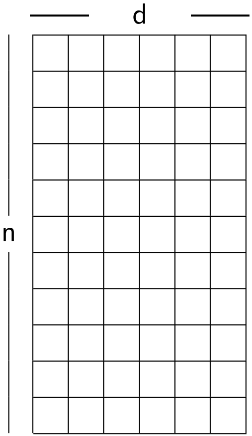
OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings

Jelani Nelson
Harvard

September 27, 2013

based on joint work with Huy L. Nguyễn (Princeton)

$A =$



- $A \in \mathbb{R}^{n \times d}$, $n \gg d$, $rank(A) = r$
- tall, skinny matrix

Numerical linear algebra

- $A \in \mathbb{R}^{n \times d}$, $n \gg d$, $rank(A) = r$

Numerical linear algebra

- $A \in \mathbb{R}^{n \times d}$, $n \gg d$, $rank(A) = r$

Classical numerical linear algebra problems

- Compute the **leverage scores** of A, i.e. the ℓ_2 norms of the n standard basis vectors when projected onto the subspace spanned by the columns of A.

Numerical linear algebra

- $A \in \mathbb{R}^{n \times d}$, $n \gg d$, $rank(A) = r$

Classical numerical linear algebra problems

- Compute the **leverage scores** of A, i.e. the ℓ_2 norms of the n standard basis vectors when projected onto the subspace spanned by the columns of A.
- **Least squares regression**: Given also $b \in \mathbb{R}^n$.

$$\text{Compute } x^* = \text{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_2$$

Numerical linear algebra

- $A \in \mathbb{R}^{n \times d}$, $n \gg d$, $rank(A) = r$

Classical numerical linear algebra problems

- Compute the **leverage scores** of A, i.e. the ℓ_2 norms of the n standard basis vectors when projected onto the subspace spanned by the columns of A.
- **Least squares regression**: Given also $b \in \mathbb{R}^n$.

$$\text{Compute } x^* = \text{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_2$$

- ℓ_p **regression** $(p \in [1, \infty))$:

$$\text{Compute } x^* = \text{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_p$$

Numerical linear algebra

- $A \in \mathbb{R}^{n \times d}$, $n \gg d$, $rank(A) = r$

Classical numerical linear algebra problems

- Compute the **leverage scores** of A, i.e. the ℓ_2 norms of the n standard basis vectors when projected onto the subspace spanned by the columns of A.
- **Least squares regression**: Given also $b \in \mathbb{R}^n$.

$$\text{Compute } x^* = \operatorname{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_2$$

- ℓ_p **regression** ($p \in [1, \infty)$):

$$\text{Compute } x^* = \operatorname{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_p$$

- **Low-rank approximation**: Given also an integer $1 \leq k \leq d$.

$$\text{Compute } A_k = \operatorname{argmin}_{\text{rank}(B) \leq k} \|A - B\|_F$$

Numerical linear algebra

- $A \in \mathbb{R}^{n \times d}$, $n \gg d$, $rank(A) = r$

Classical numerical linear algebra problems

- Compute the **leverage scores** of A, i.e. the ℓ_2 norms of the n standard basis vectors when projected onto the subspace spanned by the columns of A.
- **Least squares regression**: Given also $b \in \mathbb{R}^n$.
$$\text{Compute } x^* = \text{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_2$$
- ℓ_p **regression** $(p \in [1, \infty))$:
$$\text{Compute } x^* = \text{argmin}_{x \in \mathbb{R}^d} \|Ax - b\|_p$$
- **Low-rank approximation**: Given also an integer $1 \leq k \leq d$.
$$\text{Compute } A_k = \text{argmin}_{\text{rank}(B) \leq k} \|A - B\|_F$$
- **Preconditioning**: Compute $R \in \mathbb{R}^{d \times d}$ (for $d = r$) so that
$$\forall x \ \|ARx\|_2 \approx \|x\|_2$$

Computationally efficient solutions

Singular Value Decomposition

Theorem

Every matrix $A \in \mathbb{R}^{n \times d}$ of rank r can be written as

$$A = \underbrace{U}_{\substack{\text{orthonorm}\\\text{columns}\\ n\times r}} \quad \underbrace{\Sigma}_{\substack{\text{diagonal}\\\text{positive definite}\\ r\times r}} \quad \underbrace{V^T}_{\substack{\text{orthonorm}\\\text{columns}\\ d\times r}}$$

Can compute SVD in $\tilde{O}(nd^{\omega-1})$ [Demmel, Dumitriu, Holtz, 2007].
$\omega < 2.373\ldots$ is the exponent of square matrix multiplication [Coppersmith, Winograd, 1987], [Stothers, 2010], [Vassilevska-Williams, 2012]

Computationally efficient solutions

$$A = \underbrace{U}_{\substack{\text{orthonorm}\\ \text{columns}\\ n\times r}} \underbrace{\Sigma}_{\substack{\text{diagonal}\\ \text{positive definite}\\ r\times r}} \underbrace{V^T}_{\substack{\text{orthonorm}\\ \text{columns}\\ d\times r}}$$

- **Leverage scores**: Output row norms of U.
- **Least squares regression**: Output $V\Sigma^{-1}U^Tb$.
- **Low-rank approximation**: Output $U\Sigma_k V^T$.
- **Preconditioning**: Output $R = V\Sigma^{-1}$.

Computationally efficient solutions

$$A = \underbrace{U}_{\substack{\text{orthonorm} \\ \text{columns} \\ n\times r}} \quad \underbrace{\Sigma}_{\substack{\text{diagonal} \\ \text{positive definite} \\ r\times r}} \quad \underbrace{V^T}_{\substack{\text{orthonorm} \\ \text{columns} \\ d\times r}}$$

- **Leverage scores**: Output row norms of U.
- **Least squares regression**: Output $V\Sigma^{-1}U^T b$.
- **Low-rank approximation**: Output $U\Sigma_k V^T$.
- **Preconditioning**: Output $R = V\Sigma^{-1}$.

Conclusion: In time $\tilde{O}(nd^{\omega-1})$ we can compute the SVD then solve all the previously stated problems. Is there a faster way?

Subspace embeddings

[Sarlós, 2006]

Let $V \subseteq \mathbb{R}^n$ be a linear subspace of dimension d. A *subspace embedding* for V is a matrix $\Pi \in \mathbb{R}^{m \times n}$ so that

$$\forall x \in V,\ (1-\varepsilon)\|x\| \le \|\Pi x\| \le (1+\varepsilon)\|x\|$$

Subspace embeddings

[Sarlós, 2006]

Let $V \subseteq \mathbb{R}^n$ be a linear subspace of dimension d. A *subspace embedding* for V is a matrix $\Pi \in \mathbb{R}^{m \times n}$ so that

$$\forall x \in V, \ (1-\varepsilon)\|x\| \leq \|\Pi x\| \leq (1+\varepsilon)\|x\|$$

Subspace embeddings can be used to speed up algorithms for all five problems previously listed [Sarlós, 2006], [Dasgupta, Drineas, Harb, Kumar, Mahoney, 2008], [Clarkson, Woodruff, 2009], [Drineas, Magdon-Ismail, Mahoney, Woodruff, 2012], [Clarkson, Woodruff, 2013], [Clarkson, Drineas, Magdon-Ismail, Mahoney, Meng, Woodruff, 2013], [Woodruff, Zhang, 2013].

