Sublinear Estimation of Weighted Matchings in Dynamic Data Streams*

Marc Bury, Chris Schwiegelshohn

Efficient Algorithms and Complexity Theory, TU Dortmund, Germany.
{firstname.lastname}@tu-dortmund.de

Abstract. This paper presents an algorithm for estimating the weight of a maximum weighted matching by augmenting any estimation routine for the size of an unweighted matching. The algorithm is implementable in any streaming model including dynamic graph streams. We also give the first constant estimation for the maximum matching size in a dynamic graph stream for planar graphs (or any graph with bounded arboricity) using $\tilde{O}(n^{4/5})$ space. Using previous results by Kapralov, Khanna, and Sudan (2014) we obtain a constant approximation for the value of the maximum weighted matching in $\tilde{O}(n^{4/5})$ space for dynamic streams and a polylog(n) approximation for general graphs using polylog(n) space in random order streams, respectively. In addition, we give a space lower bound of $\Omega(n^{1-\varepsilon})$ for any randomized algorithm estimating the size of a maximum matching up to a $1 + O(\varepsilon)$ factor for adversarial streams.

1 Introduction

Large graph structures encountered in social networks or the web-graph have become focus of analysis both from theory and practice. To process such large input, conventional algorithms often require an infeasible amount of running time, space or both, giving rise to other models of computation. Much theoretical research focuses on the streaming model where the input arrives one by one with the goal of storing as much information as possible in small, preferably polylogarithmic, space. Streaming algorithms on graphs were first studied by Henzinger et al. [17], who showed that even simple problems often admit no solution with such small space requirements. The semi-streaming model [14] where the stream consists of the edges of a graph and the algorithm is allowed $O(n \cdot \text{polylog}(n))$ space and allows few (ideally just one) passes over the data relaxes these requirements and has received considerable attention. Problems studied in the semi-streaming model include sparsification, spanners, connectivity, minimum spanning trees, counting triangles and matching, for an overview we refer to a recent survey by McGregor [27]. Due to the fact that graphs motivating this research are dynamic structures that change over time there has recently been research on streaming algorithms supporting deletions. We now review the literature on streaming algorithms for matching and dynamic streams.

* Supported by Deutsche Forschungsgemeinschaft, grant BO 2755/1-2 and within the Collaborative Research Center SFB 876, project A2

Matching Maintaining a 2 approximation to the maximum matching (MM) in an insertion-only stream can be straightforwardly done by greedily maintaining a maximal matching [14]. Improving on this algorithm turns out to be difficult as Goel et al. [16] showed that no algorithm using $\tilde{O}(n)$ space can achieve an approximation ratio better than $\frac{3}{2}$ which was improved by Kapralov to $\frac{e}{e-1}$ [19]. Konrad et al. [22] gave an algorithm using $\tilde{O}(n)$ space with an approximation factor of 1.989 if the edges are assumed to arrive in random order. For weighted matching (MWM), a series of results have been published [14,26,11,32,12] with the current best bound of $4 + \varepsilon$ being due to Crouch and Stubbs [10].

To bypass the natural $\Omega(n)$ bound required by any algorithm maintaining an approximate matching, recent research has begun to focus on estimating the size of the maximum matching. Kapralov et al. [20] gave a polylogrithmic approximate estimate using polylogarithmic space for random order streams. For certain sparse graphs including planar graphs, Esfandiari et al. [13] describe how to obtain a constant factor estimation using $\tilde{O}(n^{2/3})$ space in a single pass and $\tilde{O}(\sqrt{n})$ space using two passes or assuming randomly ordered streams. The authors also gave a lower bound of $\Omega(\sqrt{n})$ for any approximation better than $\frac{3}{2}$.

