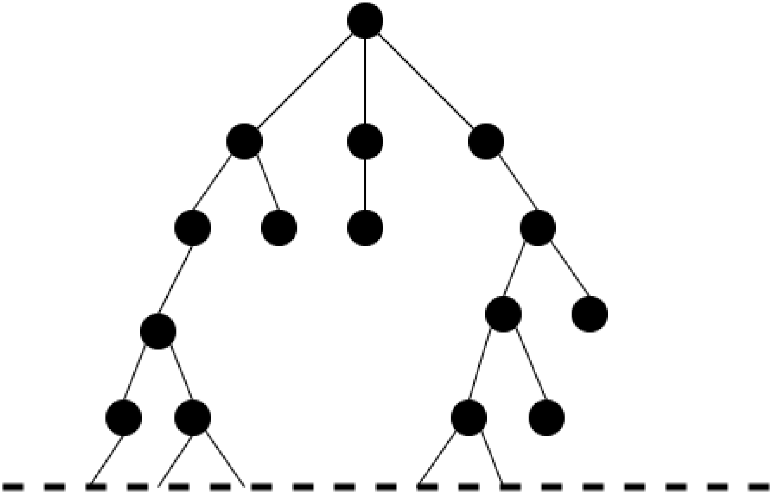
Critical Threshold

Claim: The Karp–Sipser core is $o(n)$ whenever the expected degree is less than e for every class.

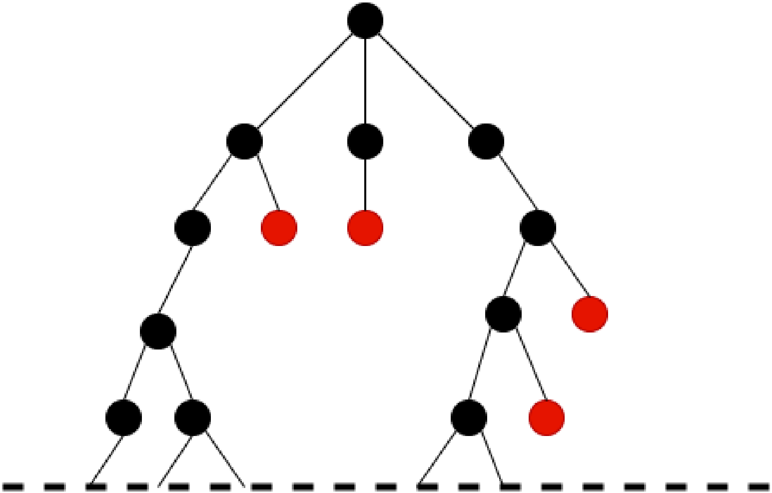
Proof Outline:

- Neighbourhood of a vertex in sparse SBM looks like a type of random tree
- Probability of tree structures near the root allowing Karp–Sipser to remove the root is bounded by the draw probability of a game on the tree

