Matching Size and Matrix Rank Estimation in Data Streams

Sepehr Assadi

University of Pennsylvania

Joint work with Sanjeev Khanna (Penn), and Yang Li (Penn)

The Streaming Model

Introduced in the seminal work of Alon, Matias, and Szegedy [Alon et al., 1996].

The Streaming Model

Introduced in the seminal work of Alon, Matias, and Szegedy [Alon et al., 1996].

- Input is presented as a data stream, for instance, as a sequence of edges in case of a graph input.

The Streaming Model

Introduced in the seminal work of Alon, Matias, and Szegedy [Alon et al., 1996].

- Input is presented as a data stream, for instance, as a sequence of edges in case of a graph input.
- Algorithm sees the entire input once and only has a small space to store information about the input as it passes by.

The Streaming Model

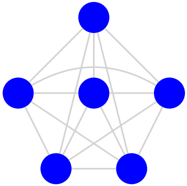
Introduced in the seminal work of Alon, Matias, and Szegedy [Alon et al., 1996].

- Input is presented as a data stream, for instance, as a sequence of edges in case of a graph input.
- Algorithm sees the entire input once and only has a small space to store information about the input as it passes by.
- At the end of the sequence, the algorithm outputs a solution using only the stored information.

The Streaming Model: Example

A graph stream:

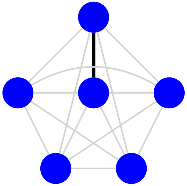
Stream:



The Streaming Model: Example

A graph stream:

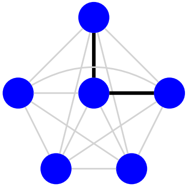
Stream: $+e_1$



The Streaming Model: Example

A graph stream:

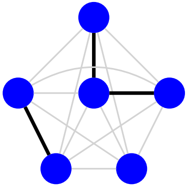
Stream: $+e_1, +e_7$



The Streaming Model: Example

A graph stream:

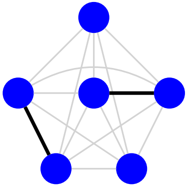
Stream: $+e_1, +e_7, +e_{11}$



The Streaming Model: Example

A graph stream:

Stream: $+e_1, +e_7, +e_{11}, -e_1$



The Streaming Model

Two relevant models for our purpose:

- Insertion-Only Streams. Only contains positive updates.

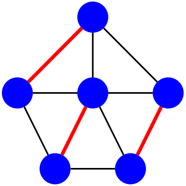
The Streaming Model

Two relevant models for our purpose:

- Insertion-Only Streams. Only contains positive updates.
- Dynamic Streams. Contains both positive and negative updates.

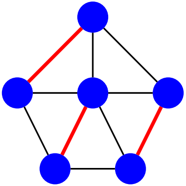
Matchings in Graphs

- Matching: A collection of vertex-disjoint edges.



Matchings in Graphs

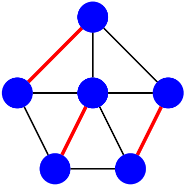
- Matching: A collection of vertex-disjoint edges.



- Perfect Matching: Every vertex is in the matching.

Matchings in Graphs

- Matching: A collection of vertex-disjoint edges.



- Perfect Matching: Every vertex is in the matching.

Maximum Matching problem: Find a matching with a largest number of edges.

Matchings in Graphs

Maximum matching is a fundamental problem with many applications.

- Many celebrated algorithms in the classical setting: Ford-Fulkerson, Edmond's, Hopcroft-Karp, Mucha-Sankowski, Madry's, ...
- Studied in various computational models: distributed, parallel, online, sub-linear time, streaming, ...

Matchings in Graphs

Maximum matching is a fundamental problem with many applications.

- Many celebrated algorithms in the classical setting: Ford-Fulkerson, Edmond's, Hopcroft-Karp, Mucha-Sankowski, Madry's, ...
- Studied in various computational models: distributed, parallel, online, sub-linear time, streaming, ...

This talk: sublinear space algorithms for the matching problem in the streaming model.

Finding a Matching vs Estimating Size

Two natural variants of the problem to consider:

- Goal 1. Output the edges in an optimal/approximate matching.

Finding a Matching vs Estimating Size

Two natural variants of the problem to consider:

- Goal 1. Output the edges in an optimal/approximate matching.
- Goal 2. Output an estimate of the size of a maximum matching.

Finding a Matching vs Estimating Size

Two natural variants of the problem to consider:

- Goal 1. Output the edges in an optimal/approximate matching.
- Goal 2. Output an estimate of the size of a maximum matching.

Are there any qualitative difference in the space needed to achieve these goals?

Finding Large Matchings

Arguably the most studied problem in the graph streaming literature.

[McGregor, 2005] [Feigenbaum et al., 2005] [Eggert et al., 2009] [Epstein et al., 2011] [Goel et al., 2012] [Konrad et al., 2012] [Zelke, 2012] [Ahn et al., 2012] [Ahn and Guha, 2013] [Guruswami and Onak, 2013] [Kapralov, 2013] [Kapralov et al., 2014] [Crouch and Stubbs, 2014] [McGregor, 2014] [Chitnis et al., 2015] [Ahn and Guha, 2015] [Esfandiari et al., 2015] [Konrad, 2015] [Bury and Schwiegelshohn, 2015] [Assadi et al., 2016] [Chitnis et al., 2016] [McGregor and Vorotnikova, 2016b] [Esfandiari et al., 2016] [Paz and Schwartzman, 2017] [Ghaffari, 2017] [Kale et al., 2017] ...

Finding Large Matchings

Insertion-only streams:

- Exact computation requires $\Omega(n^2)$ space [Feigenbaum et al., 2005].

Finding Large Matchings

Insertion-only streams:

- Exact computation requires $\Omega(n^2)$ space [Feigenbaum et al., 2005].
- 2-approximation in $O(n)$ space is easy but no better than 2-approximation is known in $o(n^2)$ space.

Finding Large Matchings

Insertion-only streams:

- Exact computation requires $\Omega(n^2)$ space [Feigenbaum et al., 2005].
- 2-approximation in $O(n)$ space is easy but no better than 2-approximation is known in $o(n^2)$ space.
- Beating $e/(e-1)$-approximation requires $n^{1+\Omega(1/\log\log n)}$ space [Goel et al., 2012, Kapralov, 2013].

Finding Large Matchings

Insertion-only streams:

- Exact computation requires $\Omega(n^2)$ space [Feigenbaum et al., 2005].
- 2-approximation in $O(n)$ space is easy but no better than 2-approximation is known in $o(n^2)$ space.
- Beating $e/(e-1)$-approximation requires $n^{1+\Omega(1/\log\log n)}$ space [Goel et al., 2012, Kapralov, 2013].