Dynamic Streams In the turnstile model, the stream consists of a sequence of additive updates to a vector. Problems studied in this model include numerical linear algebra problems such as regression and low-rank approximation, and maintaining certain statistics of a vector like frequency moments, heavy hitters or entropy. Linear sketches have proven to be the algorithmic technique of choice and might as well be the only algorithmic tool able to efficiently do so, see Li, Nguyen and Woodruff [24]. Dynamic graphs as introduced and studied by Ahn, Guha and McGregor [1,2,3,4] are similar to, but weaker than turnstile updates (see also Section 1.1). Though both streaming models assume update to the input matrix, there usually exists a consistency assumption for streams, i.e. at any given time the multiplicity of an edge is either 0 or 1 and edge weights cannot change arbitrarily but are first set to 0 and then reinserted with the desired weight The authors extend some of the aforementioned problems such as connectivity, sparsification and minimum spanning trees to this setting. Recent results by Assadi et al. [5] showed that approximating matchings in dynamic streams is hard by providing a space lower bound of $\Omega(n^{2-3\varepsilon})$ for approximating the maximum matching within a factor of $\tilde{O}(n^{\varepsilon})$. Simultaneously, Konrad [21] showed a similar but slightly weaker lower bound of $\Omega(n^{3/2-4\varepsilon})$. Both works presented an algorithm with an almost matching upper bound on the space complexity of $\tilde{O}(n^{2-2\varepsilon})$ [21] and $\tilde{O}(n^{2-3\varepsilon})$ [5]. Chitnis et al. [7] gave a streaming algorithm using $\tilde{O}(k^2)$ space that returns an exact maximum matching under the assumption that the size is at most k. It is important to note that all these results actually compute a matching. In terms of estimating the size of the maximum matching, Chitnis et al. [7] extended the estimation algorithms for sparse graphs from [13] to the settings of dynamic streams using $\tilde{O}(n^{4/5})$ space. A bridge between dynamic graphs and the insertion-only streaming model is the sliding window model studied by Crouch et al. [9]. The authors give a $(3 + \varepsilon)$-approximation algorithm for maximum matching.

The p-Schatten norm of a matrix A is defined as the ℓ_p-norm of the vector of singular values. It is well known that computing the maximum matching size is equivalent to computing the rank of the Tutte matrix [29,25] (see also Section 2.1). Estimating the maximum matching size therefore is a special case of estimating the rank or 0-Schatten norm of a matrix. Li, Nguyen and Woodruff gave strong lower bounds on the space requirement for estimating Schatten norms in dynamic streams [23]. Any estimation of the rank within any constant factor is shown to require $\Omega(n^2)$ space when using bi-linear sketches and $\Omega(\sqrt{n})$ space for general linear sketches.

	Reference	Graph class	Streaming model	Approx. factor	Space
MM:	Greedy	General	Adversarial	2	$O(n)$
	[20]	General	Random	polylog(n)	polylog(n)
	[13]	Trees	Adversarial	$2+\varepsilon$	$\tilde{O}(\sqrt{n})$
	[13]	Bounded arboricity	Adversarial	$O(1)$	$\tilde{O}(n^{2/3})$
	here	Trees	Dynamic	$2+\varepsilon$	$O(\frac{\log^2 n}{\varepsilon^2})$
	here	Bounded arboricity	Dynamic	$O(1)$	$\tilde{O}(n^{4/5})$
	[13]	Forests	Adversarial	$\frac{3}{2}-\varepsilon$	$\Omega(\sqrt{n})$
	here	General	Adversarial	$1+O(\varepsilon)$	$\Omega\left(n^{1-\varepsilon}\right)$
MWM:	[10]	General	Adversarial	$4+\varepsilon$	$O(n\log^2 n)$
	here	General	Random	polylog(n)	polylog(n)
	here	Bounded arboricity	Dynamic	$O(1)$	$\tilde{O}(n^{4/5})$

Table 1. Results for estimating the size (weight) of a maximum (weighted) matching in data streams.

Techniques and Contribution Table 1 gives an overview of our results in comparison to previously known algorithms and lower bounds. Our first main result (Section 2) is an approximate estimation algorithm for the maximum weight of a matching. We give a generic procedure using any unweighted estimation as black box. In particular:

Theorem 1 (informal version). *Given a λ-approximate estimation using S space, there exists an $O(\lambda^4)$-approximate estimation algorithm for the weighted matching problem using $O(S \cdot \log n)$ space.*

The previous algorithms for weighted matchings in insertion only streams analyzed in [14,26,11,32] extend the greedy approach by a charging scheme. If edges are mutually exclusive, the new edge will be added if the weight of the matching increases by a given threshold, implicitly partitioning the edges into sets of geometrically increasing weights. We use a similar scheme, but with a twist: Single edge weights cannot be charged to an edge with larger weight as estimation routines do not necessarily give information on distinct edges. However, entire matchings can be charged as the contribution of a specific range of weights r can only be large if these edges take up a significant part of any maximum matching in the subgraph containing only the edges of weight at least r. For analysis, we use a result on parallel algorithms by Uehara and Chen [30].