How to use subspace embeddings

Least squares regression: Let Π be a subspace embedding for the subspace spanned by b and the columns of A. Let $x^* = \operatorname{argmin} \|Ax - b\|$ and $\tilde{x} = \operatorname{argmin} \|\Pi Ax - \Pi b\|$. Then

How to use subspace embeddings

Least squares regression: Let Π be a subspace embedding for the subspace spanned by b and the columns of A. Let $x^* = \operatorname{argmin} \|Ax - b\|$ and $\tilde{x} = \operatorname{argmin} \|\Pi Ax - \Pi b\|$. Then

$$\|\Pi A\tilde{x} - \Pi b\| \leq \|\Pi Ax^* - \Pi b\|$$

How to use subspace embeddings

Least squares regression: Let Π be a subspace embedding for the subspace spanned by b and the columns of A. Let $x^* = \mathrm{argmin}\,\|Ax - b\|$ and $\tilde{x} = \mathrm{argmin}\,\|\Pi Ax - \Pi b\|$. Then

$$(1-\varepsilon)\|A\tilde{x}-b\| \leq \underbrace{\|\Pi A\tilde{x} - \Pi b\|}_{\|\Pi(A\tilde{x}-b)\|} \leq \|\Pi Ax^*-\Pi b\|$$

How to use subspace embeddings

Least squares regression: Let Π be a subspace embedding for the subspace spanned by b and the columns of A. Let $x^* = \operatorname{argmin} \|Ax - b\|$ and $\tilde{x} = \operatorname{argmin} \|\Pi Ax - \Pi b\|$. Then

$$(1-\varepsilon)\|A\tilde{x}-b\| \leq \|\Pi A\tilde{x}-\Pi b\| \leq \|\Pi Ax^*-\Pi b\| \leq (1+\varepsilon)\|Ax^*-b\|$$

$$\Rightarrow \|A\tilde{x} - b\| \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right) \cdot \|Ax^* - b\|$$

Computational gain from subspace embeddings

Computing SVD of ΠA takes time $\tilde{O}(md^{\omega-1})$, which is much faster than $\tilde{O}(nd^{\omega-1})$ if $m \ll n$.

Computational gain from subspace embeddings

Computing SVD of ΠA takes time $\tilde{O}(md^{\omega-1})$, which is much faster than $\tilde{O}(nd^{\omega-1})$ if $m \ll n$.

Good news: Known that if Π is, say, a random Gaussian matrix with $m = O(d/\varepsilon^2)$, it will be a subspace embedding with high probability [Gordon, 1988], [Klartag, Mendelson, 2005], [Arora, Hazan, Kale, 2006], [Clarkson, Woodruff, 2013].

Computational gain from subspace embeddings

Computing SVD of ΠA takes time $\tilde{O}(md^{\omega-1})$, which is much faster than $\tilde{O}(nd^{\omega-1})$ if $m \ll n$.

Good news: Known that if Π is, say, a random Gaussian matrix with $m = O(d/\varepsilon^2)$, it will be a subspace embedding with high probability [Gordon, 1988], [Klartag, Mendelson, 2005], [Arora, Hazan, Kale, 2006], [Clarkson, Woodruff, 2013].

Bad news: Computing ΠA naively takes time $O(mnd^{\omega-2})$ (even worse than $O(nd^{\omega-1})$)

Picking better subspace embeddings

The trouble is that a random Gaussian matrix is unstructured.

Sarlós' idea: Pick Π to be a structured matrix so that ΠA can be computed quickly. Sarlós used FFT-based approach of [Ailon, Chazelle, 2006]+followup work with $m = \tilde{O}(d/\varepsilon^2)$ and such that Πx can be computed in time $O(n \log n)$ for any $x \in \mathbb{R}^n$.

Picking better subspace embeddings

The trouble is that a random Gaussian matrix is unstructured.

Sarlós' idea: Pick Π to be a structured matrix so that ΠA can be computed quickly. Sarlós used FFT-based approach of [Ailon, Chazelle, 2006]+followup work with $m = \tilde{O}(d/\varepsilon^2)$ and such that Πx can be computed in time $O(n \log n)$ for any $x \in \mathbb{R}^n$.

Can compute ΠA in time $O(nd \log n)$ by computing Π times each column of A separately.

Conclusion: Can solve, e.g. least squares regression, in time $O(nd \log n) + \tilde{O}(d^\omega/\varepsilon^2)$. Nearly linear time in matrix size!

Picking better subspace embeddings

The trouble is that a random Gaussian matrix is unstructured.

Sarlós' idea: Pick Π to be a structured matrix so that ΠA can be computed quickly. Sarlós used FFT-based approach of [Ailon, Chazelle, 2006]+followup work with $m = \tilde{O}(d/\varepsilon^2)$ and such that Πx can be computed in time $O(n \log n)$ for any $x \in \mathbb{R}^n$.

Can compute ΠA in time $O(nd \log n)$ by computing Π times each column of A separately.

Conclusion: Can solve, e.g. least squares regression, in time $O(nd \log n) + \tilde{O}(d^\omega/\varepsilon^2)$. Nearly linear time in matrix size!

Can we do better?

Linear time in input sparsity

[Clarkson, Woodruff, 2013] constructed a Π with $m = \mathrm{poly}(d/\varepsilon)$ rows so that each column has exactly *one non-zero entry*!

Implication: E.g. least squares regression, running time $\mathsf{nnz}(A) + \mathrm{poly}(d/\varepsilon)$.

Linear time in input sparsity

[Clarkson, Woodruff, 2013] constructed a Π with $m = \mathrm{poly}(d/\varepsilon)$ rows so that each column has exactly *one non-zero entry*!

Implication: E.g. least squares regression, running time $\mathrm{nnz}(A) + \mathrm{poly}(d/\varepsilon)$.

Let the number of non-zeroes per column be s
(so can multiply ΠA in time $s \cdot \mathrm{nnz}(A)$)

	m	s
[Kane, N. '12]	$O(d/\varepsilon^2)$	$O(d/\varepsilon)$
[Clarkson, Woodruff '13]	$O(d^2 \log^6(d/\varepsilon)/\varepsilon^2)$	1

Linear time in input sparsity

[Clarkson, Woodruff, 2013] constructed a Π with $m = \mathrm{poly}(d/\varepsilon)$ rows so that each column has exactly *one non-zero entry*!

Implication: E.g. least squares regression, running time $\mathrm{nnz}(A) + \mathrm{poly}(d/\varepsilon)$.

Let the number of non-zeroes per column be s
(so can multiply ΠA in time $s \cdot \mathrm{nnz}(A)$)

	m	s
[Kane, N. '12]	$O(d/\varepsilon^2)$	$O(d/\varepsilon)$
[Clarkson, Woodruff '13]	$O(d^2 \log^6(d/\varepsilon)/\varepsilon^2)$	1
this work	$O(d^{1+\gamma}/\varepsilon^2)$	$O_\gamma(1/\varepsilon)$
this work	$O(d^2/\varepsilon^2)^*$	1

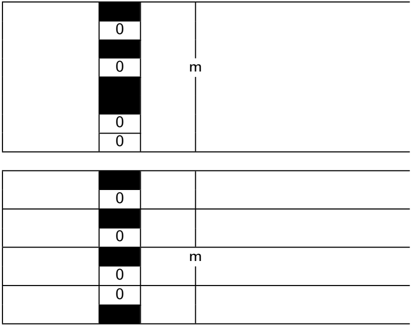
$\gamma > 0$ can be chosen as an arbitrarily small constant.
* Also obtained by [Mahoney, Meng '13], and also follows from [Thorup, Zhang '04] + [Kane, N., '12] (observed by Nguyễn).

