Matrix Queries for Solving Linear Algebra, Statistics, and Graph Problems

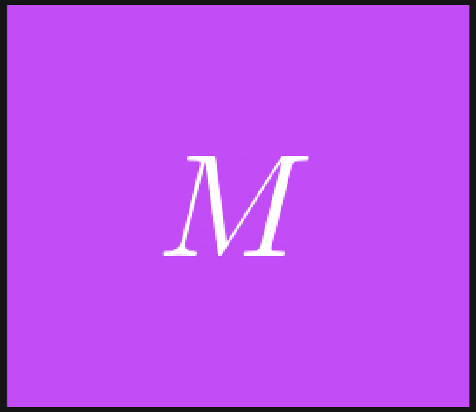
Cyrus Rashtchian

UCSD

12/02/20

Joint work w/ David P. Woodruff, Peng Ye, Hanlin Zhu

Vector Matrix Vector Queries



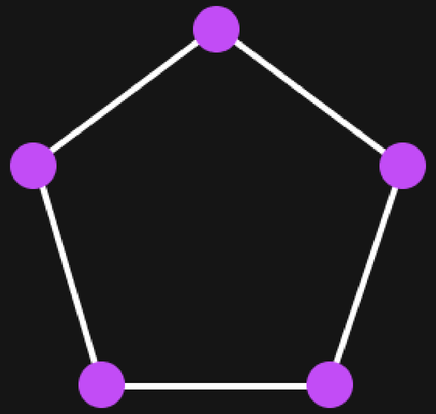
$n \times n$

Vector Matrix Vector Queries

$u, v \in \mathbb{F}^n$

$u^\top M v$

graph queries



Measure number of queries to solve a problem

Randomized, adaptive, approximation algorithms

Bounded entries and $O(\log n)$ bit complexity

Small space streaming algorithm

$(\#\text{queries}) \cdot O(\log n)$

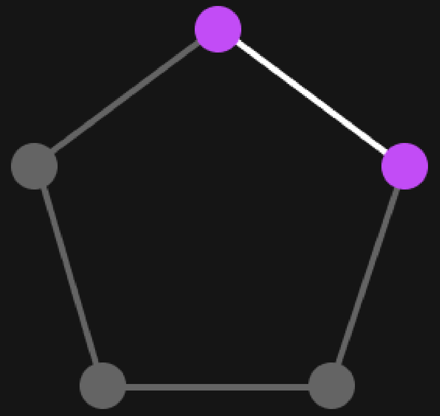
Vector Matrix Vector Queries

$$u^\top = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad M = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & \boxed{1} & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad v = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$u, v \in \mathbb{F}^n$

$$u^\top M v$$

edge-probe query



Measure number of queries to solve a problem

Randomized, adaptive, approximation algorithms

Bounded entries and $O(\log n)$ bit complexity

Small space streaming algorithm

$(\#\text{queries}) \cdot O(\log n)$

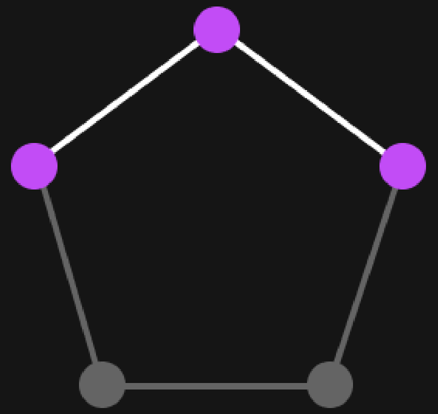
Vector Matrix Vector Queries

$$u^\top = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad M = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad v = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$u, v \in \mathbb{F}^n$

$$u^\top M v$$

degree query



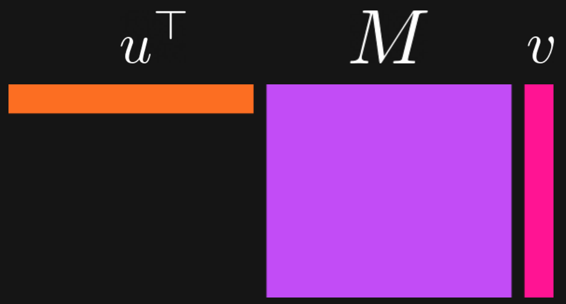
Measure number of queries to solve a problem

Randomized, adaptive, approximation algorithms

Bounded entries and $O(\log n)$ bit complexity

Small space streaming algorithm
$(\#\text{queries}) \cdot O(\log n)$

Vector Matrix Vector Queries



Unifies previous models

- edge/degree queries
- sample random edge with $O(\log n)$ queries

All of it can be implemented
in the uMv model
with negligible overhead

Lots and lots of work on sublinear time algorithms for graph problems…

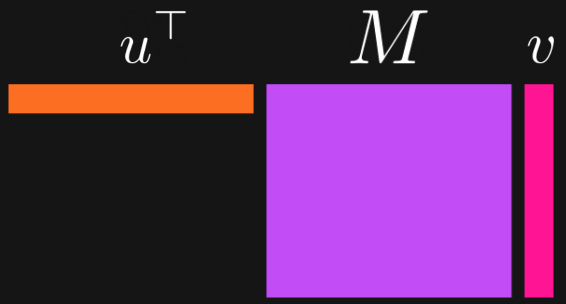
[Feige '04; Goldreich, Ron '04]

…

[Eden, Levi, Ron, Seshadhri '15; Eden, Ron, Seshadhri '18]