Dynamic streams:

- $\Omega(n^2/\alpha^3)$ space is necessary for α-approximation [Assadi et al., 2016].
- $\widetilde{O}(n^2/\alpha^3)$ space is sufficient for α-approximation [Assadi et al., 2016, Chitnis et al., 2016].

Estimating Maximum Matching Size

- In general, no better algorithms are known for the seemingly easier problem of estimating the size of a maximum matching.

Estimating Maximum Matching Size

- In general, no better algorithms are known for the seemingly easier problem of estimating the size of a maximum matching.

- However, under certain conditions on the input, sublinear (in n) space algorithms exist:
  - Random arrival insertion-only streams [Kapralov et al., 2014].
  - Bounded arboricity graphs [Esfandiari et al., 2015] ...

Estimating Maximum Matching Size

- In general, no better algorithms are known for the seemingly easier problem of estimating the size of a maximum matching.

- However, under certain conditions on the input, sublinear (in n) space algorithms exist:
  - Random arrival insertion-only streams [Kapralov et al., 2014].
  - Bounded arboricity graphs [Esfandiari et al., 2015] ...

- Lower bounds (Insertion-only streams):
  - $(1+\varepsilon)$-approximation requires $\Omega(n^{1-O(\varepsilon)})$ space [Esfandiari et al., 2015, Bury and Schwiegelshohn, 2015].
  - Deterministic α-approximation requires $\Omega(n/\alpha)$ space [Chakrabarti and Kale, 2016].

Estimating Maximum Matching Size

Question. Is matching size estimation strictly easier than finding an approximate matching?

Estimating Maximum Matching Size

Question. Is matching size estimation strictly easier than finding an approximate matching? Yes!

Estimating Maximum Matching Size

Question. Is matching size estimation strictly easier than finding an approximate matching? Yes!

Theorem

There is a randomized algorithm that outputs an α-approximate estimate of maximum matching size in:

- $\widetilde{O}(n/\alpha^2)$ *space in insertion-only streams.*
- $\widetilde{O}(n^2/\alpha^4)$ *space in dynamic streams.*

Estimating Maximum Matching Size

Question. Is matching size estimation strictly easier than finding an approximate matching? Yes!

Theorem

There is a randomized algorithm that outputs an α-approximate estimate of maximum matching size in:

- $\widetilde{O}(n/\alpha^2)$ *space in insertion-only streams.*
- $\widetilde{O}(n^2/\alpha^4)$ *space in dynamic streams.*

In constrast, to find an α-approximate matching, the space necessary is:

- $\Omega(n/\alpha)$ in insertion-only streams.
- $\Omega(n^2/\alpha^3)$ in dynamic streams.

Algorithms for α-Estimation of Matching Size

The main ingredient of our algorithms is the following sampling lemma:

Lemma (Vertex Sampling Lemma)

Let H be a subgraph of G obtained by sampling each vertex independently w.p. $1/\alpha$. Define:

μ_G*: the maximum matching size in* G,

μ_H*: the maximum matching size in* H.

Then, w.h.p.,

$$\frac{\mu_G}{\alpha^2} \leq \mu_H \leq \frac{2\mu_G}{\alpha}.$$

Algorithms for α-Estimation of Matching Size

The main ingredient of our algorithms is the following sampling lemma:

Lemma (Vertex Sampling Lemma)

Let H be a subgraph of G obtained by sampling each vertex independently w.p. $1/\alpha$. Define:

> μ_G*: the maximum matching size in* G,
>
> μ_H*: the maximum matching size in* H.

Then, w.h.p.,

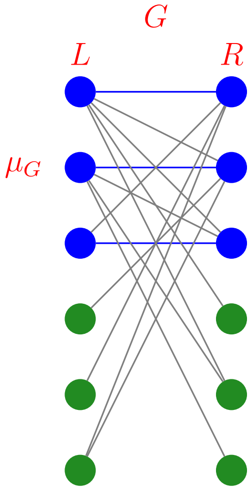
$$\frac{\mu_G}{\alpha^2} \leq \mu_H \leq \frac{2\mu_G}{\alpha}.$$

Therefore, maximum matching size in H is an α-estimation for the maximum matching size in G.

Proof by Picture

Any graph G with a maximum matching size of μ_G looks as follows:

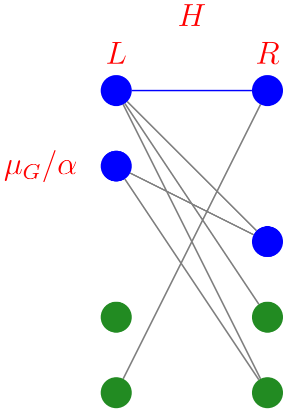
- A matching of size μ_G between the blue vertices.
- No edges between the green vertices.



Proof by Picture

The vertex sampled graph H then look as follows:

- A matching of size μ_G/α^2 between the blue vertices $\implies \mu_H \geq \mu_G/\alpha^2$.
- All edges are incident on μ_G/α blue vertices $\implies \mu_H \leq 2\mu_G/\alpha$.



An α-Estimation Algorithm

To distinguish between graphs with maximum matching of size $\geq k$ and $o(k/\alpha)$:

1. Sample each vertex in G w.p. $1/\alpha$ to obtain H.
2. Test whether H has a matching of size at least $\Omega(k/\alpha^2)$ or not.

An α-Estimation Algorithm

To distinguish between graphs with maximum matching of size $\geq k$ and $o(k/\alpha)$:

1. Sample each vertex in G w.p. $1/\alpha$ to obtain H.
2. Test whether H has a matching of size at least $\Omega(k/\alpha^2)$ or not.

Can be implemented in:

- $\widetilde{O}(k/\alpha^2) = \widetilde{O}(n/\alpha^2)$ in insertion-only streams.
- $\widetilde{O}(k^2/\alpha^4) = \widetilde{O}(n^2/\alpha^4)$ in dynamic streams.

Estimating Maximum Matching Size

Matching size estimation is indeed easier than finding an approximate matching!

Estimating Maximum Matching Size

Matching size estimation is indeed easier than finding an approximate matching!

Question. Is it possible to achieve an arbitrary good estimation of matching size in sub-quadratic space?

Estimating Maximum Matching Size

Matching size estimation is indeed easier than finding an approximate matching!

Question. Is it possible to achieve an arbitrary good estimation of matching size in sub-quadratic space?

Question. In general, what is the space-approximation tradeoff for matching size estimation?

Estimating Maximum Matching Size

Matching size estimation is indeed easier than finding an approximate matching!

Question. Is it possible to achieve an arbitrary good estimation of matching size in sub-quadratic space?

Question. In general, what is the space-approximation tradeoff for matching size estimation?

We make progress on each of these questions.

Our Main Results

Near-optimal approximation of maximum matching size may require almost quadratic space even in insertion-only streams.

Our Main Results

Near-optimal approximation of maximum matching size may require almost quadratic space even in insertion-only streams.