The embedding Π

OSNAP distributions
(Oblivious Sparse Norm-Approximating Projections)

Construction 1:

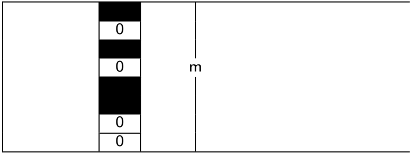
Construction 2:



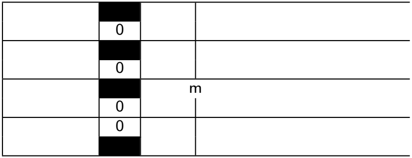
Each black cell is $\pm 1/\sqrt{s}$ at random, s black cells per column

OSNAP distributions
(Oblivious Sparse Norm-Approximating Projections)

Construction 1:



Construction 2:



Each black cell is $\pm 1/\sqrt{s}$ at random, s black cells per column

These matrices first found applications to other problems in the data streams literature in [Charikar, Chen, Farach-Colton '02], [Thorup, Zhang '04].
Also used in "sparse Johnson-Lindenstrauss" [Kane, N. '12].

Analysis

Analysis outline

Recall we have $V \subset \mathbb{R}^n$ a linear subspace of dimension d and want

$$\forall x \in V,\ (1-\varepsilon)\|x\| \le \|\Pi x\| \le (1+\varepsilon)\|x\| \quad (*)$$

Analysis outline

Recall we have $V \subset \mathbb{R}^n$ a linear subspace of dimension d and want

$$\forall x \in V,\ (1-\varepsilon)\|x\| \leq \|\Pi x\| \leq (1+\varepsilon)\|x\| \quad (*)$$

$V = \{Uy : y \in \mathbb{R}^d\}$, where the columns of U form an orthonormal basis for V. Thus (*) is equivalent to

$$\forall y \in \mathbb{R}^d,\ \|\Pi U y\| = (1 \pm \varepsilon)\|Uy\| = (1 \pm \varepsilon)\|y\| \quad (**)$$

Analysis outline

Recall we have $V \subset \mathbb{R}^n$ a linear subspace of dimension d and want

$$\forall x \in V,\ (1-\varepsilon)\|x\| \le \|\Pi x\| \le (1+\varepsilon)\|x\| \quad (*)$$

$V = \{Uy : y \in \mathbb{R}^d\}$, where the columns of U form an orthonormal basis for V. Thus (*) is equivalent to

$$\forall y \in \mathbb{R}^d,\ \|\Pi U y\| = (1 \pm \varepsilon)\|Uy\| = (1 \pm \varepsilon)\|y\| \quad (**)$$

(**) equivalent to all eigenvals of $S = (\Pi U)^T(\Pi U)$ being $(1 \pm \varepsilon)^2$, which is equivalent to $\|S - I\| \le \varepsilon$ (up to a factor of 2).

Analysis outline

Recall we have $V \subset \mathbb{R}^n$ a linear subspace of dimension d and want

$$\forall x \in V,\ (1-\varepsilon)\|x\| \le \|\Pi x\| \le (1+\varepsilon)\|x\| \quad (*)$$

$V = \{Uy : y \in \mathbb{R}^d\}$, where the columns of U form an orthonormal basis for V. Thus (*) is equivalent to

$$\forall y \in \mathbb{R}^d,\ \|\Pi Uy\| = (1 \pm \varepsilon)\|Uy\| = (1 \pm \varepsilon)\|y\| \quad (**)$$

(**) equivalent to all eigenvals of $S = (\Pi U)^T(\Pi U)$ being $(1 \pm \varepsilon)^2$, which is equivalent to $\|S - I\| \le \varepsilon$ (up to a factor of 2).

Markov's inequality:

$$\mathbb{P}(\|S-I\| > \varepsilon) = \mathbb{P}(\|S-I\|^\ell > \varepsilon^\ell) < \frac{1}{\varepsilon^\ell}\,\mathbb{E}\,\|S-I\|^\ell \le \frac{1}{\varepsilon^\ell}\,\mathbb{E}\,\mathrm{tr}((S-I)^\ell)$$

Analysis outline

Recall we have $V \subset \mathbb{R}^n$ a linear subspace of dimension d and want

$$\forall x \in V,\ (1-\varepsilon)\|x\| \le \|\Pi x\| \le (1+\varepsilon)\|x\| \quad (*)$$

$V = \{Uy : y \in \mathbb{R}^d\}$, where the columns of U form an orthonormal basis for V. Thus (*) is equivalent to

$$\forall y \in \mathbb{R}^d,\ \|\Pi U y\| = (1 \pm \varepsilon)\|Uy\| = (1 \pm \varepsilon)\|y\| \quad (**)$$

(**) equivalent to all eigenvals of $S = (\Pi U)^T(\Pi U)$ being $(1 \pm \varepsilon)^2$, which is equivalent to $\|S - I\| \le \varepsilon$ (up to a factor of 2).

Markov's inequality:

$$\mathbb{P}(\|S-I\| > \varepsilon) = \mathbb{P}(\|S-I\|^\ell > \varepsilon^\ell) < \frac{1}{\varepsilon^\ell}\,\mathbb{E}\,\|S-I\|^\ell \le \frac{1}{\varepsilon^\ell}\,\mathbb{E}\,\mathrm{tr}((S-I)^\ell)$$

This is the classical "moment method" in random matrix theory; see e.g. [Wigner, 1955], [Füredi, Komlós, 1981], [Bai, Yin, 1993]

Natural "matrix extension" of JL

Johnson-Lindenstrauss lemma

Theorem

Let $u \in \mathbb{R}^n$ be arbitrary, unit ℓ_2 norm, Π random sign matrix. Then

$$\mathbb{P}_{\Pi}\left(\left|\|\Pi u\|^2 - 1\right| > \varepsilon\right) < \delta$$

as long as

$$m \gtrsim \frac{\log(1/\delta)}{\varepsilon^2}, \ell = \log(1/\delta) ([\text{Achlioptas}'01])$$

or

$$m \gtrsim \frac{1}{\varepsilon^2 \delta}, \ell = 2 \ ([\text{Alon, Matias, Szegedy}'96])$$

Natural "matrix extension" of JL

Johnson-Lindenstrauss lemma

Theorem

Let $u \in \mathbb{R}^{n\times 1}$ be arbitrary, o.n. cols, Π random sign matrix. Then

$$\mathbb{P}_{\Pi}\left(\|(\Pi u)^*(\Pi u) - I_1\| > \varepsilon\right) < \delta$$

as long as

$$m \gtrsim \frac{1+\log(1/\delta)}{\varepsilon^2}, \ell = \log(1/\delta)$$

or

$$m \gtrsim \frac{1^2}{\varepsilon^2\delta}, \ell = 2$$

Natural “matrix extension” of JL

Conjecture

Theorem

Let $u \in \mathbb{R}^{n \times d}$ *be arbitrary, o.n. cols,* Π *random sign matrix. Then*

$$\mathbb{P}_{\Pi}\left(\|(\Pi u)^*(\Pi u) - I_d\| > \varepsilon\right) < \delta$$

as long as

$$m \gtrsim \frac{d + \log(1/\delta)}{\varepsilon^2}, \ell = \log(d/\delta)$$

or

$$m \gtrsim \frac{d^2}{\varepsilon^2 \delta}, \ell = 2$$

Natural "matrix extension" of sparse JL

[Kane, N. '12]