[Eden, Ron, Rosenbaum '19; Assadi, Kapralov, Khanna '19]

Vector Matrix Vector Queries



Unifies previous models

- edge/degree queries
- sample random edge with $O(\log n)$ queries
- edge count queries

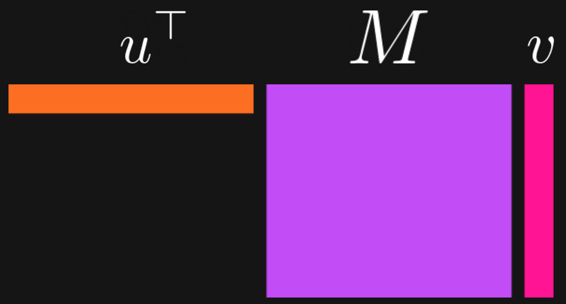
$$\mathbf{1}_A^\top M \mathbf{1}_B$$

number of edges between $A, B \subseteq V_G$

All of it can be implemented
in the uMv model
with negligible overhead

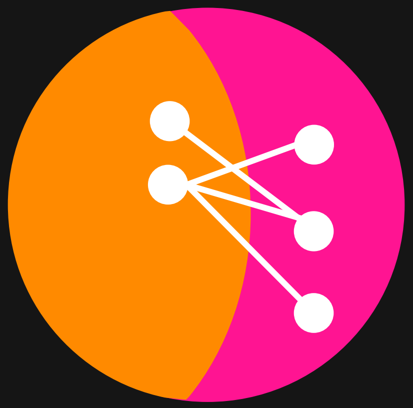
[Alon, Asodi '04; Angluin, Chen '06; Reyzin, Srivastava '07]

Vector Matrix Vector Queries



Unifies previous models

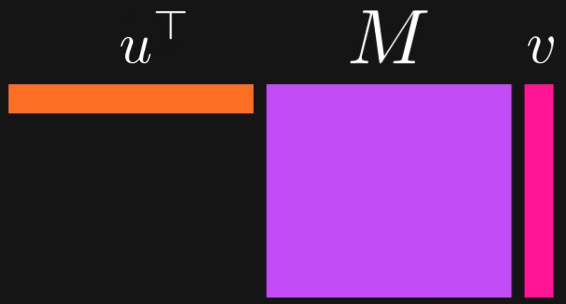
- edge/degree queries
- sample random edge with $O(\log n)$ queries
- edge count queries
- cut queries



[Rubinstein, Schramm, Weinberg '18]

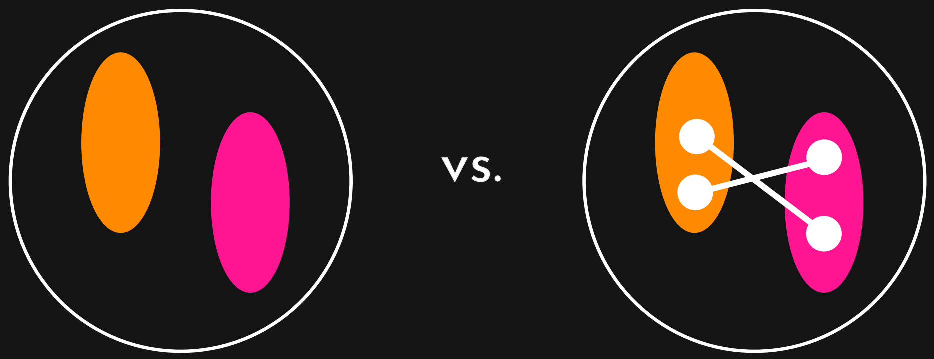
All of it can be implemented
in the uMv model
with negligible overhead

Vector Matrix Vector Queries



Unifies previous models

- edge/degree queries
- sample random edge with $O(\log n)$ queries
- edge count queries
- cut queries
- (bipartite) independent set queries



All of it can be implemented
in the uMv model
with negligible overhead

[Beame, Har-Peled, Natarajan Ramamoorthy, R., Sinha '18]
[Chen, Levi, Waingarten '19]

Vector Matrix Vector Queries

Unifies previous models

- edge/degree queries
- sample random edge with $O(\log n)$ queries
- edge count queries
- cut queries
- (bipartite) independent set queries

Specializes other models

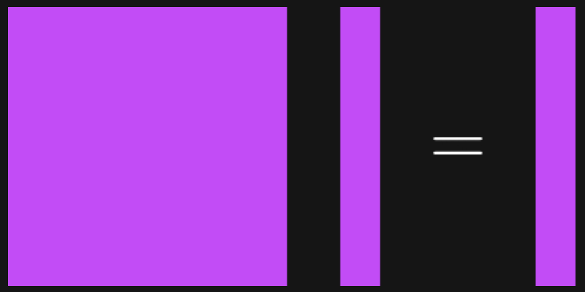
Vector Matrix Vector Queries

Unifies previous models

- edge/degree queries
- sample random edge with $O(\log n)$ queries
- edge count queries
- cut queries
- (bipartite) independent set queries

Specializes other models

- Matrix vector queries



Mv

[Sun, Woodruff, Yang, Zhang '19]