We show that the weight outputted by our algorithm is close to the weight of the matching computed by the authors, implying an approximation to the maximum weight.

We can implement this algorithm in dynamic streams although at submission, we were unaware of any estimations for dynamic streams. Building on the work by Esfandiari et al. [13], we give a constant estimation on the matching size in bounded arboricity graphs. The main obstacle to adapt their algorithms for bounded arboricity graphs is that they maintain a small size matching using the greedy algorithm which is hard for dynamic streams. Instead of maintaining a matching, we use the Tutte matrix to get a 1-pass streaming algorithm using $\tilde{O}(n^{4/5})$ space, which immediately extends to weighted matching. Similar bounds have been obtained independently by Chitnis et al. [7].

Our lower bound (Section 3) is proven via reduction from the Boolean Hidden Hypermatching problem introduced by Verbin and Yu [31]. In this setting, two players Alice and Bob are given a binary n-bit string and a perfect t-hypermatching on n nodes, respectively. Bob also gets a binary string w. The players are promised that the parity of bits corresponding to the nodes of the i-th hypermatching either are equal to w_i for all i or equal to $1 - w_i$ for all i and the task is to find out which case holds using only a single round of communication. We construct a graph consisting of a t-clique for each hyperedge of Bob's matching and a single edge for each bit of Alice's input that has one node in common with the t-cliques. Then we show that approximating the matching size within a factor better than $1 + O(1/t)$ can also solve the Boolean Hidden Hypermatching instance. Using the lower bound of $\Omega(n^{1-1/t})$ from [31] we have

Theorem 2 (informal version). *Any* 1*-pass streaming algorithm approximating the size of the maximum matching matching up to an* $(1 + O(\varepsilon))$ *factor requires* $\Omega(n^{1-\varepsilon})$ *bits of space.*

This lower bound also implies an $\Omega(n^{1-\varepsilon})$ space bound for $1+O(\varepsilon)$ approximating the rank of a matrix in data streams which also improves the $\Omega(\sqrt{n})$ bound by Li, Nguyen, and Woodruff [23] for linear sketches.

1.1 Preliminaries

We use $\tilde{O}(f(n))$ to hide factors polylogarithmic in $f(n)$. Any randomized algorithm succeeding with high probability has at least $1 - 1/n$ chance of success. Graphs are denoted by $G(V, E, w)$ where V is the set of n nodes, E is the set of edges and $w : E \to \mathbb{R}^+$ is a weight function. We omit w for unweighted graphs. For a subset of nodes $S \subseteq V$, we denote by $E(S)$ the edges of the subgraph induced by S. Our estimated value $\widehat{M}$ is a λ-approximation to the size of the maximum matching M if $\widehat{M} \leq |M| \leq \lambda\widehat{M}$.

2 Weighted Matching

We start by describing the parallel algorithm by Uehara and Chen [30] which we call the *partitioning algorithm.* Let $\gamma > 1$ and $k > 0$ be constant. We partition the edge set by t ranks where all edges e in rank $i \in \{1, \ldots, t\}$ have a weight $w(e) \in \left(\gamma^{i-1} \cdot \frac{w_{max}}{kN}, \gamma^{i} \cdot \frac{w_{max}}{kN}\right]$ where w_{max} is the maximal weight in G. Let $G' = (V, E, w)$ be equal to G but each edge e in rank i has weight $r_i := \gamma^i$ for all $i = 1, \ldots, t$. Starting with $i = t$, we compute an unweighted maximal matching M_i considering only edges in rank i (in G') and remove all edges incident to a matched node. Continue with $i - 1$. The weight of the matching $M = \bigcup M_i$ is $w(M) = \sum_{i=1}^{t} r_i \cdot |M_i|$ and satisfies $w_G(M^*) \geq w_{G'}(M) \geq \frac{1}{2\gamma} \cdot w_G(M^*)$ where M^* is an optimal weighted matching in G. The previous algorithms [14,26,11,32,10] for insertion-only streams use a similar partitioning of edge weights. Since these algorithms are limited to storing one maximal matching (in case of [10] one maximal matching per rank), they cannot compute residual maximal matchings in each rank. However, by charging the smaller edge weights into the higher ones, the resulting approximation factor can be made reasonably close to that of Uehara and Chen. Since these algorithms maintain matchings, they cannot have sublinear space in an insertion-only stream and they need at least $\Omega(n^{2-3\varepsilon})$ in a dynamic stream even when the maintained matching is only a $O(n^{\varepsilon})$ approximation ([5]). Though the complexity for unweighted estimating unweighted matchings is not settled for any streaming model, there exist graph classes for which one can improve on these algorithms wrt space requirement. Therefore, we assume the existence of a black box λ-approximate matching estimation algorithm.