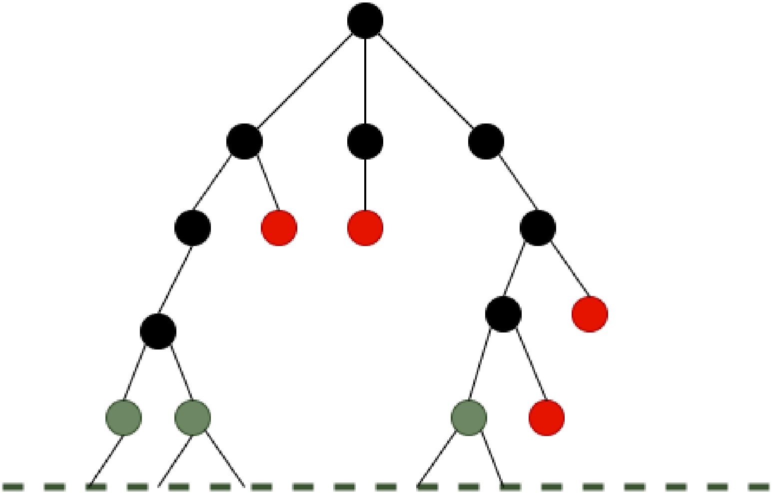
Critical Threshold



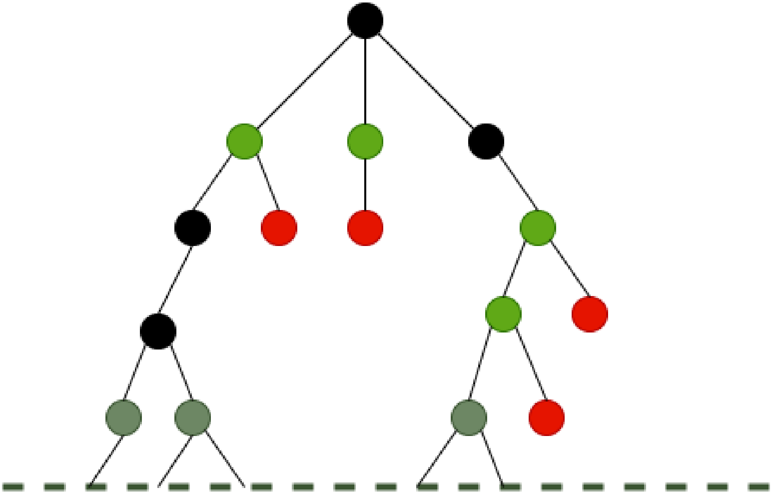
Critical Threshold



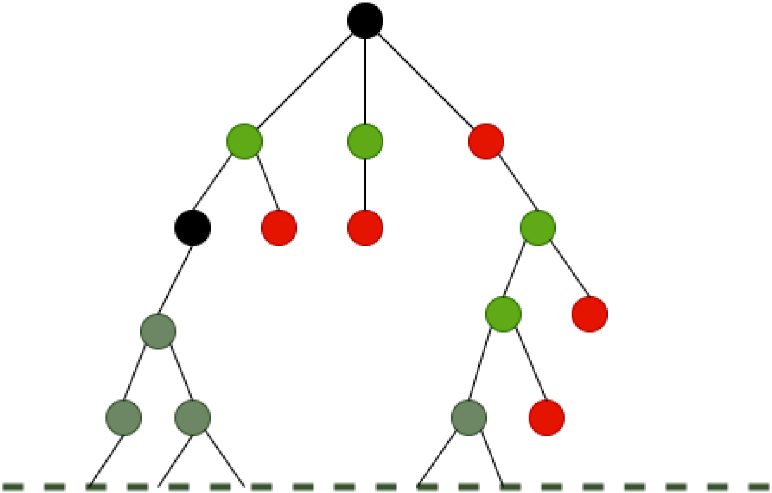
Critical Threshold



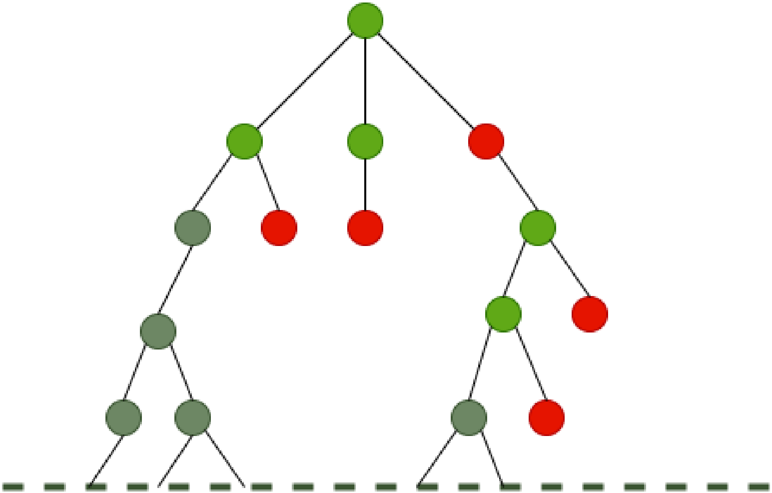
Critical Threshold



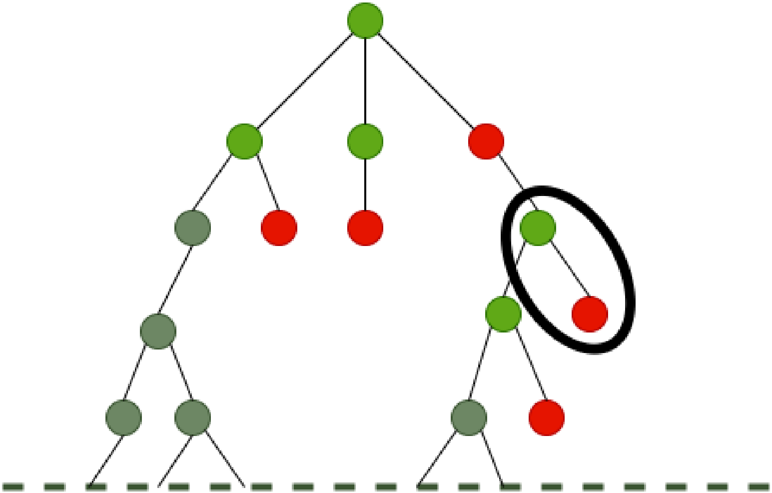
Critical Threshold



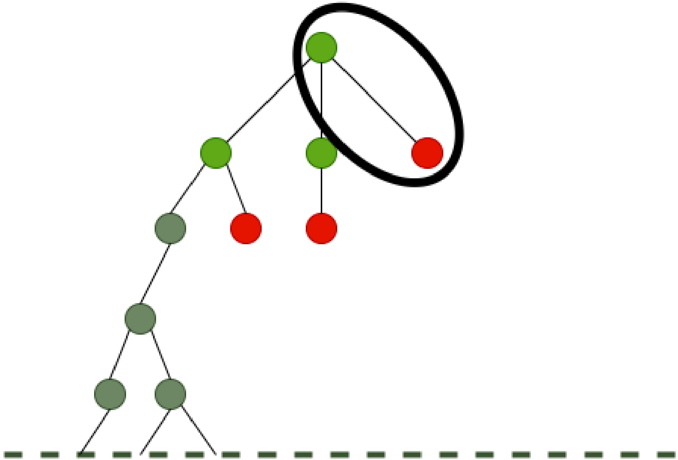
Critical Threshold



Critical Threshold



Critical Threshold



Critical Threshold

Claim: The Karp–Sipser core is $o(n)$ whenever the expected degree is less than e for every class.

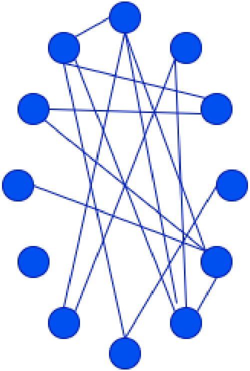
Proof Outline:

- Neighbourhood of a vertex in sparse SBM looks like a type of random tree
- Probability of tree structures near the root allowing Karp–Sipser to remove the root is bounded by the draw probability of a game on the tree
- Draw probability $\to 0$ iff there's exactly one fixed point of

$$\begin{bmatrix} x_1 \\ \dots \\ x_q \end{bmatrix} \mapsto \begin{bmatrix} e^{-\left(\sum_j c_{1j}\bar{S}_j e^{-\left(\sum_k c_{jk}\bar{S}_k x_k\right)}\right)} \\ \dots \\ e^{-\left(\sum_j c_{qj}\bar{S}_j e^{-\left(\sum_k c_{jk}\bar{S}_k x_k\right)}\right)} \end{bmatrix}$$

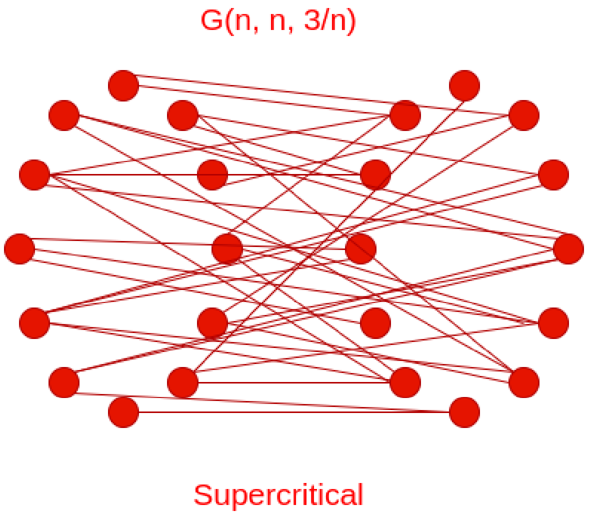
Critical Threshold is Weird

G(n, 3/n)