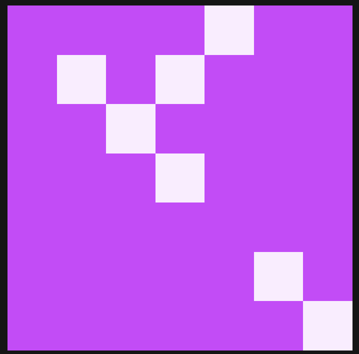
Vector Matrix Vector Queries

Unifies previous models

- edge/degree queries
- sample random edge with $O(\log n)$ queries
- edge count queries
- cut queries
- (bipartite) independent set queries

Specializes other models

- Matrix vector queries
- Linear sketching



$\mathbb{R}$ or $\mathbb{F}_2$

$v^\top \text{vec}(M)$

$v \in \mathbb{F}^{n^2}$

Agenda

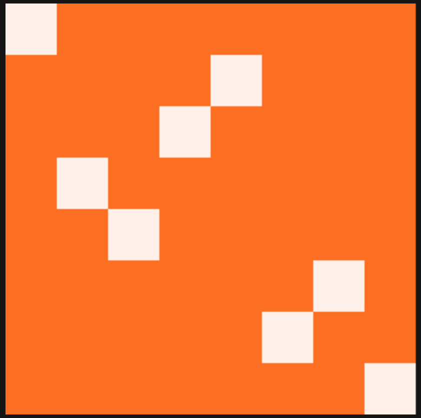
Focus on uMv and edge-probe queries

Permutation Matrices

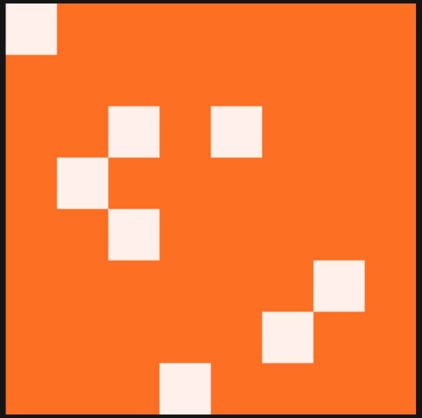
Planted Clique

Other Results & Open Questions

Permutation Matrices



vs.



Constant success probability

Assume binary matrix $\{0,1\}^{n\times n}$

Assume n is even

Theorem: $O(1)$ queries over $\mathbb{R}$

$\Omega(n)$ queries over $\mathbb{F}_2$

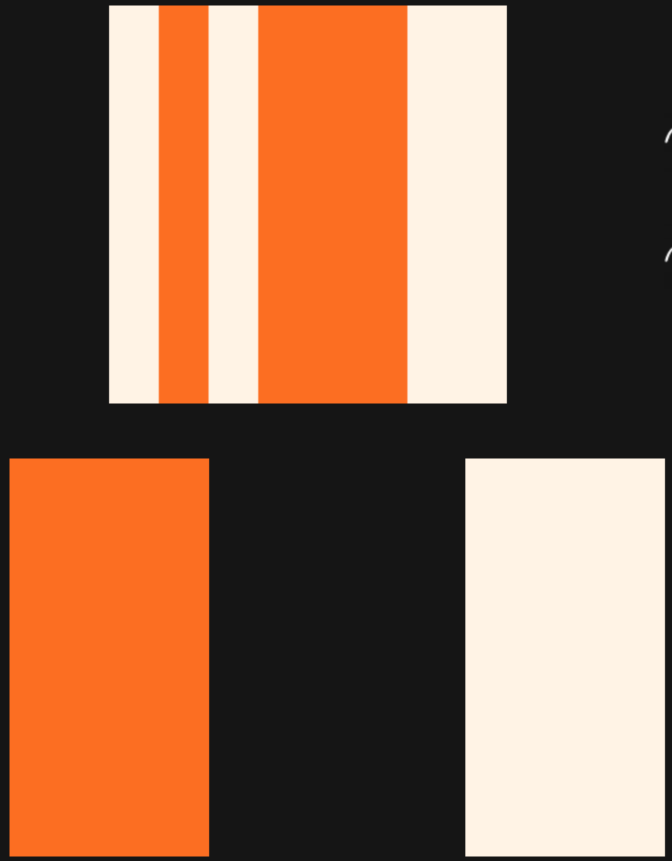
in uMv or Linear Sketching models

Permutation Matrices

Test permutation over $\mathbb{R}$ with $O(1)$ queries

Algorithm:

- Choose subset of $\frac{n}{2}$ columns
- Check # ones
- Reject if not $\frac{n}{2}$
- Else, repeat



$u = \mathbf{1}$

$v = \mathbf{1}_A$

$|A| = \frac{n}{2}$

$[n] \setminus A$ $\quad$ A

Permutation Matrices

Test permutation over $\mathbb{R}$ with $O(1)$ queries

Algorithm:

Choose subset of $\frac{n}{2}$ columns

Check # ones

Reject if not $\frac{n}{2}$

Else, repeat

Run algorithm for rows & cols

$u = \mathbf{1}$

$v = \mathbf{1}_A$

$|A| = \frac{n}{2}$

$[n] \setminus A$ $\qquad$ A

Permutation Matrices

Test permutation over $\mathbb{R}$ with $O(1)$ queries

Algorithm:

Choose subset of $\frac{n}{2}$ columns

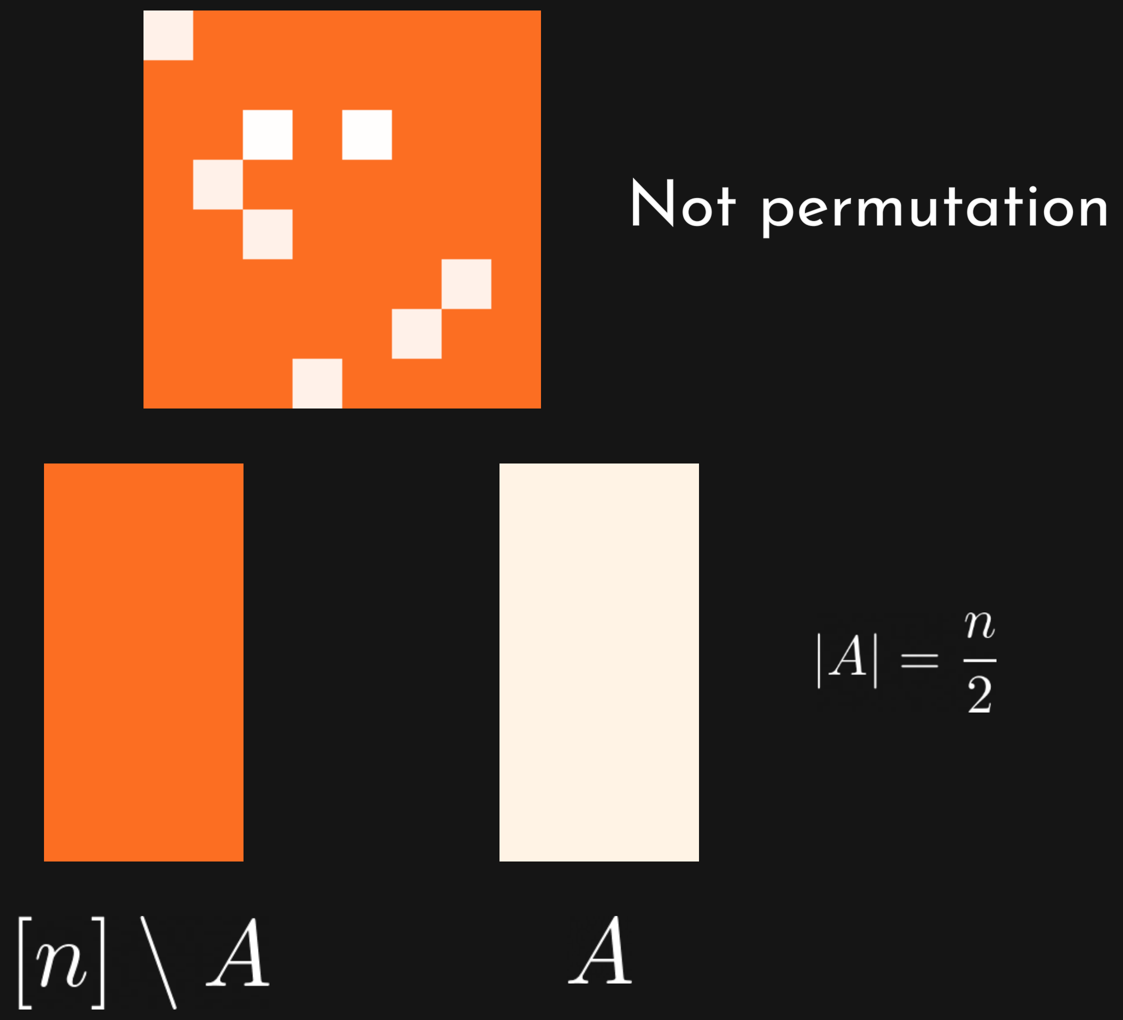
Check # ones

Reject if not $\frac{n}{2}$

Else, repeat

Claim: with const probability,

see sum other than $\frac{n}{2}$



Permutation Matrices

Test permutation over $\mathbb{R}$ with $O(1)$ queries

Algorithm:

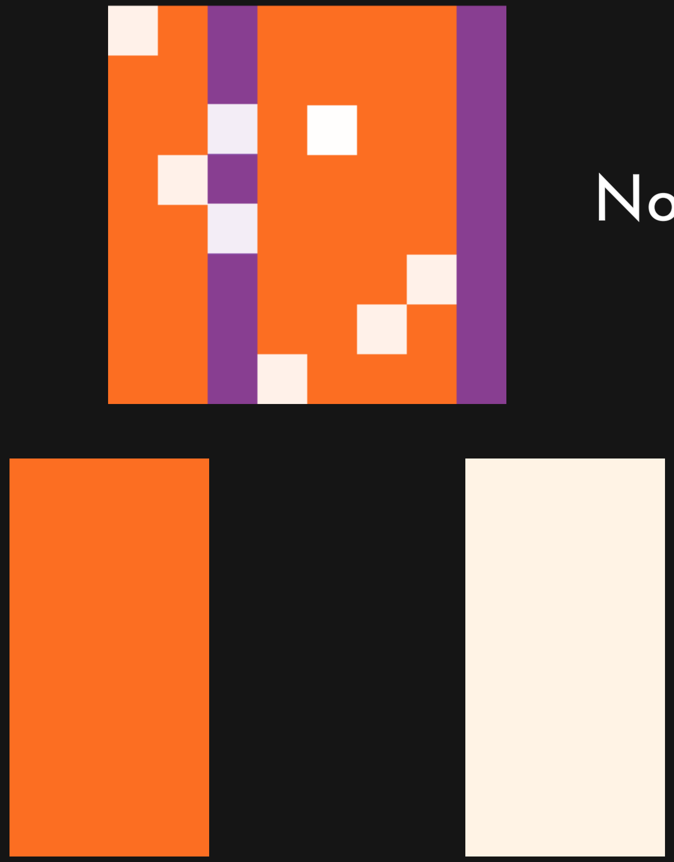
Choose subset of $\frac{n}{2}$ columns

Check # ones

Reject if not $\frac{n}{2}$

Else, repeat

Claim: with const probability, see sum other than $\frac{n}{2}$



Not permutation

$|A| = \frac{n}{2}$

$[n] \setminus A$ $\quad$ A

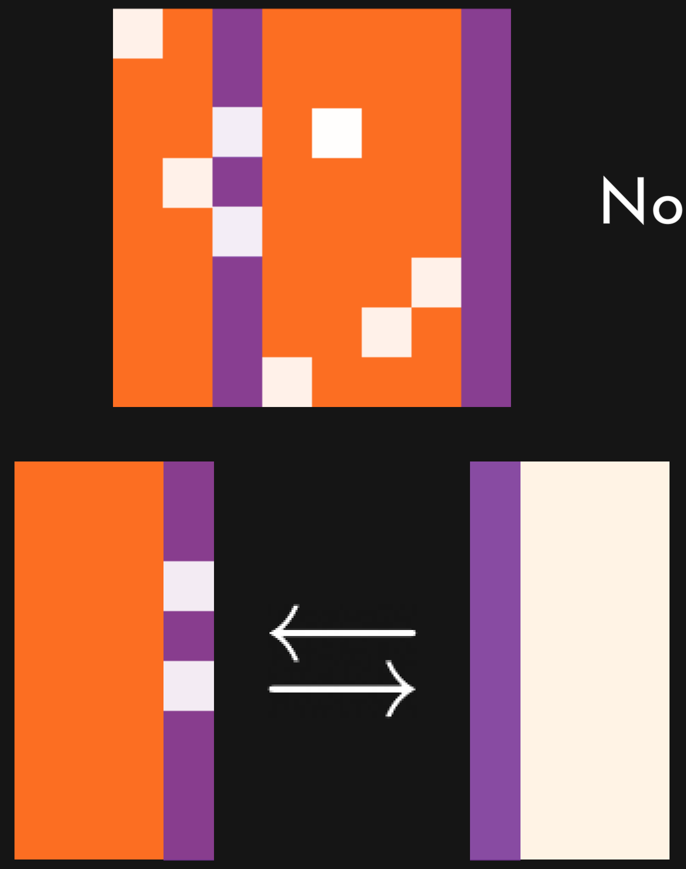
Permutation Matrices

Test permutation over $\mathbb{R}$ with $O(1)$ queries

Algorithm:

- Choose subset of $\frac{n}{2}$ columns
- Check # ones
- Reject if not $\frac{n}{2}$
- Else, repeat

Claim: with const probability, see sum other than $\frac{n}{2}$



Permutation Matrices

Test permutation over $\mathbb{R}$ with $O(1)$ queries

Permutation $\Longleftrightarrow$ Perfect Matching

Same idea works for Doubly Stochastic Matrices

Test if a matrix is diagonal also $O(1)$ queries (random vectors, zero diagonal)

Permutation Matrices

Test permutation requires $\Omega(n)$ queries over $\mathbb{F}_2$

Proof sketch in the video, with animations

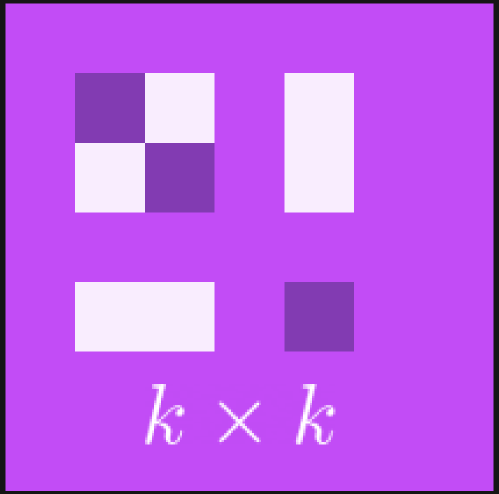
Idea: communication complexity

- reduce to disjointness (2-player)
- Alice & Bob build matrices based on their strings
- Using 3 x 3 gadgets → XOR of matrices permutation iff disjoint

Planted Clique

Random instance (null hypothesis) $G(n, 0.5)$

Planted instance (alternate hypothesis) $G(n, 0.5, k)$



null: w.h.p. max clique $\leq 2 \log n$

determine null vs. alternate:

possible to detect $k \geq (2 + o(1)) \log n$

unknown if polytime $k = o(\sqrt{n})$

easy & efficient $k \gg \sqrt{n}$

Edge Probe Prior Results

Theorem [Rácz, Schiffer '19]: $\tilde{\Theta}\left(\frac{n^2}{k^2}\right)$ edge-probe queries are necessary and sufficient for detecting or finding a planted k-clique

Algorithm:

1. Sample $\gg \dfrac{n \log n}{k}$ vertices uniformly at random; query all pairs
2. Check if there is a clique of size $\geq 3 \log n$ induced by sampled vertices
3. If so, claim there is a planted k-clique; otherwise, claim the graph is random

Recall: in $G(n, 0.5)$ largest clique $\leq (2 + o(1)) \log n$ w.h.p.