Theorem

Let $u \in \mathbb{R}^n$ be arbitrary, unit ℓ_2 norm, Π sparse sign matrix. Then

$$\mathbb{P}_{\Pi}\left(\left|\|\Pi u\|^2 - 1\right| > \varepsilon\right) < \delta$$

as long as

$$m \gtrsim \frac{\log(1/\delta)}{\varepsilon^2}, s \gtrsim \frac{\log(1/\delta)}{\varepsilon}, \ell = \log(1/\delta)$$

or

$$m \gtrsim \frac{1}{\varepsilon^2 \delta}, s = 1, \ell = 2 \text{ ([Thorup, Zhang'04])}$$

Natural "matrix extension" of sparse JL

[Kane, N. '12]

Theorem

Let $u \in \mathbb{R}^{n\times 1}$ *be arbitrary, o.n. cols,* Π *sparse sign matrix. Then*

$$\mathbb{P}_{\Pi}(\|(\Pi u)^*(\Pi u) - I_1\| > \varepsilon) < \delta$$

as long as

$$m \gtrsim \frac{1+\log(1/\delta)}{\varepsilon^2}, s \gtrsim \frac{\log(1/\delta)}{\varepsilon}, \ell = \log(1/\delta)$$

or

$$m \gtrsim \frac{1^2}{\varepsilon^2\delta}, s = 1, \ell = 2$$

Natural “matrix extension” of sparse JL

Conjecture

Theorem

Let $u \in \mathbb{R}^{n\times d}$ be arbitrary, o.n. cols, Π sparse sign matrix. Then

$$\mathbb{P}_{\Pi}(\|(\Pi u)^*(\Pi u) - I_d\| > \varepsilon) < \delta$$

as long as

$$m \gtrsim \frac{d + \log(1/\delta)}{\varepsilon^2}, s \gtrsim \frac{\log(d/\delta)}{\varepsilon}, \ell = \log(d/\delta)$$

or

$$m \gtrsim \frac{d^2}{\varepsilon^2\delta}, s = 1, \ell = 2$$

Natural "matrix extension" of sparse JL

What we prove

Theorem

Let $u \in \mathbb{R}^{n\times d}$ be arbitrary, o.n. cols, Π sparse sign matrix. Then

$$\mathbb{P}_{\Pi}(\|(\Pi u)^*(\Pi u) - I_d\| > \varepsilon) < \delta$$

as long as

$$m \gtrsim \frac{d\cdot\log^c(d/\delta)}{\varepsilon^2}, s \gtrsim \frac{\log^c(d/\delta)}{\varepsilon} \text{ or } m \gtrsim \frac{d^{1.01}}{\varepsilon^2}, s \gtrsim \frac{1}{\varepsilon}$$

or

$$m \gtrsim \frac{d^2}{\varepsilon^2\delta}, s = 1$$

Back to the analysis

Analysis ($\ell = 2$)
$s = 1$, $m = O(d^2/\varepsilon^2)$

Want to understand $S - I$, $S = (\Pi U)^T(\Pi U)$

Analysis ($\ell = 2$) $s = 1$, $m = O(d^2/\varepsilon^2)$

Want to understand $S - I$, $S = (\Pi U)^T(\Pi U)$

Let the columns of U be $u^1, \ldots, u^d$
Recall $\Pi_{i,j} = \delta_{i,j}\sigma_{i,j}/\sqrt{s}$

Analysis ($\ell = 2$)
$s = 1$, $m = O(d^2/\varepsilon^2)$

Want to understand $S - I$, $S = (\Pi U)^T(\Pi U)$

Let the columns of U be $u^1, \ldots, u^d$
Recall $\Pi_{i,j} = \delta_{i,j}\sigma_{i,j}/\sqrt{s}$

Some computations yield

$$(S - I)_{k,k'} = \frac{1}{s}\sum_{r=1}^{m}\sum_{i \neq j}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}u_i^k u_j^{k'}$$

Analysis ($\ell = 2$)
$s = 1$, $m = O(d^2/\varepsilon^2)$

Want to understand $S - I$, $S = (\Pi U)^T(\Pi U)$

Let the columns of U be $u^1, \ldots, u^d$
Recall $\Pi_{i,j} = \delta_{i,j}\sigma_{i,j}/\sqrt{s}$

Some computations yield

$$(S - I)_{k,k'} = \frac{1}{s}\sum_{r=1}^{m}\sum_{i \neq j}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}u_i^k u_j^{k'}$$

Computing $\mathbb{E}\,\|S - I\|_F^2$ is straightforward, and can show $\mathbb{E}\,\|S - I\|_F^2 \leq (d^2 + d)/m$

$$\mathbb{P}(\|S - I\| > \varepsilon) < \frac{1}{\varepsilon^2}\frac{d^2 + d}{m}$$

Analysis ($\ell = 2$)
$s = 1$, $m = O(d^2/\varepsilon^2)$

Want to understand $S - I$, $S = (\Pi U)^T(\Pi U)$

Let the columns of U be $u^1, \ldots, u^d$
Recall $\Pi_{i,j} = \delta_{i,j}\sigma_{i,j}/\sqrt{s}$

Some computations yield

$$(S - I)_{k,k'} = \frac{1}{s}\sum_{r=1}^{m}\sum_{i \neq j}\delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j}u_i^k u_j^{k'}$$

Computing $\mathbb{E}\,\|S - I\|_F^2$ is straightforward, and can show $\mathbb{E}\,\|S - I\|_F^2 \leq (d^2 + d)/m$

$$\mathbb{P}(\|S - I\| > \varepsilon) < \frac{1}{\varepsilon^2}\frac{d^2 + d}{m}$$

Set $m \geq \delta^{-1}(d^2 + d)/\varepsilon^2$ for success probability $1 - \delta$

Analysis (large ℓ)
$s = O_\gamma(1/\varepsilon)$, $m = O(d^{1+\gamma}/\varepsilon^2)$

$$(S - I)_{k,k'} = \frac{1}{s} \sum_{r=1}^{m} \sum_{i \neq j} \delta_{r,i} \delta_{r,j} \sigma_{r,i} \sigma_{r,j} u_i^k u_j^{k'}$$

Analysis (large ℓ) $s = O_\gamma(1/\varepsilon)$, $m = O(d^{1+\gamma}/\varepsilon^2)$

$$(S - I)_{k,k'} = \frac{1}{s} \sum_{r=1}^{m} \sum_{i \neq j} \delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j} u_i^k u_j^{k'}$$

By induction, for any square matrix B and integer $\ell \geq 1$,

$$(B^\ell)_{i,j} = \sum_{\substack{i_1,\ldots,i_{\ell+1} \\ i_1 = i, i_{\ell+1} = j}} \prod_{t=1}^{\ell} B_{i_t, i_{t+1}}$$

Analysis (large ℓ)
$s = O_\gamma(1/\varepsilon)$, $m = O(d^{1+\gamma}/\varepsilon^2)$

$$(S - I)_{k,k'} = \frac{1}{s} \sum_{r=1}^{m} \sum_{i \neq j} \delta_{r,i}\delta_{r,j}\sigma_{r,i}\sigma_{r,j} u_i^k u_j^{k'}$$

By induction, for any square matrix B and integer $\ell \geq 1$,

$$(B^\ell)_{i,j} = \sum_{\substack{i_1,\ldots,i_{\ell+1} \\ i_1 = i, i_{\ell+1} = j}} \prod_{t=1}^{\ell} B_{i_t, i_{t+1}}$$

$$\Rightarrow \mathrm{tr}(B^\ell) = \sum_{\substack{i_1,\ldots,i_{\ell+1} \\ i_1 = i_{\ell+1}}} \prod_{t=1}^{\ell} B_{i_t, i_{t+1}}$$