Algorithm and Analysis For the analysis, we use a result on parallel algorithms by Uehara and Chen [30]. We show that the weight outputted by our algorithm is close to the weight of the matching computed by their algorithm, implying an approximation to the maximum weight.

The partitioning of Uehara and Chen can be constructed almost obliviously: Let $(e_0, w(e_0))$ be the first inserted edge. Then an edge e belongs to rank i iff $2^{i-1} \cdot w(e_0) < w(e) \leq 2^i \cdot w(e_0)$ for some $i \in \mathbb{N}$. Note that we can assume that the weights are greater than 0. Then the number of sets is $O(\log \frac{w_{max}}{w_{min}})$. For the sake of simplicity, we assume that the edge weights are in $[1, W]$. Further details can be found in the full version of the paper.

We now introduce a bit of notation we will use in the algorithm and throughout the proof. We partition the edge set $E = \bigcup_{i=0}^{t} E_i$ by $t+1 = O(\log W)$ ranks where the set E_i contains all edges e with weight $w(e) \in \left[2^i, 2^{i+1}\right)$. Wlog we assume $E_t \neq \emptyset$ (otherwise let t be the largest rank with $E_t \neq \emptyset$). Let $G' = (V, E, w')$ be equal to G but each edge $e \in E_i$ has weight $w'(e) = r_i := 2^i$ for all $i = 0, \ldots, t$. Let $M = \bigcup_{i=0}^{t} M_i$ be the matching computed by the partitioning algorithm and S be a $(t+1)$-dimensional vector with $S_i = \sum_{j=i}^{t} |M_i|$.

Algorithm 1 now proceeds as follows: For every $i \in \{0, \ldots t\}$ the size of a maximum matching in $(V, \bigcup_{j=i}^{t} E_j)$ and S_i differ by only a constant factor. Conceptually, we set our estimator $\widehat{S_i}$ of S_i to be the approximation of the size of the maximum matching of $(V, \bigcup_{j=i}^{t} E_i)$ and the estimator of the contribution of the edges in E_i to the weight of an optimal weighted matching is $\widehat{R_i} = \widehat{S_i} - \widehat{S_{i+1}}$. The estimator $\widehat{R_i}$ is crude and generally not a good approximation to $|M_i|$. What helps us is that if the edges M_i have a significant contribution to $w(M)$, then

Algorithm 1 Weighted Matching Approximation

```
Require: Graph $G = (V, \bigcup_{i=0}^{t} E_i)$ with weights $r_i$ for edges in $E_i$
Ensure: Estimator of the weighted matching
  for $i = t$ to $0$ do
      $\widehat{S_i} = \widehat{R_i} = 0$
  $weight = 0$, $last = t$
  $\widehat{R_t} = \widehat{S_t} =$ Unweighted Matching Estimation$(V, E_t)$
  for $i = t-1$ to $0$ do
      $\widehat{S_i} =$ Unweighted Matching Estimation$(V, \bigcup_{j=i}^{t} E_j)$
      if $\widehat{S_i} > \widehat{S_{last}} \cdot T$ then          ▷ Add current index $i$ to $I_{good}$
          if $\widehat{S_i} - \widehat{S_{last}} \geq c \cdot \widehat{R}_{last}$ then     ▷ Add current index $i$ to $I_{sign}$
              $\widehat{R_i} = \widehat{S_i} - \widehat{S_{last}}$
              $last = i$
      else
          $\widehat{S_i} = 0$
  return $\frac{2}{5} \sum_{i=0}^{t} r_i \cdot \widehat{R_i}$
```

$|M_i| \gg \sum_{j=i+1}^{t} |M_j| = S_{i+1}$. In order to detect whether the matching M_i has a significant contribution to the objective value, we introduce two parameters T and c. The first matching M_t is always significant (and the simplest to approximate by setting $\widehat{R_t} = \widehat{S_t}$). For all subsequent matchings $i < t$, let j be the most recent matching which we deemed to be significant. We require $\widehat{S_i} \geq T \cdot \widehat{S_j}$ and $\widehat{R_i} \geq c \cdot \widehat{R_j}$. If both criteria are satisfied, we use the estimator $\widehat{R_i} = \widehat{S_i} - \widehat{S_j}$ and set i to be the now most recent, significant matching, otherwise we set $\widehat{R_i} = 0$. The final estimator of the weight is $\sum_{i=0}^{t} r_i \cdot \widehat{R_i}$. The next definition gives a more detailed description of the two sets of ranks which are important for the analysis.