Theorem

Any randomized $(1+\varepsilon)$-approximate estimation of maximum matching size requires:

- $\mathsf{RS}(n) \cdot n^{1-O(\varepsilon)}$ *space in insertion-only streams.*
- $n^{2-O(\varepsilon)}$ *space in dynamic streams.*

Our Main Results

Near-optimal approximation of maximum matching size may require almost quadratic space even in insertion-only streams.

Theorem

Any randomized $(1+\varepsilon)$-approximate estimation of maximum matching size requires:

- $\mathsf{RS}(n) \cdot n^{1-O(\varepsilon)}$ *space in insertion-only streams.*
- $n^{2-O(\varepsilon)}$ *space in dynamic streams.*

$\mathsf{RS}(n)$ denotes the maximum number of edge-disjoint induced matchings of size $\Theta(n)$ in an n-vertex graph:

[Fischer et al., 2002]$n^{\Omega(1/\log\log n)} \leq \mathsf{RS}(n) \leq n/\log n$[Fox et al., 2015]

Our Main Results

We further establish the first non-trivial lower bound for super-constant approximation of matching size.

Our Main Results

We further establish the first non-trivial lower bound for super-constant approximation of matching size.

Theorem

Any randomized α-approximate estimate of maximum matching size requires $\Omega(n/\alpha^2)$ in dynamic streams.
Furthermore, even if we restrict to sparse graphs with arboricity $O(\alpha)$, $\Omega(\sqrt{n}/\alpha^{2.5})$ space is necessary.

Our Main Results

We further establish the first non-trivial lower bound for super-constant approximation of matching size.

Theorem

Any randomized α-approximate estimate of maximum matching size requires $\Omega(n/\alpha^2)$ in dynamic streams.
Furthermore, even if we restrict to sparse graphs with arboricity $O(\alpha)$, $\Omega(\sqrt{n}/\alpha^{2.5})$ space is necessary.

There is an active line of research on estimating matching size of bounded arboricity graphs in graph streams [Chitnis et al., 2016] [Bury and Schwiegelshohn, 2015] [Esfandiari et al., 2015] [McGregor and Vorotnikova, 2016b] [Cormode et al., 2016] [McGregor and Vorotnikova, 2016a] . . .

Schatten p-Norms

Given an $n \times n$ matrix A, for any $p \in [0, \infty)$:

Schatten p-norm of A is the p-th frequency moment of vector of singular values $(\sigma_1, \ldots, \sigma_n)$ of A.

$$\|A\|_p := \left(\sum_{i=1}^{n} \sigma_i^p\right)^{1/p}$$

Schatten p-Norms

Given an $n \times n$ matrix A, for any $p \in [0, \infty)$:

Schatten p-norm of A is the p-th frequency moment of vector of singular values $(\sigma_1, \ldots, \sigma_n)$ of A.

$$\|A\|_p := \left(\sum_{i=1}^{n} \sigma_i^p\right)^{1/p}$$

- $\|A\|_0 =$ Rank of A.
- $\|A\|_1 =$ Trace norm of A.
- $\|A\|_2 =$ Frobenius norm of A.
- $\|A\|_\infty =$ Operator norm of A.

Schatten p-Norms in Data Streams

Question. What is the space complexity of $(1+\varepsilon)$-approximating the Schatten p-norm of a matrix which its entries are revealed in a data stream?

Schatten p-Norms in Data Streams

Question. What is the space complexity of $(1+\varepsilon)$-approximating the Schatten p-norm of a matrix which its entries are revealed in a data stream?

Previous work:

- For $p = 0$, $\Omega(n^{1-g(\varepsilon)})$ space is necessary [Bury and Schwiegelshohn, 2015].
- For $p \in (0, \infty) \setminus 2\mathbb{Z}$, $\Omega(n^{1-g(\varepsilon)})$ space is necessary [Li and Woodruff, 2016].
- For $p \in 2\mathbb{Z} \setminus \{0\}$, $\Omega(n^{1-2/p})$ space is necessary [Li and Woodruff, 2016] (and is sufficient for sparse matrices).

Schatten p-Norms in Data Streams

Question. What is the space complexity of $(1+\varepsilon)$-approximating the Schatten p-norm of a matrix which its entries are revealed in a data stream?

Previous work:

- For $p = 0$, $\Omega(n^{1-g(\varepsilon)})$ space is necessary [Bury and Schwiegelshohn, 2015].
- For $p \in (0, \infty) \setminus 2\mathbb{Z}$, $\Omega(n^{1-g(\varepsilon)})$ space is necessary [Li and Woodruff, 2016].
- For $p \in 2\mathbb{Z} \setminus \{0\}$, $\Omega(n^{1-2/p})$ space is necessary [Li and Woodruff, 2016] (and is sufficient for sparse matrices).

We answer this question for the case of rank computation.

Matrix Rank Computation in Data Streams

It is well-known that computing maximum matching size of a graph is equivalent to computing the rank of the (symbolic) Tutte matrix.

Matrix Rank Computation in Data Streams

It is well-known that computing maximum matching size of a graph is equivalent to computing the rank of the (symbolic) Tutte matrix.

As a corollary, all our lower bounds for matching size estimation also extend to the matrix rank computation problem. In particular,

- An $\Omega(n^{2-O(\varepsilon)})$ space lower bound for $(1+\varepsilon)$-estimation of rank in dense matrices.
- An $\widetilde{\Omega}(\sqrt{n})$ space lower bound for any $\mathrm{polylog}(n)$-estimation of rank in sparse matrices.

An $\Omega(n^{2-O(\varepsilon)})$ Lower Bound for Dynamic Streams

Theorem

Any randomized $(1+\varepsilon)$-approximate estimation of maximum matching size requires $\Omega(n^{2-O(\varepsilon)})$ space in dynamic streams.

Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

Consider the following two-player one-way communication problem.

MaxMatching:

1. Alice is given a matching M on vertices V.
2. Bob is given a collection of edges E_B on vertices V.
3. Alice sends a single message to Bob and Bob outputs an estimation of maximum matching size in $G(V, M \cup E_B)$.

Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

Consider the following two-player one-way communication problem.

MaxMatching:

1. Alice is given a matching M on vertices V.
2. Bob is given a collection of edges E_B on vertices V.
3. Alice sends a single message to Bob and Bob outputs an estimation of maximum matching size in $G(V, M \cup E_B)$.

CC(MaxMatching): minimum length message to solve this problem with probability, say, $2/3$.

Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

Consider the following two-player one-way communication problem.

MaxMatching:

1. Alice is given a matching M on vertices V.
2. Bob is given a collection of edges E_B on vertices V.
3. Alice sends a single message to Bob and Bob outputs an estimation of maximum matching size in $G(V, M \cup E_B)$.

CC(MaxMatching): minimum length message to solve this problem with probability, say, $2/3$.