Analysis (large ℓ) $s = O_\gamma(1/\varepsilon)$, $m = O(d^{1+\gamma}/\varepsilon^2)$

$$\mathbb{E}\,\mathrm{tr}((S-I)^\ell) = \sum_{\substack{i_1 \neq j_1, \ldots, i_\ell \neq j_\ell \\ r_1, \ldots, r_\ell \\ k_1, \ldots, k_{\ell+1} \\ k_1 = k_{\ell+1}}} \left(\mathbb{E} \prod_{t=1}^{\ell} \delta_{r_t, i_t} \delta_{r_t, j_t}\right) \left(\mathbb{E} \prod_{t=1}^{\ell} \sigma_{r_t, i_t} \sigma_{r_t, j_t}\right) \prod_{t=1}^{\ell} u_{i_t}^{k_t} u_{j_t}^{k_{t+1}}$$

Analysis (large ℓ)
$s = O_\gamma(1/\varepsilon)$, $m = O(d^{1+\gamma}/\varepsilon^2)$

$$\mathbb{E}\,\mathrm{tr}((S-I)^\ell) = \sum_{\substack{i_1\neq j_1,\ldots,i_\ell\neq j_\ell \\ r_1,\ldots,r_\ell \\ k_1,\ldots,k_{\ell+1} \\ k_1=k_{\ell+1}}} \left(\mathbb{E}\prod_{t=1}^{\ell}\delta_{r_t,i_t}\delta_{r_t,j_t}\right)\left(\mathbb{E}\prod_{t=1}^{\ell}\sigma_{r_t,i_t}\sigma_{r_t,j_t}\right)\prod_{t=1}^{\ell}u_{i_t}^{k_t}u_{j_t}^{k_{t+1}}$$

The strategy: Associate each monomial in summation above with a graph, group monomials that have the same graph, and estimate the contribution of each graph then do some combinatorics

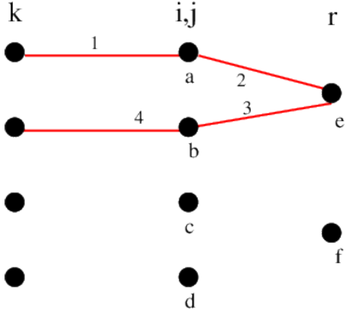
(a common strategy; see [Wigner, 1955], [Füredi, Komlós, 1981], [Bai, Yin, 1993])

Example monomial→graph correspondence

$$\mathrm{tr}((S-I)^{\ell}) = \sum_{\substack{i_1 \neq j_1, \ldots, i_\ell \neq j_\ell \\ r_1, \ldots, r_\ell \\ k_1, \ldots, k_{\ell+1} \\ k_1 = k_{\ell+1}}} \prod_{t=1}^{\ell} \delta_{r_t, i_t} \delta_{r_t, j_t} \cdot \prod_{t=1}^{\ell} \sigma_{r_t, i_t} \sigma_{r_t, j_t} \cdot \prod_{t=1}^{\ell} u_{i_t}^{k_t} u_{j_t}^{k_{t+1}}$$

$$\ell = 4$$

$$\delta_{r_e, i_a} \delta_{r_e, i_b} \sigma_{r_e, i_a} \sigma_{r_e, i_b} u_{i_a}^{k_1} u_{i_b}^{k_2}$$

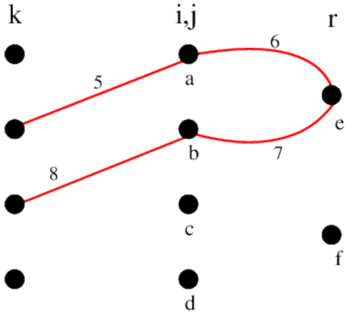


Example monomial→graph correspondence

$$\mathrm{tr}((S-I)^\ell) = \sum_{\substack{i_1\neq j_1,\ldots,i_\ell\neq j_\ell \\ r_1,\ldots,r_\ell \\ k_1,\ldots,k_{\ell+1} \\ k_1=k_{\ell+1}}} \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t} \cdot \prod_{t=1}^{\ell} \sigma_{r_t,i_t}\sigma_{r_t,j_t} \cdot \prod_{t=1}^{\ell} u_{i_t}^{k_t} u_{j_t}^{k_{t+1}}$$

$$\ell = 4$$

$$\times \delta_{r_e,i_a}\delta_{r_e,i_b}\sigma_{r_e,i_a}\sigma_{r_e,i_b} u_{i_a}^{k_2} u_{i_b}^{k_3}$$

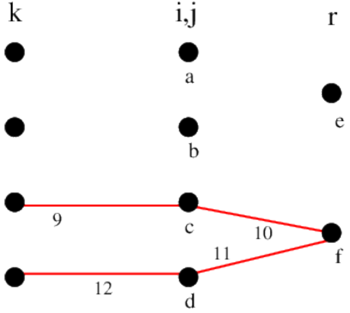


Example monomial→graph correspondence

$$\mathrm{tr}((S-I)^{\ell}) = \sum_{\substack{i_1\neq j_1,\ldots,i_\ell\neq j_\ell \\ r_1,\ldots,r_\ell \\ k_1,\ldots,k_{\ell+1} \\ k_1=k_{\ell+1}}} \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t} \cdot \prod_{t=1}^{\ell} \sigma_{r_t,i_t}\sigma_{r_t,j_t} \cdot \prod_{t=1}^{\ell} u_{i_t}^{k_t} u_{j_t}^{k_{t+1}}$$

$$\ell = 4$$

$$\times \delta_{r_f,i_c}\delta_{r_f,i_d}\sigma_{r_f,i_c}\sigma_{r_f,i_d} u_{i_c}^{k_3} u_{i_d}^{k_4}$$

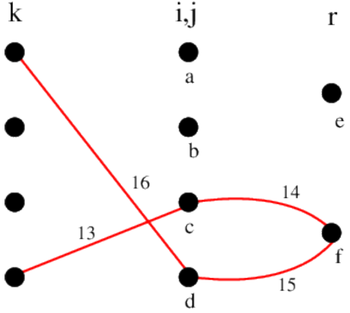


Example monomial→graph correspondence

$$\mathrm{tr}((S-I)^{\ell}) = \sum_{\substack{i_1\neq j_1,\ldots,i_\ell\neq j_\ell \\ r_1,\ldots,r_\ell \\ k_1,\ldots,k_{\ell+1} \\ k_1=k_{\ell+1}}} \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t} \cdot \prod_{t=1}^{\ell} \sigma_{r_t,i_t}\sigma_{r_t,j_t} \cdot \prod_{t=1}^{\ell} u_{i_t}^{k_t} u_{j_t}^{k_{t+1}}$$

$$\ell = 4$$

$$\times \delta_{r_f,i_c}\delta_{r_f,i_d}\sigma_{r_f,i_c}\sigma_{r_f,i_d} u_{i_c}^{k_4} u_{i_d}^{k_1}$$