Definition 1 (Good and Significant Ranks). *Let $\widehat{S}$ and $\widehat{R}$ be the vectors at the end of Algorithm 1. An index i is called to be a* good rank *if $\widehat{S_i} \neq 0$ and i is a* significant rank *if $\widehat{R_i} \neq 0$. We denote the set of good ranks by I_{good} and the set of significant ranks by I_{sign}, i. e., $I_{good} := \left\{ i \subseteq \{0, \ldots t\} \; | \widehat{S_i} \neq 0 \right\}$ and $I_{sign} := \left\{ i \subseteq \{0, \ldots t\} \; | \widehat{R_i} \neq 0 \right\}$. We define I_{good} and I_{sign} to be in descending order and we will refer to the ℓ-th element of I_{good} and I_{sign} by $I_{good}(\ell)$ and $I_{sign}(\ell)$, respectively. That means $I_{good}(1) > I_{good}(2) > \ldots > I_{good}(|I_{good}|)$ and $I_{sign}(1) > I_{sign}(2) > \ldots > I_{sign}(|I_{sign}|)$. We slightly abuse the notation and set $I_{sign}(|I_{sign}| + 1) = 0$. Let $D_1 := |M_t|$ and for $\ell \in \{2, \ldots, |I_{sign}|\}$ we define the sum of the matching sizes between two significant ranks $I_{sign}(\ell)$ and $I_{sign}(\ell - 1)$ where the smaller significant rank is included by $D_\ell := \sum_{i=I_{sign}(\ell)}^{I_{sign}(\ell-1)-1} |M_i|$.*

In the following, we subscript indices of significant ranks by s and of good ranks by g. We state some simple properties of I_{good} and I_{sign}.

Lemma 1. *Let I_{good} and I_{sign} be defined as in Definition 1. Then*

1. $I_{good}(1) = I_{sign}(1) = t$ *and* $I_{sign} \subseteq I_{good}$.
2. *For every good rank* $i_g \in I_{good}$ *there is an* $\ell \in \{0, \ldots, |I_{sign}|\}$ *such that* $I_{sign}(\ell) > i_g \geq I_{sign}(\ell+1)$ *and* $\widehat{S_{i_g}} > T \cdot \widehat{S_{I_{sign}(\ell)}}$.
3. $\widehat{S_{i_s}} > T \cdot \widehat{S_{i'_s}}$ *for every* $i_s, i'_s \in I_{sign}$ *with* $i_s < i'_s$.
4. $\widehat{R_{i'_s}} > c \cdot \widehat{R_{i_s}}$ *for any* $i_s \in I_{sign}$ *and* $i'_s \in I_{sign}$ *with* $i'_s < i_s$.
5. $\widehat{S_{i_g}} > T \cdot \widehat{S_{i_s}}$ *for any* $i_s \in I_{sign}$ *and* $i_g \in I_{good}$ *with* $i_g < i_s$.

Now, we have the necessary notations and properties of good and significant ranks to proof our main theorem.

Theorem 1. *Let $G = (V, E, w)$ be a weighted graph where the weights are from $[1, W]$. Let A be an algorithm that returns an λ-estimator $\widehat{M}$ for the size of a maximum matching M of a graph with $1/\lambda \cdot |M| \leq \widehat{M} \leq |M|$ with failure probability at most δ and needs space S. If we partition the edge set into sets $E_0, \ldots, E_t$ with $t = \lfloor \log W \rfloor$ where E_i consists of all edges with weight in $[2^i, 2^{i+1})$, set $r_i = 2^i$, and use A as the unweighted matching estimator in Algorithm 1, then there are parameters T and c depending on λ such that the algorithm returns an $O(\lambda^4)$-estimator $\widehat{W}$ for the weight of the maximum weighted matching with failure probability at most $\delta \cdot (t+1)$ using $O(S \cdot t)$ space, i.e. there is a constant c such that $\frac{1}{c\lambda^4} \cdot w(M^*) \leq \widehat{W} \leq w(M^*)$ where M^* is an optimal weighted matching.*