Fact. CC(MaxMatching) $\leq$ space complexity of any streaming algorithm for estimating maximum matching size.

Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

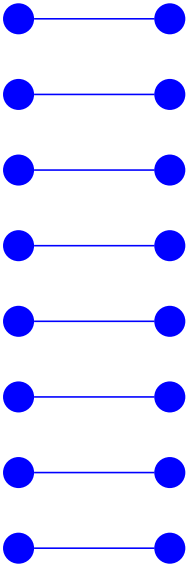
[Bury and Schwiegelshohn, 2015]:
$\mathsf{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

[Bury and Schwiegelshohn, 2015]:
$\text{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Proof Sketch. (for $\varepsilon = 1/2$)

- Alice is given a random subset of size $n/2$ from a fixed perfect matching between L and R.

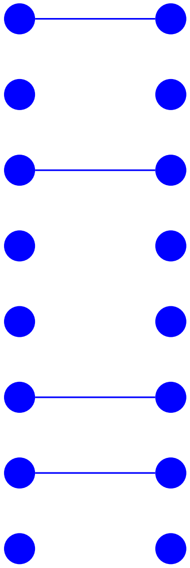


Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

[Bury and Schwiegelshohn, 2015]:
$\text{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Proof Sketch. (for $\varepsilon = 1/2$)

- Alice is given a random subset of size $n/2$ from a fixed perfect matching between L and R.

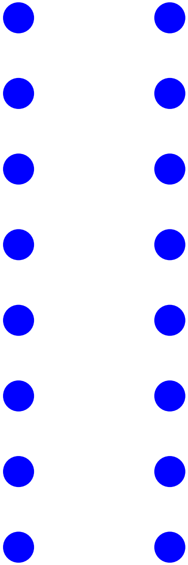


Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

[Bury and Schwiegelshohn, 2015]:
$\text{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Proof Sketch. (for $\varepsilon = 1/2$)

- Alice is given a random subset of size $n/2$ from a fixed perfect matching between L and R.
- Bob is given a matching of size $n/2$ incident on R.

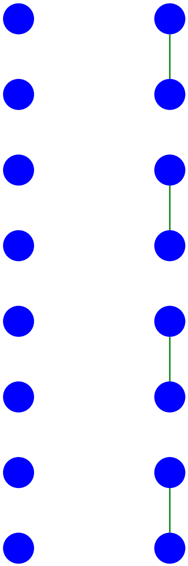


Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

[Bury and Schwiegelshohn, 2015]:
$\text{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Proof Sketch. (for $\varepsilon = 1/2$)

- Alice is given a random subset of size $n/2$ from a fixed perfect matching between L and R.
- Bob is given a matching of size $n/2$ incident on R.

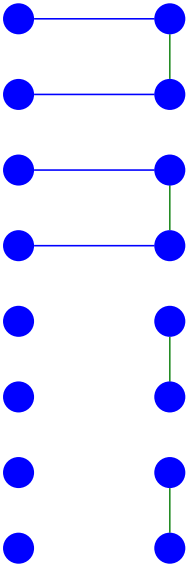


Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

[Bury and Schwiegelshohn, 2015]:
$\mathrm{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Proof Sketch. (for $\varepsilon = 1/2$)

- Alice is given a random subset of size $n/2$ from a fixed perfect matching between L and R.
- Bob is given a matching of size $n/2$ incident on R.
- Yes case: Each edge of Bob's matching is incident on even number of Alice's matching $\implies$ $\textsc{MaxMatching} = 3n/4$.

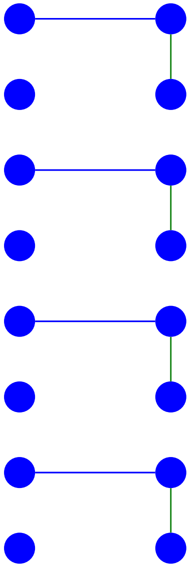


Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

[Bury and Schwiegelshohn, 2015]:
$\text{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Proof Sketch. (for $\varepsilon = 1/2$)

- Alice is given a random subset of size $n/2$ from a fixed perfect matching between L and R.
- Bob is given a matching of size $n/2$ incident on R.
- No case: Each edge of Bob's matching is incident on odd number of Alice's matching $\implies$ $\textsc{MaxMatching} = n/2$.

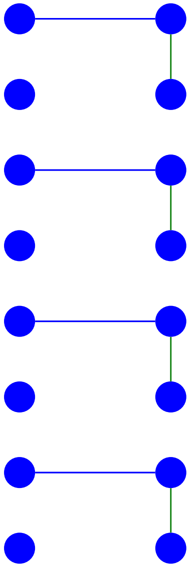


Previous Approaches: An $n^{1-O(\varepsilon)}$ Lower Bound

[Bury and Schwiegelshohn, 2015]:
$\text{CC}(\textsc{MaxMatching}) = \Omega(n^{1-O(\varepsilon)})$.

Proof Sketch. (for $\varepsilon = 1/2$)

- A better than $3/2$-approximation distinguishes between the two cases.
- Distinguishing between the two cases requires $\Omega(\sqrt{n})$ communication by a reduction from the boolean hidden matching problem of [Gavinsky et al., 2007].



Our Approach

A natural idea to boost the previous lower bound:

1. Instead of one matching M, provide Alice with t independently chosen matchings $M_1, \ldots, M_t$.

Our Approach

A natural idea to boost the previous lower bound:

1. Instead of one matching M, provide Alice with t independently chosen matchings $M_1, \ldots, M_t$.

2. Provide Bob with a single set E_B of edges as before.

Our Approach

A natural idea to boost the previous lower bound:

1. Instead of one matching M, provide Alice with t independently chosen matchings $M_1, \ldots, M_t$.
2. Provide Bob with a single set E_B of edges as before.
3. "Ask" Alice and Bob to solve the MaxMatching problem for a uniformly at random chosen matching $M_{j^\star}$ and E_B (the index $j^\star$ is unknown to Alice).

Our Approach

A natural idea to boost the previous lower bound:

1. Instead of one matching M, provide Alice with t independently chosen matchings $M_1, \ldots, M_t$.
2. Provide Bob with a single set E_B of edges as before.
3. "Ask" Alice and Bob to solve the MaxMatching problem for a uniformly at random chosen matching $M_{j^\star}$ and E_B (the index $j^\star$ is unknown to Alice).

The hope is that communication complexity of this problem is now $\geq t \cdot \text{CC}(\text{MaxMatching})$.

Our Approach

There are three main obstacles in implementing this idea:

1. The matchings $M_1, \ldots, M_t$ should be supported on $\Theta(n)$ vertices as opposed to trivial $\Theta(t \cdot n)$ vertices.

Our Approach

There are three main obstacles in implementing this idea:

1. The matchings $M_1, \ldots, M_t$ should be supported on $\Theta(n)$ vertices as opposed to trivial $\Theta(t \cdot n)$ vertices.