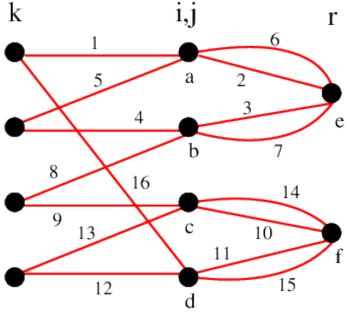


Example monomial→graph correspondence

$$\mathrm{tr}((S-I)^\ell) = \sum_{\substack{i_1\neq j_1,\ldots,i_\ell\neq j_\ell \\ r_1,\ldots,r_\ell \\ k_1,\ldots,k_{\ell+1} \\ k_1=k_{\ell+1}}} \prod_{t=1}^{\ell} \delta_{r_t,i_t}\delta_{r_t,j_t} \cdot \prod_{t=1}^{\ell} \sigma_{r_t,i_t}\sigma_{r_t,j_t} \cdot \prod_{t=1}^{\ell} u_{i_t}^{k_t} u_{j_t}^{k_{t+1}}$$

$$\ell = 4$$

$$\begin{aligned} &\delta_{r_e,i_a}\delta_{r_e,i_b}\sigma_{r_e,i_a}\sigma_{r_e,i_b} u_{i_a}^{k_1} u_{i_b}^{k_2} \\ \times &\delta_{r_e,i_a}\delta_{r_e,i_b}\sigma_{r_e,i_a}\sigma_{r_e,i_b} u_{i_a}^{k_2} u_{i_b}^{k_3} \\ \times &\delta_{r_f,i_c}\delta_{r_f,i_d}\sigma_{r_f,i_c}\sigma_{r_f,i_d} u_{i_c}^{k_3} u_{i_d}^{k_4} \\ \times &\delta_{r_f,i_c}\delta_{r_f,i_d}\sigma_{r_f,i_c}\sigma_{r_f,i_d} u_{i_c}^{k_4} u_{i_d}^{k_1} \end{aligned}$$

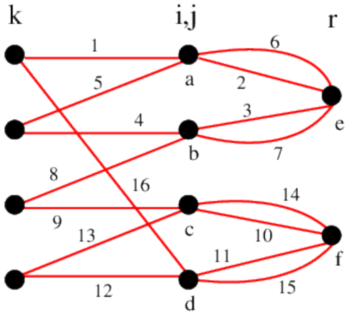


Example monomial→graph correspondence

$$\mathrm{tr}((S-I)^{\ell}) = \sum_{\substack{i_1 \neq j_1, \ldots, i_\ell \neq j_\ell \\ r_1, \ldots, r_\ell}} \prod_{t=1}^{\ell} \delta_{r_t, i_t} \delta_{r_t, j_t} \cdot \prod_{t=1}^{\ell} \sigma_{r_t, i_t} \sigma_{r_t, j_t} \cdot \prod_{t=1}^{\ell} \langle u_{i_t}, u_{i_{t+1}} \rangle$$

$$\ell = 4$$

$$\delta_{r_e,i_a}\delta_{r_e,i_b}\sigma_{r_e,i_a}\sigma_{r_e,i_b}u_{i_a}^{k_1}u_{i_b}^{k_2}$$
$$\times\delta_{r_e,i_a}\delta_{r_e,i_b}\sigma_{r_e,i_a}\sigma_{r_e,i_b}u_{i_a}^{k_2}u_{i_b}^{k_3}$$
$$\times\delta_{r_f,i_c}\delta_{r_f,i_d}\sigma_{r_f,i_c}\sigma_{r_f,i_d}u_{i_c}^{k_3}u_{i_d}^{k_4}$$
$$\times\delta_{r_f,i_c}\delta_{r_f,i_d}\sigma_{r_f,i_c}\sigma_{r_f,i_d}u_{i_c}^{k_4}u_{i_d}^{k_1}$$



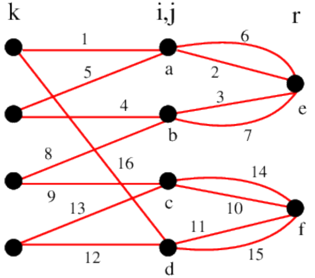
Grouping monomials by graph

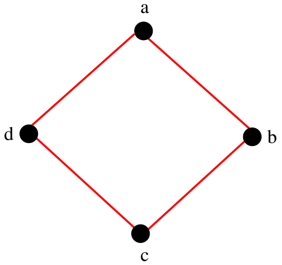
z right vertices, b distinct edges between middle and right

$$\mathbb{E}\,\mathrm{tr}((S-I)^{\ell}) = \sum_{\substack{i_1 \neq j_1, \ldots, i_\ell \neq j_\ell \\ r_1, \ldots, r_\ell}} \left(\mathbb{E} \prod_{t=1}^{\ell} \delta_{r_t, i_t} \delta_{r_t, j_t} \right) \left(\mathbb{E} \prod_{t=1}^{\ell} \sigma_{r_t, i_t} \sigma_{r_t, j_t} \right) \prod_{t=1}^{\ell} \langle u_{i_t}, u_{i_{t+1}} \rangle$$

$$\leq \sum_{G} m^{z} \left(\frac{s}{m} \right)^{b} \left| \sum_{i_1 \neq \ldots \neq i_y} \prod_{e=(\alpha,\beta) \in \hat{G}} \langle u_{i_\alpha}, u_{i_\beta} \rangle \right|$$

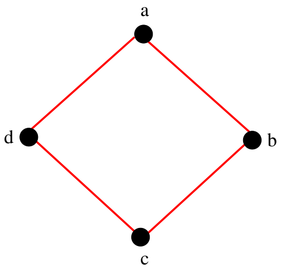
G	$\hat{G}$





Understanding $\hat{G}$

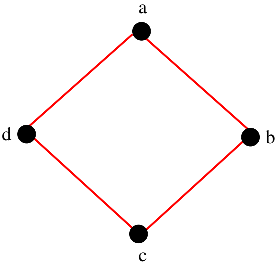
$$F(\hat{G}) = \left| \sum_{i_1 \neq \ldots \neq i_y} \prod_{e=(\alpha,\beta)\in \hat{G}} \left\langle u_{i_\alpha}, u_{i_\beta} \right\rangle \right|$$



Let C be the number of connected components of $\hat{G}$. It turns out the right upper bound for $F(\hat{G})$ is roughly d^C

Understanding $\hat{G}$

$$F(\hat{G}) = \left| \sum_{i_1 \neq \ldots \neq i_y} \prod_{e=(\alpha,\beta)\in\hat{G}} \left\langle u_{i_\alpha}, u_{i_\beta} \right\rangle \right|$$

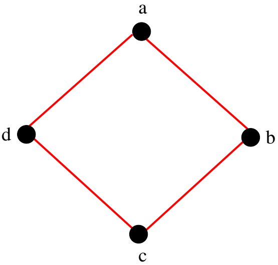


Let C be the number of connected components of $\hat{G}$. It turns out the right upper bound for $F(\hat{G})$ is roughly d^C

- Can get d^C bound if all edges in $\hat{G}$ have even multiplicity

Understanding $\hat{G}$

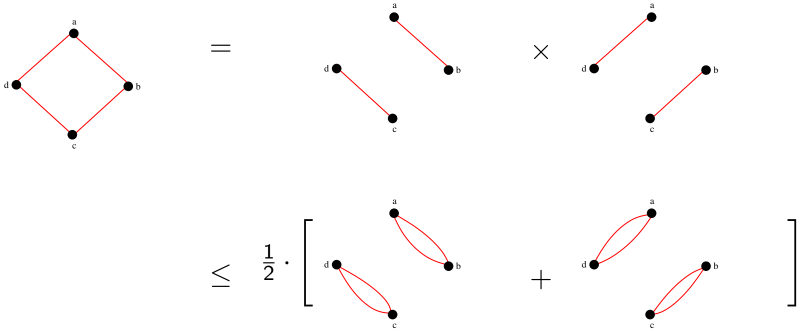
$$F(\hat{G}) = \left| \sum_{i_1 \neq \ldots \neq i_y} \prod_{e=(\alpha,\beta)\in\hat{G}} \left\langle u_{i_\alpha}, u_{i_\beta} \right\rangle \right|$$



Let C be the number of connected components of $\hat{G}$. It turns out the right upper bound for $F(\hat{G})$ is roughly d^C

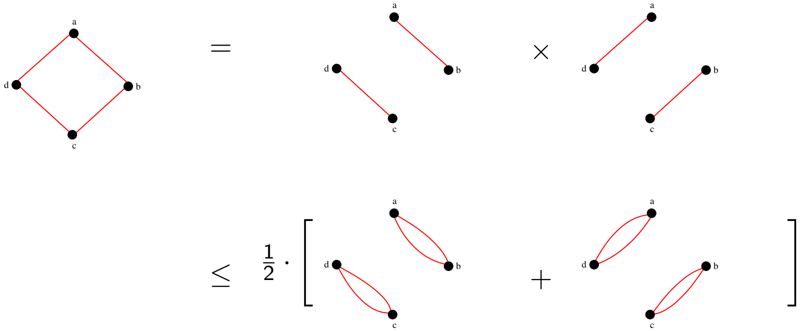
- Can get d^C bound if all edges in $\hat{G}$ have even multiplicity
- How about $\hat{G}$ where this isn't the case, e.g. as above?