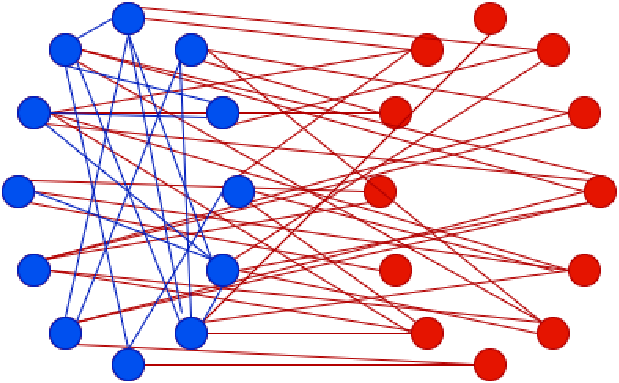
Supercritical

Critical Threshold is Weird



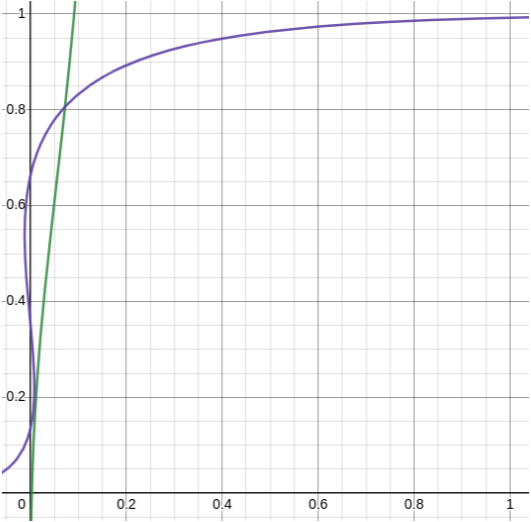
Critical Threshold is Weird

All of the edges from both

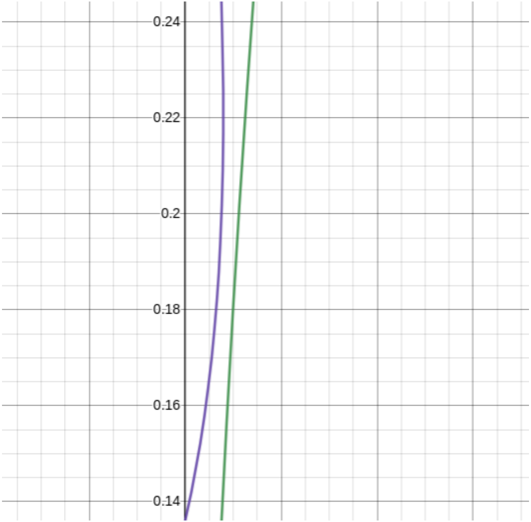


Subcritical

Critical Threshold is Weird



Critical Threshold is Weird



Open Questions

- Is there a simpler characterization of the critical threshold?

Open Questions

- Is there a simpler characterization of the critical threshold?
- Is a label-aware version of Karp–Sipser optimal?

Open Questions

- Is there a simpler characterization of the critical threshold?
- Is a label-aware version of Karp–Sipser optimal?
- Does SHORTSIGHTED always achieve competitive ratio close to BRUTE-FORCE?

Open Questions

- Is there a simpler characterization of the critical threshold?
- Is a label-aware version of Karp–Sipser optimal?
- Does SHORTSIGHTED always achieve competitive ratio close to BRUTE-FORCE?
- Does there exist a linear-time algorithm with the same competitive ratio as BRUTE-FORCE?

Acknowledgements

- Thanks to the organizers of SPUR for putting together such a wonderful program! (And for matching us to an interesting problem :))
- Thanks to our mentors, Anna and Byron, for working with us (and sitting through a lot of baffled chalkboard rambling)
- Thanks to Elchanan Mossel for suggesting the problem topic