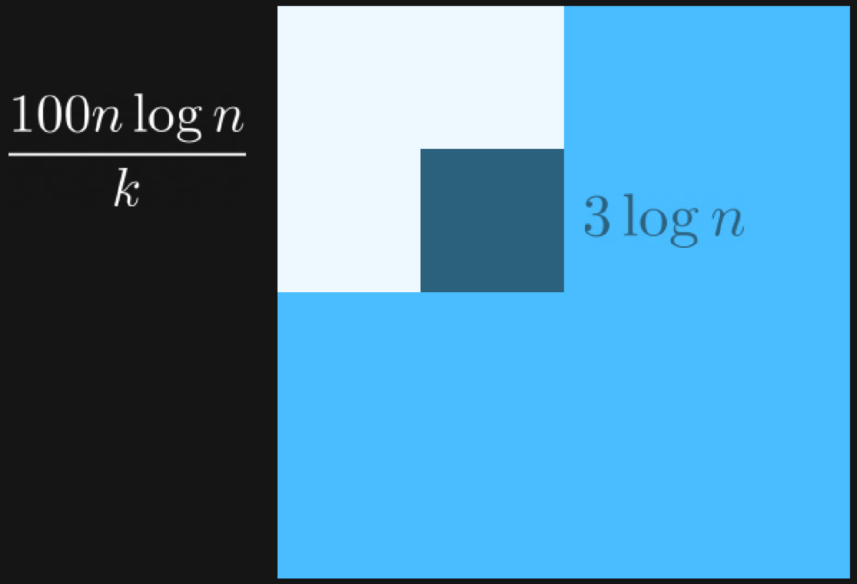
Edge Probe Prior Results

Theorem [Rácz, Schiffer '19]: $\tilde{\Theta}\left(\frac{n^2}{k^2}\right)$ edge-probe queries are necessary and sufficient for detecting or finding a planted k-clique

Algorithm:

1. Sample $\gg \frac{n \log n}{k}$ vertices uniformly at random
2. Check if there is a clique of size $\geq 3 \log n$
3. If so, claim there is a planted k-clique

Recall: in $G(n, 0.5)$ largest clique $\leq (2 + o(1)) \log n$ w.h.p.



Clique Decomposition

Provide an alternate proof of lower bound via communication complexity

Assign edges to Alice and Bob, decomposing graph into random subsets of k-cliques

Lemma [Conlon, Fox, Sudakov '12]: if $k = o(\sqrt{n})$ then the complete n-graph contains $\tilde{\Theta}\left(\frac{n^2}{k^2}\right)$ edge-disjoint k-cliques, covering $\Omega(n^2)$ edges

Prior work: similar ideas for MAX-Clique, but we must preserve the distribution

[Halldórsson, Sun, Szegedy, Wang '12]
[Braverman, Liu, Singh, Vinodchandran, Yang '18]

Edge Probe Lower Bound

Theorem: if $k = o(\sqrt{n})$, then $\Omega\left(\frac{n^2}{k^2}\right)$ edge-probe queries are necessary to detect planted k-clique with constant success probability

We prove a communication lower bound of $\Omega\left(\frac{n^2}{k^2}\right)$ for solving a related "PC Game"

Alice and Bob get matrices, where actual graph will be $G = G_1 \oplus G_2$

Decide if graph random or planted k-clique with const. prob.

Reduce to UDISJ (planted clique iff sets intersect)

either unique index such that $x_i = y_i = 1$

or one of $x_i = 0$ or $y_i = 0 \quad \forall\, i$

UDISJ to PC Game

Start with UDISJ instance, inputs $\mathbf{x}, \mathbf{y} \in \{0,1\}^\ell$ where $\ell = \Theta(n^2/k^2)$

Alice and Bob detect planted clique on XOR of adj. matrices $G = G_1 \oplus G_2$

Use clique decomposition lemma $K_k^1, K_k^2, \ldots, K_k^\ell$

Alice gets edges in G_1^i and Bob gets edges in G_2^i

Randomly 4-color edges in K_k^i

- $\boldsymbol{x}_i = 0 \implies$ add all edges in K_k^i with colors 1 or 3 to G_1^i
- $\boldsymbol{x}_i = 1 \implies$ add all edges in K_k^i with colors 1 or 2 to G_1^i
- $\boldsymbol{y}_i = 0 \implies$ add all edges in K_k^i with colors 1 or 4 to G_2^i
- $\boldsymbol{y}_i = 1 \implies$ add all edges in K_k^i with colors 3 or 4 to G_2^i

1010

1100

1001

0011

Claim: using this reduction:
not disjoint leads to $G(n, 0.5, k)$
disjoint leads to $G(n, 0.5)$

UDISJ to PC Game

Start with UDISJ instance, inputs $\mathbf{x}, \mathbf{y} \in \{0,1\}^\ell$ where $\ell = \Theta(n^2/k^2)$

Alice and Bob detect planted clique on XOR of adj. matrices $G = G_1 \oplus G_2$

Use clique decomposition lemma $K_k^1, K_k^2, \ldots, K_k^\ell$

Alice gets edges in G_1^i and Bob gets edges in G_2^i

Randomly 4-color edges in K_k^i

Claim: using this reduction:
not disjoint leads to $G(n, 0.5, k)$
disjoint leads to $G(n, 0.5)$

Theorem: if $k = o(\sqrt{n})$, then $\Omega\left(\frac{n^2}{k^2}\right)$ edge-probe queries are necessary to detect planted k-clique with constant success probability