Bounding $F(\hat{G})$ with odd multiplicities



Reduces back to case of even edge multiplicities! (AM-GM)

Bounding $F(\hat{G})$ with odd multiplicities



Reduces back to case of even edge multiplicities! (AM-GM)

Caveat: # connected components increased (unacceptable)

AM-GM trick done right

Theorem (Tutte '61, Nash-Williams '61)

Let G be a multigraph with edge-connectivity at least $2k$. Then G must have at least k edge-disjoint spanning trees.

AM-GM trick done right

Theorem (Tutte '61, Nash-Williams '61)

Let G be a multigraph with edge-connectivity at least $2k$. Then G must have at least k edge-disjoint spanning trees.

Using the theorem ($k = 2$)

- If every connected component (CC) of $\hat{G}$ has 2 edge-disjoint spanning trees, we are done

AM-GM trick done right

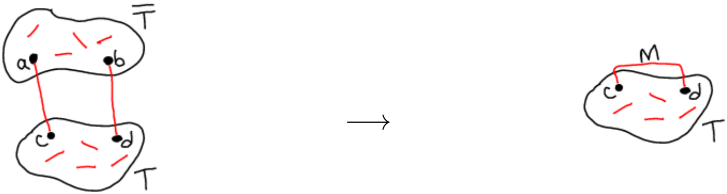
Theorem (Tutte '61, Nash-Williams '61)

Let G be a multigraph with edge-connectivity at least $2k$. Then G must have at least k edge-disjoint spanning trees.

Using the theorem ($k = 2$)

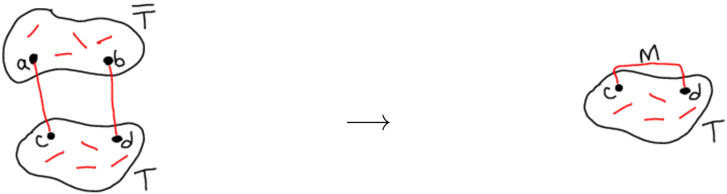
- If every connected component (CC) of $\hat{G}$ has 2 edge-disjoint spanning trees, we are done
- Otherwise, some CC is not 4 edge-connected. Since each CC is Eulerian, there must be a cut of size 2

AM-GM trick done right



$$\sum_{\substack{i_v\\ v\in T}} \left(\prod_{(q,r)\in T}\langle u_{i_q}, u_{i_r}\rangle\right) u_{i_c}^T \underbrace{\left(\sum_{\substack{i_v\\ v\in \bar{T}}} u_{i_a}\left(\prod_{(q,r)\in\bar{T}}\langle u_{i_q},u_{i_r}\rangle\right)u_{i_b}^T\right)}_{M} u_{i_d}$$

AM-GM trick done right

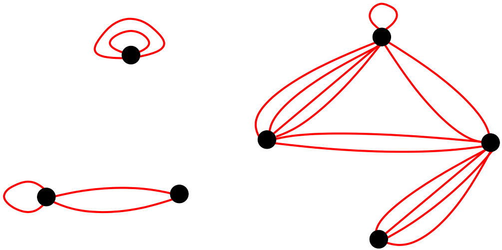


$$\sum_{\substack{i_v \\ v \in T}} \left(\prod_{(q,r) \in T} \langle u_{i_q}, u_{i_r} \rangle \right) u_{i_c}^T \underbrace{\left(\sum_{\substack{i_v \\ v \in \bar{T}}} u_{i_a} \left(\prod_{(q,r) \in \bar{T}} \langle u_{i_q}, u_{i_r} \rangle \right) u_{i_b}^T \right)}_{M} u_{i_d}$$

- Repeatedly eliminate size-2 cuts until every CC has two edge-disjoint spanning trees
- Show all M's along the way have bounded operator norm
- Show that even edge multiplicities are still easy to handle when all M's have bounded operator norm

Handling even edge multiplicities

$\hat{G}$



Handling even edge multiplicities

Rough idea

- Note
  1. $\langle u_i, u_j \rangle^2 = u_j^T u_i u_i^T u_j$
  2. Also $\sum_{i=1}^{n} u_i u_i^T = I$

Handling even edge multiplicities

Rough idea

- Note
  1. $\langle u_i, u_j \rangle^2 = u_j^T u_i u_i^T u_j$
  2. Also $\sum_{i=1}^{n} u_i u_i^T = I$
- In graph terms, we can choose to remove any vertex x we want from the dot product graph (by summing over its assignments). Then for each neighbor of x we attach self-loops (one self-loop for every two edges to x).

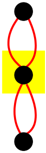
Handling even edge multiplicities

Rough idea

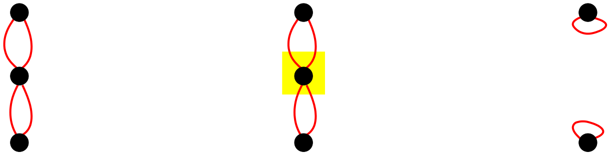
- Note
  1. $\langle u_i, u_j \rangle^2 = u_j^T u_i u_i^T u_j$
  2. Also $\sum_{i=1}^{n} u_i u_i^T = I$
- In graph terms, we can choose to remove any vertex x we want from the dot product graph (by summing over its assignments). Then for each neighbor of x we attach self-loops (one self-loop for every two edges to x).
- What order do we sum over vertices?

Vertex summation order: even edge multiplicities

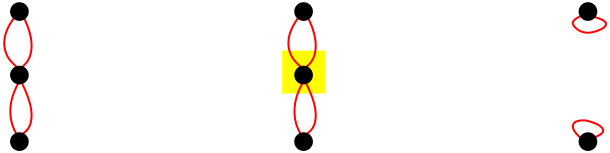
Vertex summation order: even edge multiplicities



Vertex summation order: even edge multiplicities



Vertex summation order: even edge multiplicities



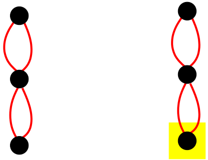
Bad order: increased the number of connected components

Vertex summation order: even edge multiplicities

A better order:

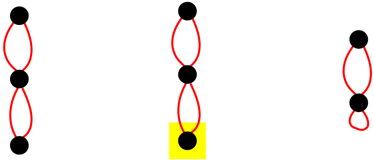
Vertex summation order: even edge multiplicities

A better order:



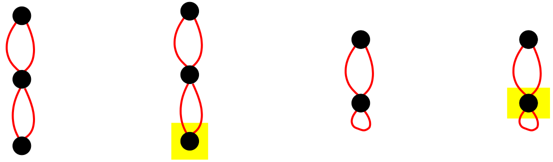
Vertex summation order: even edge multiplicities