2. The matchings should be chosen independently even though they are supported on the same set of $\Theta(n)$ vertices.

Our Approach

There are three main obstacles in implementing this idea:

1. The matchings $M_1, \ldots, M_t$ should be supported on $\Theta(n)$ vertices as opposed to trivial $\Theta(t \cdot n)$ vertices.

2. The matchings should be chosen independently even though they are supported on the same set of $\Theta(n)$ vertices.

3. The reduction should ensure that Alice and Bob indeed need to solve the $j^{\star}$-th embedded instance.

Our Approach

There are three main obstacles in implementing this idea:

1. The matchings $M_1, \ldots, M_t$ should be supported on $\Theta(n)$ vertices as opposed to trivial $\Theta(t \cdot n)$ vertices.
2. The matchings should be chosen independently even though they are supported on the same set of $\Theta(n)$ vertices.
3. The reduction should ensure that Alice and Bob indeed need to solve the $j^\star$-th embedded instance.

(1) + (2) $\implies$ Ruzsa-Szemerédi graphs (RS graphs).

Our Approach

There are three main obstacles in implementing this idea:

1. The matchings $M_1, \ldots, M_t$ should be supported on $\Theta(n)$ vertices as opposed to trivial $\Theta(t \cdot n)$ vertices.
2. The matchings should be chosen independently even though they are supported on the same set of $\Theta(n)$ vertices.
3. The reduction should ensure that Alice and Bob indeed need to solve the $j^\star$-th embedded instance.

(1) + (2) $\implies$ Ruzsa-Szemerédi graphs (RS graphs).

RS graphs + (3) $\implies$ characterization of dynamic streaming algorithms via simultaneous communication complexity.

Our Approach

There are three main obstacles in implementing this idea:

1. The matchings $M_1, \ldots, M_t$ should be supported on $\Theta(n)$ vertices as opposed to trivial $\Theta(t \cdot n)$ vertices.

2. The matchings should be chosen independently even though they are supported on the same set of $\Theta(n)$ vertices.

3. The reduction should ensure that Alice and Bob indeed need to solve the $j^\star$-th embedded instance.

(1) + (2) $\implies$ Ruzsa-Szemerédi graphs (RS graphs).

RS graphs + (3) $\implies$ characterization of dynamic streaming algorithms via simultaneous communication complexity.

Formalizing the lower bound $\implies$ a direct-sum style argument using information complexity.

Ruzsa-Szemerédi Graphs

Definition ((r,t)-RS graphs)

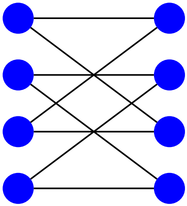
A graph $G(V,E)$ whose edges can be partitioned into t induced matchings of size r each.

Ruzsa-Szemerédi Graphs

Definition ((r, t)-RS graphs)

A graph $G(V, E)$ whose edges can be partitioned into t induced matchings of size r each.

1. Example. A $(2, 4)$-RS graph on 8 vertices:

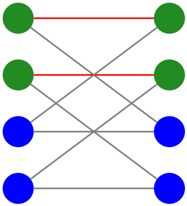


Ruzsa-Szemerédi Graphs

Definition ((r,t)-RS graphs)

A graph $G(V,E)$ whose edges can be partitioned into t induced matchings of size r each.

1. Example. A $(2,4)$-RS graph on 8 vertices:

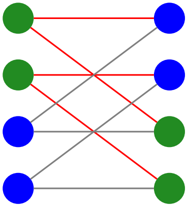


Ruzsa-Szemerédi Graphs

Definition ((r,t)-RS graphs)

A graph $G(V,E)$ whose edges can be partitioned into t induced matchings of size r each.

1. Example. A $(2,4)$-RS graph on 8 vertices:

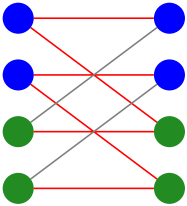


Ruzsa-Szemerédi Graphs

Definition ((r, t)-RS graphs)

A graph $G(V, E)$ whose edges can be partitioned into t induced matchings of size r each.

1. Example. A $(2, 4)$-RS graph on 8 vertices:

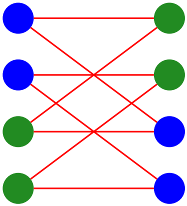


Ruzsa-Szemerédi Graphs

Definition ((r, t)-RS graphs)

A graph $G(V, E)$ whose edges can be partitioned into t induced matchings of size r each.

1. Example. A $(2, 4)$-RS graph on 8 vertices:



Ruzsa-Szemerédi Graphs

We are typically interested in RS graphs with large values of r and t as a function of n.

Ruzsa-Szemerédi Graphs

We are typically interested in RS graphs with large values of r and t as a function of n.

How dense a graph with many large induced matching can be?

Ruzsa-Szemerédi Graphs

We are typically interested in RS graphs with large values of r and t as a function of n.

How dense a graph with many large induced matching can be?

Theorem ([Fischer et al., 2002])

There exists an (r,t)-RS graph on n vertices with $t = n^{\Omega(1/\log\log n)}$ induced matchings of size $r = (1-\varepsilon) \cdot n/4$.

Ruzsa-Szemerédi Graphs

We are typically interested in RS graphs with large values of r and t as a function of n.

How dense a graph with many large induced matching can be?

Theorem ([Fischer et al., 2002])

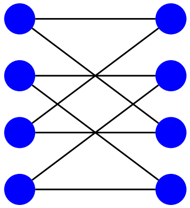
There exists an (r,t)-RS graph on n vertices with $t = n^{\Omega(1/\log\log n)}$ induced matchings of size $r = (1-\varepsilon)\cdot n/4$.

Theorem ([Alon et al., 2012])

There exists an (r,t)-RS graph on n vertices with $t = n^{1+o(1)}$ induced matchings of size $r = n^{1-o(1)}$.

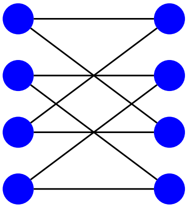
A Simple Lower Bound for Insertion-Only Streams

- Let G_1 be an (r, t)-RS bipartite graph on n vertices on each side.



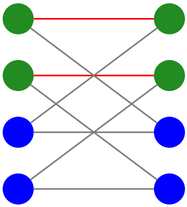
A Simple Lower Bound for Insertion-Only Streams

- Let G_1 be an (r, t)-RS bipartite graph on n vertices on each side.
- To Alice, we give random subset of size $r/2$ from each induced matchings $M_1, \ldots, M_t$ of G_1.