Also holds for $\mathbb{F}_2$ linear sketching queries (but proof fails for uMv queries…)

uMv and Linear Sketching Lower Bound

Corollary: if $k = o(\sqrt{n})$, then $\tilde{\Omega}\left(\frac{n^2}{k^4}\right)$ uMv or linear sketching queries are necessary to detect planted k-clique with constant success probability

Theorem: if $k = o(\sqrt{n})$, then $\Omega\left(\frac{n^2}{k^2}\right)$ bits necessary to solve the $\binom{k}{2}$-player planted k-clique communication game with constant success probability

Implies query lower bound via natural simulation

$\Theta(k^2 \log n)$ bits communication for each query

$\tilde{\Omega}\left(\frac{n^2}{k^4}\right)$ queries are necessary for uMv or Linear Sketching

Multi-party Communication LB

Theorem: if $k = o(\sqrt{n})$, then $\Omega\left(\frac{n^2}{k^2}\right)$ bits necessary to solve the $\binom{k}{2}$-player planted k-clique communication game with constant success probability

Use standard direct sum framework for information complexity [BYJKS '04]

Combine ideas from distributed data processing inequality [Braverman, Garg, Ma, Nguyen, Woodruff '16]

Still use the clique decomposition lemma

But now each player gets at most one edge from each clique

Essentially multiplayer unique disjointness with appropriate input distribution

Multi-party Communication LB

Theorem: if $k = o(\sqrt{n})$, then $\Omega\left(\frac{n^2}{k^2}\right)$ bits necessary to solve the $\binom{k}{2}$-player planted k-clique communication game with constant success probability

Multiparty game

$\binom{k}{2}$

Input

$\Theta\left(\frac{n^2}{k^2}\right)$

Random row $\mathbf{X}_j$
prob 1/2 all ones
prob 1/2 random

Output: determine if there is a planted all ones row or not

Information lower bound:

$$\text{comm} \geq I(\mathbf{X}; \Pi(\mathbf{X}) \mid \text{not planted})$$

$$\geq \sum_{i=1}^{\ell} I(\mathbf{X}_j; \Pi(\mathbf{X}) \mid \text{not planted})$$

$$\geq \Omega(\ell) = \Omega(n^2/k^2)$$

Other results

Linear Algebra Problems		
Schatten p-norm	$\Omega(\sqrt{n})$ for $p \in [0,4)$, const. factor approx. over $\mathbb{R}$	Theorem 3.2
	$\Omega(n^{1-2/p})$ for $p \geq 4$, const. factor approx. over $\mathbb{R}$	Theorem 3.2
Rank testing	$\Omega(k^2)$ to distinguish rank k vs. $k+1$ over $\mathbb{F}_p$	Theorem 3.3
	$\Omega(n^{2-O(\varepsilon)})$ for $(1 \pm \varepsilon)$ approx. over $\mathbb{R}$, non-adaptive	Theorem 3.4
Trace estimation	$\Omega(n/\log n)$ and $O(n)$ for entries in $\{0, 1, 2, \ldots, n^3\}$	Theorem 3.5
Diagonal matrix	$O(1)$	Theorem 3.6
Symmetric matrix	$O(1)$	Theorem 3.7
Unitary matrix	$\Omega(n/\log n)$ and $O(n)$ for randomized queries over $\mathbb{C}$	Theorem 3.8
	$\Omega(n^2/\log n)$ for deterministic queries over $\mathbb{C}$	Theorem 3.9
Statistics Problems		
All ones column	$\Omega(n/\log n)$ and $O(n)$ over $\mathbb{R}$	Section 4.1
Two identical columns	$\Omega(n)$ and $O(n \log n)$ over $\mathbb{F}_2$	Section 4.2
	$O(n)$ over $\mathbb{R}$	Theorem 4.3
Column-wise majority	$\Theta(n^2)$ over $\mathbb{F}_2$	Theorem 4.4
Permutation matrix	$O(1)$ over $\mathbb{R}$	Theorem 4.5
	$\Omega(n)$ over $\mathbb{F}_2$	Theorem 4.6
Doubly stochastic matrix	$O(1)$ over $\mathbb{R}$	Theorem 4.7
Negative entry detection	$\Omega(n^2/\log n)$ over $\mathbb{R}$	Theorem 4.8
Graph Problems		
Triangle detection	$\Omega(n^2/\log n)$	Theorem 5.1
Star graph	$O(1)$ over $\mathbb{R}$	Theorem 5.2

Open Questions

1. Planted clique with uMv queries? $\widetilde{O}\left(\frac{n^2}{k^2}\right)$ vs. $\widetilde{\Omega}\left(\frac{n^2}{k^4}\right)$

2. Improve upon graph queries, e.g., triangle/clique approx. counting
   [Eden, Levi, Ron, Seshadhri '15; Assadi, Kapralov, Khanna '19]

3. Test whether matrix is PSD under eigenvalue assumptions
   [Bakshi, Chepurko, Jayaram '20]

4. Generalize to higher rank measurement matrices, query returns $\mathrm{trace}(U^T M)$

5. Generalize uMv to k-tensors with k query vectors (Quantum? Hypergraphs?)

6. More average-case reductions, e.g., stochastic block model
   [Brennan, Bresler '20; Brennan, Bresler, Huleihel '18]

7. Connections to fine-grained complexity
   [Dell, Lapinkas '17; Dell, Lapiskas, Meeks '19]

Thanks!