A better order:



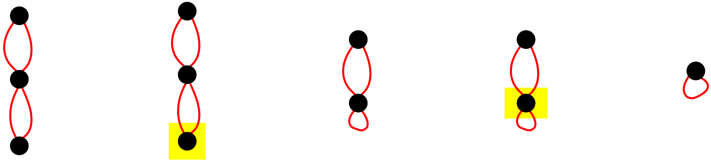
Vertex summation order: even edge multiplicities

A better order:



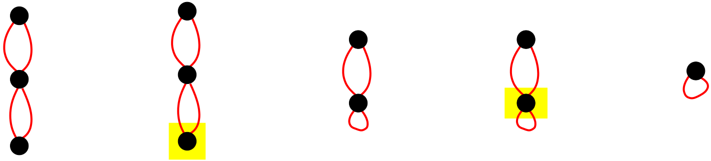
Vertex summation order: even edge multiplicities

A better order:



Vertex summation order: even edge multiplicities

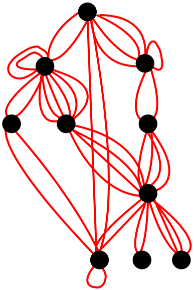
A better order:



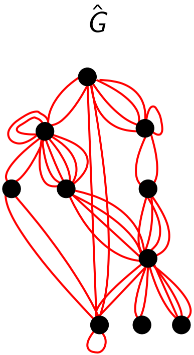
In general: for each connected component of $\hat{G}$ take some spanning tree, then sum over the vertices that are lower in the tree first.

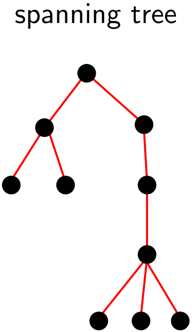
Vertex summation order: even edge multiplicities

$\hat{G}$



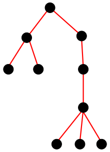
Vertex summation order: even edge multiplicities



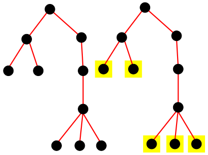


Step 1: Take a spanning tree of $\hat{G}$

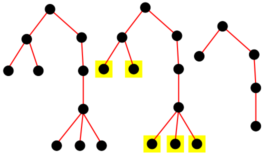
Vertex summation order: even edge multiplicities



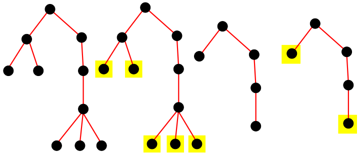
Vertex summation order: even edge multiplicities



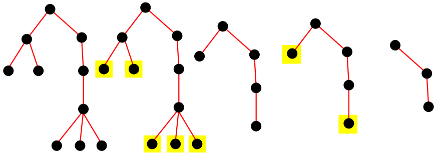
Vertex summation order: even edge multiplicities



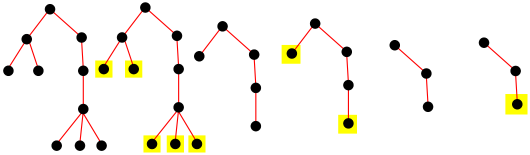
Vertex summation order: even edge multiplicities



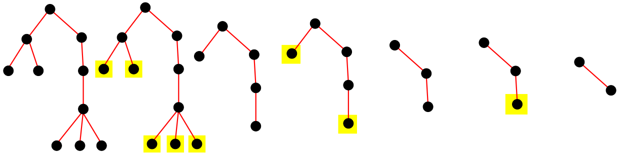
Vertex summation order: even edge multiplicities



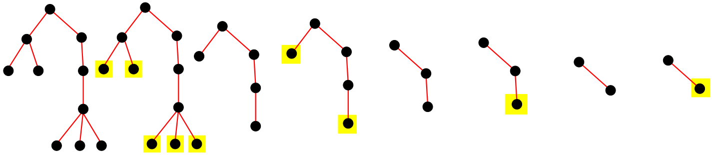
Vertex summation order: even edge multiplicities



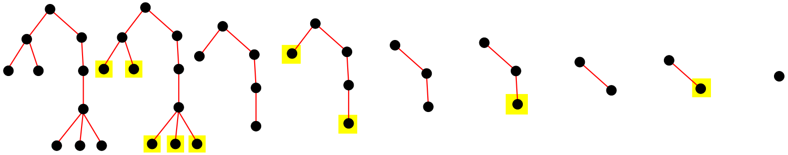
Vertex summation order: even edge multiplicities



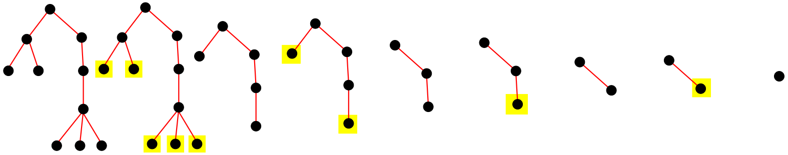
Vertex summation order: even edge multiplicities



Vertex summation order: even edge multiplicities



Vertex summation order: even edge multiplicities



Summing in this order ensures the number of connected components never increases

Conclusion

Other recent progress

- Can show any oblivious subspace embedding succeeding with probability $\geq 2/3$ must have $\Omega(d/\varepsilon^2)$ rows [N., Nguyễn]

Other recent progress

- Can show any oblivious subspace embedding succeeding with probability $\geq 2/3$ must have $\Omega(d/\varepsilon^2)$ rows [N., Nguyễn]
- Can show any oblivious subspace embedding with $O(d^{1+\gamma})$ rows must have sparsity $s = \Omega(1/(\varepsilon\gamma))$* [N., Nguyễn]

Other recent progress

- Can show any oblivious subspace embedding succeeding with probability $\geq 2/3$ must have $\Omega(d/\varepsilon^2)$ rows [N., Nguyễn]
- Can show any oblivious subspace embedding with $O(d^{1+\gamma})$ rows must have sparsity $s = \Omega(1/(\varepsilon\gamma))$* [N., Nguyễn]
- Can provide upper bounds on m, s to preserve an arbitrary bounded set $T \subset \mathbb{R}^n$, in terms of the geometry of T, in the style of [Gordon '88], [Klartag, Mendelson '05], [Mendelson, Pajor, Tomczak-Jaegermann '07], [Dirksen '13] (in the current notation, these works analyzed dense Π, i.e. $m = s$) [Bourgain, N.]

* Has restriction that $1/(\varepsilon\gamma) \ll d$.

Open Problems

- **OPEN:** Improve ω, the exponent of matrix multiplication
- **OPEN:** Find exact algorithm for least squares regression (or any of these problems) in time faster than $\tilde{O}(nd^{\omega-1})$
- **OPEN:** Prove the following conjecture: to have a subspace embedding with probability $1-\delta$, suffices to set $m = O((d + \log(1/\delta))/\varepsilon^2), s = O(\log(d/\delta)/\varepsilon)$. Or even, obtain this bound for m for a dense sign matrix using the moment method, with the $\ell = \Theta(\log(d/\delta))$th moment.
- **OPEN:** Show that the tradeoff $m = O(d^{1+\gamma}/\varepsilon^2)$, $s = \mathrm{poly}(1/\gamma) \cdot 1/\varepsilon$ is optimal for any distribution over subspace embeddings
- **OPEN:** Show that $m = \Omega(d^2/\varepsilon^2)$ is optimal for $s = 1$

  Partial progress: [N., Nguyễn, 2012] shows $m = \Omega(d^2)$