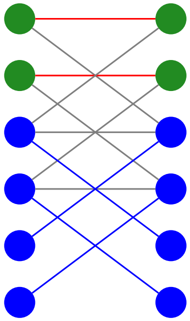
A Simple Lower Bound for Insertion-Only Streams

- Let G_1 be an (r, t)-RS bipartite graph on n vertices on each side.
- To Alice, we give random subset of size $r/2$ from each induced matchings $M_1, \ldots, M_t$ of G_1.
- Choose $M_{j^\star}$ uniformly at random.



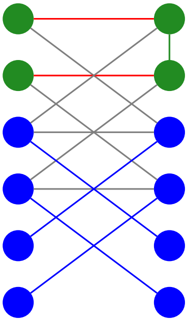
A Simple Lower Bound for Insertion-Only Streams

- Let G_1 be an (r,t)-RS bipartite graph on n vertices on each side.
- To Alice, we give random subset of size $r/2$ from each induced matchings $M_1, \ldots, M_t$ of G_1.
- Choose $M_{j^\star}$ uniformly at random. To Bob we give the following input:
  - A matching between vertices not in $M_{j^\star}$ and a new set of vertices.



A Simple Lower Bound for Insertion-Only Streams

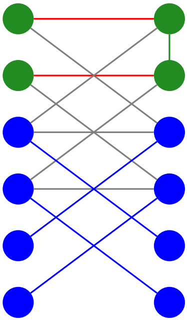
- Let G_1 be an (r,t)-RS bipartite graph on n vertices on each side.
- To Alice, we give random subset of size $r/2$ from each induced matchings $M_1, \ldots, M_t$ of G_1.
- Choose $M_{j^\star}$ uniformly at random. To Bob we give the following input:
  - A matching between vertices not in $M_{j^\star}$ and a new set of vertices.
  - A graph E_B over the set of vertices in $M_{j^\star}$.



A Simple Lower Bound for Insertion-Only Streams

- Size of the maximum matching in this graph:

  $2(n-r) + \textsc{MaxMatching}(M_{j^\star}, E_B)$

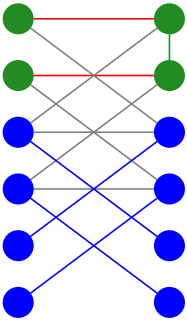


A Simple Lower Bound for Insertion-Only Streams

- Size of the maximum matching in this graph:

  $2(n-r) + \textsc{MaxMatching}(M_{j^\star}, E_B)$

- For $r = \Theta(n)$, Alice and Bob need to solve $\textsc{MaxMatching}(M_{j^\star}, E_B)$ for $(1+\varepsilon)$-approximation.



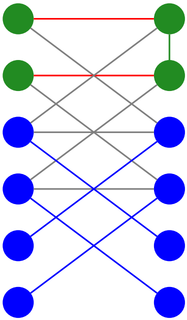
A Simple Lower Bound for Insertion-Only Streams

- Size of the maximum matching in this graph:

  $2(n-r) + \textsc{MaxMatching}(M_{j^\star}, E_B)$

- For $r = \Theta(n)$, Alice and Bob need to solve $\textsc{MaxMatching}(M_{j^\star}, E_B)$ for $(1+\varepsilon)$-approximation.

- To solve this for an unknown matching $M_{j^\star}$, the message length must be $\geq t \cdot \text{CC}(\textsc{MaxMatching})$.



A Simple Lower Bound for Insertion-Only Streams

Main limitation of this approach:

- Requires $r = \Theta(n) \implies t = \mathsf{RS}(n)$.
- $\mathsf{RS}(n)$ maybe as large as $n/\log n$.
- However, best known bound for $\mathsf{RS}(n)$ is only $n^{\Omega(1/\log\log n)}$.

A Simple Lower Bound for Insertion-Only Streams

Main limitation of this approach:

- Requires $r = \Theta(n) \implies t = \mathsf{RS}(n)$.
- $\mathsf{RS}(n)$ maybe as large as $n/\log n$.
- However, best known bound for $\mathsf{RS}(n)$ is only $n^{\Omega(1/\log\log n)}$.

We bypass the $r = \Theta(n)$ limitation in dynamic streams using the characterization result of [Li et al., 2014, Ai et al., 2016] in terms of the simultaneous communication complexity.

The Simultaneous Communication Model

- The input is partitioned between k players $P^{(1)}, \ldots, P^{(k)}$.

The Simultaneous Communication Model

- The input is partitioned between k players $P^{(1)}, \ldots, P^{(k)}$.
- There exists an additional party called the referee.

The Simultaneous Communication Model

- The input is partitioned between k players $P^{(1)}, \ldots, P^{(k)}$.
- There exists an additional party called the referee.
- Players $P^{(1)}, \ldots, P^{(k)}$ simultaneously send a message to the referee who outputs the answer.

The Simultaneous Communication Model

- The input is partitioned between k players $P^{(1)}, \ldots, P^{(k)}$.
- There exists an additional party called the referee.
- Players $P^{(1)}, \ldots, P^{(k)}$ simultaneously send a message to the referee who outputs the answer.
- The players have access to public random coins.

The Simultaneous Communication Model

- The input is partitioned between k players $P^{(1)}, \ldots, P^{(k)}$.
- There exists an additional party called the referee.
- Players $P^{(1)}, \ldots, P^{(k)}$ simultaneously send a message to the referee who outputs the answer.
- The players have access to public random coins.
- Communication complexity measure: maximum number of bits sent by any player.

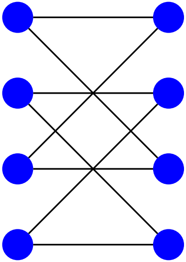
The Simultaneous Communication Model

- The input is partitioned between k players $P^{(1)}, \ldots, P^{(k)}$.
- There exists an additional party called the referee.
- Players $P^{(1)}, \ldots, P^{(k)}$ simultaneously send a message to the referee who outputs the answer.
- The players have access to public random coins.
- Communication complexity measure: maximum number of bits sent by any player.

[Ai et al., 2016]: Communication lower bounds in this model imply identical space lower bound for dynamic streaming algorithms.

A Hard Input Distribution

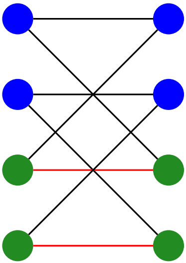
- Each player is given an (r, t)-RS graph on N vertices.



Local view of P^i

A Hard Input Distribution

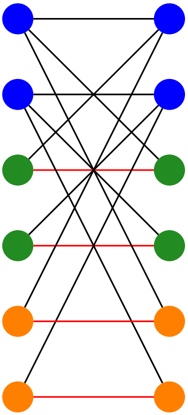
- Each player is given an (r, t)-RS graph on N vertices.
- One of the induced matching $M_{j^\star}^{(i)}$ of each player $P^{(i)}$'s graph is special, unknown to the player.