Cyrus Rashtchian

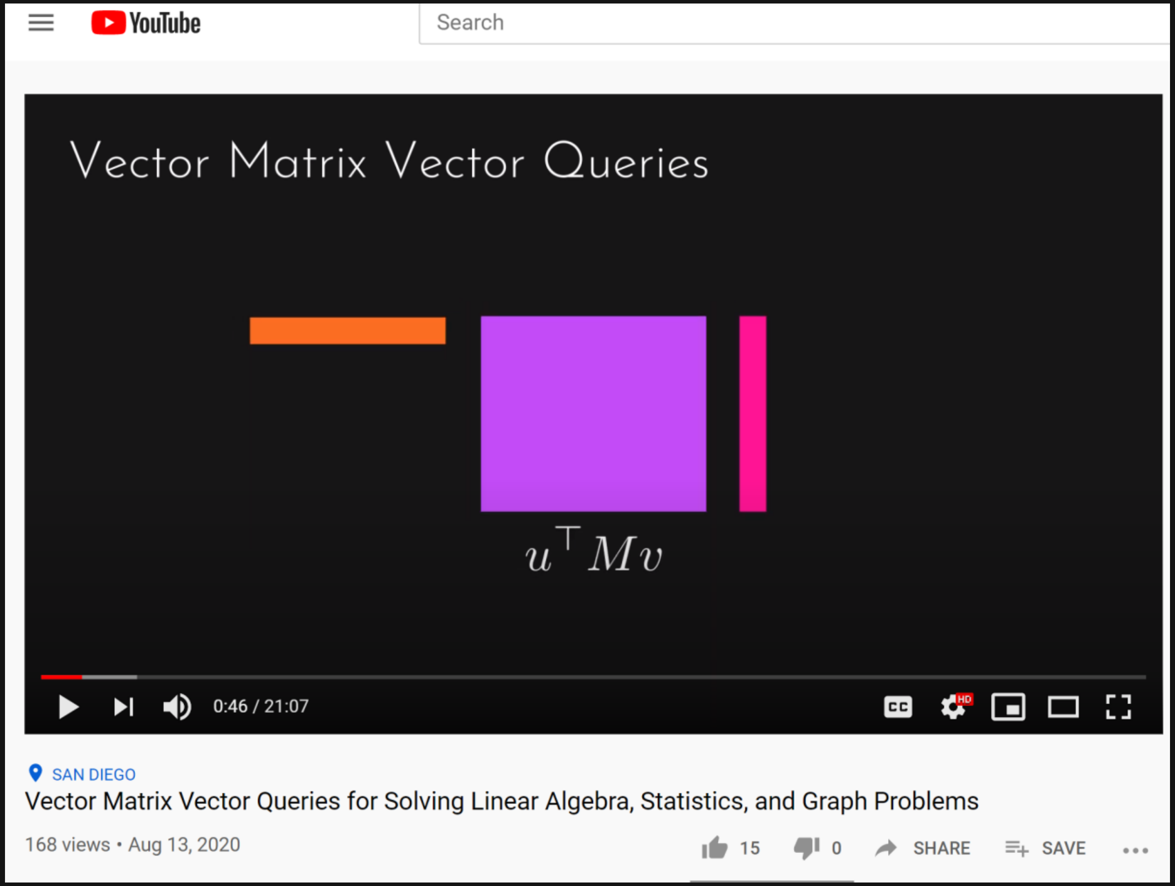
www.cyrusrashtchian.com

UCSD

ML blog: ucsdml.github.io

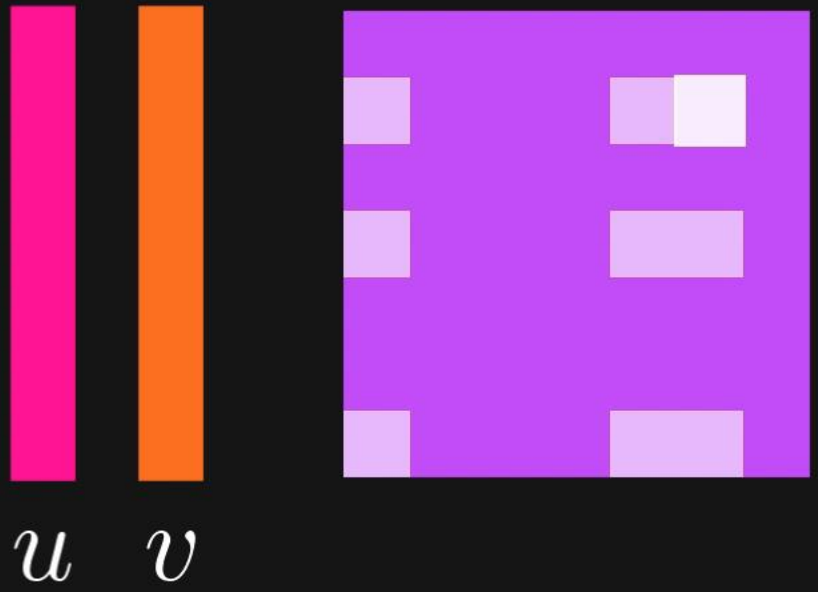
crashtchian@eng.ucsd.edu

@CyrusRashtchian



Diagonal Matrices

Test whether a matrix is diagonal with O(1) queries



Repeat many times

$$u^\top M v \stackrel{?}{=} 0$$

Choose random vectors such that diagonal always zero $\quad (u_i v_i = 0)$