Special matching of P^i

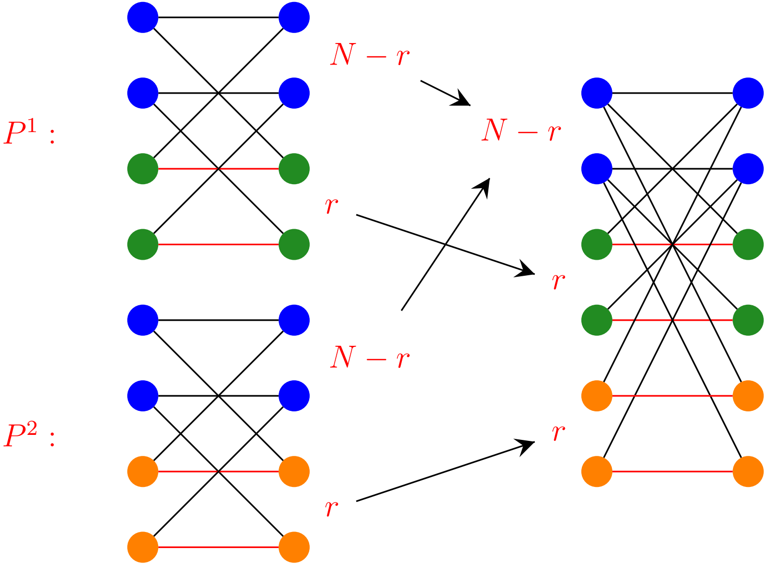
A Hard Input Distribution

- Each player is given an (r, t)-RS graph on N vertices.
- One of the induced matching $M_{j^\star}^{(i)}$ of each player $P^{(i)}$'s graph is special, unknown to the player.
- Across the players, vertices in the special matchings are unique, while other vertices are shared.



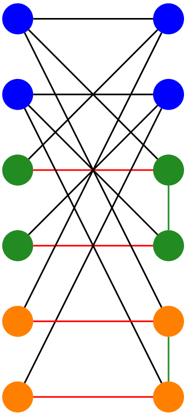
Global view

A Hard Input Distribution



A Hard Input Distribution

- Each player is given an (r,t)-RS graph on N vertices.
- One of the induced matching $M^{(i)}_{j^\star}$ of each player $P^{(i)}$'s graph is special, unknown to the player.
- Across the players, vertices in the special matchings are unique, while other vertices are shared.
- To the referee, we provide k subgraphs $E^{(1)}_B, \ldots, E^{(k)}_B$ such that each pair $(M^{(i)}_{j^\star}, E^{(i)}_B)$ forms the same instance of MaxMatching.



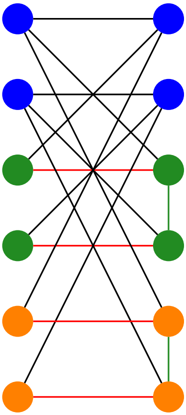
Global view

Proof Sketch

- Define $\textsc{MaxMatching}(M, E_B) := \textsc{MaxMatching}(M^{(1)}_{j^\star}, E^{(1)}_B) = \ldots = \textsc{MaxMatching}(M^{(k)}_{j^\star}, E^{(k)}_B)$.
- The maximum matching size in G is:

  $$\approx 2(N - r) + k \cdot \textsc{MaxMatching}(M, E_B)$$

- For $r = N^{1-o(1)}$ and $k \approx \frac{1}{\varepsilon} \cdot N^{o(1)}$, $\textsc{MaxMatching}(M, E_B)$ is the dominating term for $(1 + \varepsilon)$-approximation.



Global view

Proof Sketch

- The players need to solve $\textsc{MaxMatching}(M, E_B)$.

Proof Sketch

- The players need to solve $\textsc{MaxMatching}(M, E_B)$.
- **Key Lemma.** Each player can reveal at most $\approx s/t$ bits of information about M, by sending a message of size s.

Proof Sketch

- The players need to solve $\textsc{MaxMatching}(M, E_B)$.
- **Key Lemma.** Each player can reveal at most $\approx s/t$ bits of information about M, by sending a message of size s.
  - The players are oblivious to the identity of their special matching.

Proof Sketch

- The players need to solve $\textsc{MaxMatching}(M, E_B)$.
- **Key Lemma.** Each player can reveal at most $\approx s/t$ bits of information about M, by sending a message of size s.
  - The players are oblivious to the identity of their special matching.
- To solve $\textsc{MaxMatching}(M, E_B)$, the referee needs to receive $\Omega(|M|^{1-O(\varepsilon)}) = \Omega(N^{1-O(\varepsilon)})$ bits of information.

Proof Sketch

- The players need to solve $\textsc{MaxMatching}(M, E_B)$.
- **Key Lemma.** Each player can reveal at most $\approx s/t$ bits of information about M, by sending a message of size s.
  - The players are oblivious to the identity of their special matching.
- To solve $\textsc{MaxMatching}(M, E_B)$, the referee needs to receive $\Omega(|M|^{1-O(\varepsilon)}) = \Omega(N^{1-O(\varepsilon)})$ bits of information.
- To conclude,

$$N^{1-O(\varepsilon)} \leq k \cdot s/t \implies s \geq \frac{1}{k} \cdot t \cdot N^{1-O(\varepsilon)} \approx N^{2-O(\varepsilon)}$$

as $t = N^{1+o(1)}$ by [Alon et al., 2012] and $k = \Theta_\varepsilon(N^{o(1)})$.

Wrap Up

- We proved that simultaneous communication complexity of $(1+\varepsilon)$-estimation of maximum matching size is $\Omega(n^{2-O(\varepsilon)})$.

Wrap Up

- We proved that simultaneous communication complexity of $(1+\varepsilon)$-estimation of maximum matching size is $\Omega(n^{2-O(\varepsilon)})$.
- By result of [Ai et al., 2016], this implies an identical lower bound for space complexity of dynamic streaming algorithms.

Wrap Up

- We proved that simultaneous communication complexity of $(1+\varepsilon)$-estimation of maximum matching size is $\Omega(n^{2-O(\varepsilon)})$.
- By result of [Ai et al., 2016], this implies an identical lower bound for space complexity of dynamic streaming algorithms.

Theorem

Any randomized $(1+\varepsilon)$-approximate estimation of maximum matching size requires $\Omega(n^{2-O(\varepsilon)})$ space in dynamic streams.

Concluding Remarks

- Matching size estimation is provably easier than finding an approximate matching.

Concluding Remarks

- Matching size estimation is provably easier than finding an approximate matching.
- However, the space complexity of both problems converges together to quadratic space as the desired accuracy approaches one.

Concluding Remarks

- Matching size estimation is provably easier than finding an approximate matching.
- However, the space complexity of both problems converges together to quadratic space as the desired accuracy approaches one.

Open problems.

- Non-trivial space lower bounds for, say, poly-log approximation in insertion-only streams?

Concluding Remarks

- Matching size estimation is provably easier than finding an approximate matching.
- However, the space complexity of both problems converges together to quadratic space as the desired accuracy approaches one.

Open problems.

- Non-trivial space lower bounds for, say, poly-log approximation in insertion-only streams?
- The exact space-approximation tradeoff for matching size estimation in dynamic streams?
  - $\Omega(n/\alpha^2)$ space is necessary vs. $\widetilde{O}(n^2/\alpha^4)$ space is sufficient.

Concluding Remarks

- Matching size estimation is provably easier than finding an approximate matching.
- However, the space complexity of both problems converges together to quadratic space as the desired accuracy approaches one.

Open problems.

- Non-trivial space lower bounds for, say, poly-log approximation in insertion-only streams?
- The exact space-approximation tradeoff for matching size estimation in dynamic streams?
  - $\Omega(n/\alpha^2)$ space is necessary vs. $\widetilde{O}(n^2/\alpha^4)$ space is sufficient.
- Similar-in-spirit lower bounds for Schatten p-norms for $p > 0$?

Ahn, K. J. and Guha, S. (2013).
Linear programming in the semi-streaming model with application to the maximum matching problem.
Inf. Comput., 222:59–79.

Ahn, K. J. and Guha, S. (2015).
Access to data and number of iterations: Dual primal algorithms for maximum matching under resource constraints.
In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 202–211.

Ahn, K. J., Guha, S., and McGregor, A. (2012).
Graph sketches: sparsification, spanners, and subgraphs.
In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 5–14.

Ai, Y., Hu, W., Li, Y., and Woodruff, D. P. (2016).

New characterizations in turnstile streams with applications.
In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 20:1–20:22.

Alon, N., Matias, Y., and Szegedy, M. (1996).
The space complexity of approximating the frequency moments.
In *STOC*, pages 20–29. ACM.

Alon, N., Moitra, A., and Sudakov, B. (2012).
Nearly complete graphs decomposable into large induced matchings and their applications.
In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1079–1090.

Assadi, S., Khanna, S., Li, Y., and Yaroslavtsev, G. (2016).
Maximum matchings in dynamic graph streams and the simultaneous communication model.

In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1345–1364.

Bury, M. and Schwiegelshohn, C. (2015).
Sublinear estimation of weighted matchings in dynamic data streams.
In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 263–274.

Chakrabarti, A. and Kale, S. (2016).
Strong fooling sets for multi-player communication with applications to deterministic estimation of stream statistics.
In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 41–50.

Chitnis, R., Cormode, G., Esfandiari, H., Hajiaghayi, M., McGregor, A., Monemizadeh, M., and Vorotnikova, S. (2016).
Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams.
In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1326–1344.

Chitnis, R. H., Cormode, G., Hajiaghayi, M. T., and Monemizadeh, M. (2015).
Parameterized streaming: Maximal matching and vertex cover.
In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1234–1251.

Cormode, G., Jowhari, H., Monemizadeh, M., and Muthukrishnan, S. (2016).

The sparse awakens: Streaming algorithms for matching size estimation in sparse graphs.
CoRR, abs/1608.03118.

Crouch, M. and Stubbs, D. S. (2014).
Improved streaming algorithms for weighted matching, via unweighted matching.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pages 96–104.

Eggert, S., Kliemann, L., and Srivastav, A. (2009).
Bipartite graph matchings in the semi-streaming model.
In *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, pages 492–503.

Epstein, L., Levin, A., Mestre, J., and Segev, D. (2011).

Improved approximation guarantees for weighted matching in the semi-streaming model.
SIAM J. Discrete Math., 25(3):1251–1265.

Esfandiari, H., Hajiaghayi, M., and Monemizadeh, M. (2016).
Finding large matchings in semi-streaming.
In *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain.*, pages 608–614.

Esfandiari, H., Hajiaghayi, M. T., Liaghat, V., Monemizadeh, M., and Onak, K. (2015).
Streaming algorithms for estimating the matching size in planar graphs and beyond.
In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1217–1233.

Feigenbaum, J., Kannan, S., McGregor, A., Suri, S., and Zhang, J. (2005).
On graph problems in a semi-streaming model.
Theor. Comput. Sci., 348(2-3):207–216.

Fischer, E., Lehman, E., Newman, I., Raskhodnikova, S., Rubinfeld, R., and Samorodnitsky, A. (2002).
Monotonicity testing over general poset domains.
In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483.

Fox, J., Huang, H., and Sudakov, B. (2015).
On graphs decomposable into induced matchings of linear sizes.
arXiv preprint arXiv:1512.07852.

Gavinsky, D., Kempe, J., Kerenidis, I., Raz, R., and de Wolf, R. (2007).

Exponential separations for one-way quantum communication complexity, with applications to cryptography.
STOC, pages 516–525.

Ghaffari, M. (2017).
Space-optimal semi-streaming for $(2+\epsilon)$-approximate matching.
CoRR, abs/1701.03730.

Goel, A., Kapralov, M., and Khanna, S. (2012).
On the communication and streaming complexity of maximum bipartite matching.
In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 468–485. SIAM.

Guruswami, V. and Onak, K. (2013).
Superlinear lower bounds for multipass graph processing.

In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 287–298.

Kale, S., Tirodkar, S., and Vishwanathan, S. (2017).
Maximum matching in two, three, and a few more passes over graph streams.
CoRR, abs/1702.02559.

Kapralov, M. (2013).
Better bounds for matchings in the streaming model.
In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697.

Kapralov, M., Khanna, S., and Sudan, M. (2014).
Approximating matching size from random streams.

In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751.

Konrad, C. (2015).
Maximum matching in turnstile streams.
In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852.

Konrad, C., Magniez, F., and Mathieu, C. (2012).
Maximum matching in semi-streaming with few passes.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 231–242.

Li, Y., Nguyen, H. L., and Woodruff, D. P. (2014).

Turnstile streaming algorithms might as well be linear sketches.
In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 174–183.

Li, Y. and Woodruff, D. P. (2016).
On approximating functions of the singular values in a stream.
In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 726–739.

McGregor, A. (2005).
Finding graph matchings in data streams.
In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, pages 170–181.

McGregor, A. (2014).
Graph stream algorithms: a survey.
SIGMOD Record, 43(1):9–20.

McGregor, A. and Vorotnikova, S. (2016a).
A note on logarithmic space stream algorithms for matchings in low arboricity graphs.
CoRR, abs/1612.02531.

McGregor, A. and Vorotnikova, S. (2016b).
Planar matching in streams revisited.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2016, September 7-9, 2016, Paris, France*, pages 17:1–17:12.

Paz, A. and Schwartzman, G. (2017).
A $(2 + \varepsilon)$-approximation for maximum weight matching in the semi-streaming model.

In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2153–2161.

Zelke, M. (2012).
Weighted matching in the semi-streaming model.
Algorithmica, 62(1-2):1–20.