Proof (sketch). In the following we condition of the event that all calls to the unweighted estimation routine succeed, which happens with probability at least $1 - \delta \cdot (t+1)$. The estimator returned by Algorithm 1 can be written as $\sum_{\ell=1}^{|I_{sign}|} r_{I_{sign}(\ell)} \cdot \widehat{R_{I_{sign}(\ell)}}$. Using similar arguments as found in Lemma 4 of [30], we have $\frac{1}{8} \cdot w(M^*) \leq \sum_{i=0}^{t} r_i |M_i| \leq w(M^*)$. Thus, it is sufficient to show that $\sum_{\ell=1}^{|I_{sign}|} r_{I_{sgin}(\ell)} \cdot \widehat{R_{I_{sign}(\ell)}}$ is a good estimator for $\sum_{i=0}^{t} r_i |M_i|$. We first consider the problem of estimating D_ℓ, and then how to charge the matching sizes.

(1) Estimation of $\boldsymbol{D_\ell}$ Since $\bigcup_{j=i}^{t} M_j$ is a maximal matching in $\bigcup_{j=i}^{t} E_j$, $\widehat{S}_i$ is a good estimator for S_i:

Lemma 2. *For all $i \in \{0, \ldots, t\}$ we have* $\dfrac{1}{\lambda} \cdot S_i \leq \widehat{S}_i \leq 2 \cdot S_i$.

Next, we show that for an index $i_g \in I_{good}$ the difference $\widehat{S_{i_g}} - \widehat{S_{I_{sign}(\ell)}}$ to the last significant rank is a good estimator for $\sum_{i=i_g}^{I_{sign}(\ell)-1} |M_i|$.

Lemma 3. *For all $i_g \in I_{good}$ with $I_{sign}(\ell+1) \leq i_g < I_{sign}(\ell)$ for some $\ell \in \{1, \ldots, |I_{sign}|\}$ and $T = 8\lambda^2 - 2\lambda$,*

$$\frac{1}{2\lambda} \cdot \sum_{i=i_g}^{I_{sign}(\ell)-1} |M_i| < \widehat{S_{i_g}} - \widehat{S_{I_{sign}(\ell)}} < \frac{5}{2} \cdot \sum_{i=i_g}^{I_{sign}(\ell)-1} |M_i|$$

and $\frac{1}{\lambda}|M_t| \leq \widehat{S}_t \leq 2|M_t|$.

From Lemma 1 (1) we know that $I_{sign} \subseteq I_{good}$ which together with the last Lemma 3 implies that $\widehat{R_{I_{sign}(\ell)}}$ is a good estimator for D_ℓ.

Corollary 1. *For $\ell \in \{1, \ldots, |I_{sign}|\}$, $\frac{1}{2\lambda} \cdot D_\ell \leq \widehat{R_{J(\ell)}} \leq \frac{5}{2} \cdot D_\ell$. Furthermore, if $c > 5\lambda$ then the values of the D_ℓ are exponentially increasing:*

$$D_1 \leq \frac{5\lambda}{c} D_2 \leq \ldots \leq \left(\frac{5\lambda}{c}\right)^{|I_{sign}|-1} D_{|I_{sign}|-1}.$$

(2) The Charging Argument We show that sum of the matching sizes between two significant ranks $I_{sign}(\ell+1)$ and $I_{sign}(\ell)$ is bounded by $O(\lambda \cdot T \cdot D_\ell) = O\left(\lambda \cdot T \cdot \sum_{i=I_{sign}(\ell)}^{I_{sign}(\ell-1)+1} |M_i|\right)$.

Lemma 4. *Setting $c = \frac{2}{5} \cdot T + 5\lambda$ in Algorithm 1. Then for $\ell \in \{1, \ldots, |I_{sign}|-1\}$, $\sum_{i=I_{sign}(\ell+1)+1}^{I_{sign}(\ell)-1} |M_i| \leq (2\lambda \cdot T + 25\lambda^2) \cdot D_\ell$ and $\sum_{i=0}^{I_{sign}(|I_{sign}|)-1} |M_i| \leq (2\lambda \cdot T + 25\lambda^2) \cdot D_{|I_{sign}|}$ if $0 \notin I_{sign}$.*

Proof. For the proof of the first inequality, let $i_g \in I_{good}$ be minimal such that $I_{sign}(\ell+1) < i_g < I_{sign}(\ell)$ for $\ell \in \{1, \ldots, |I_{sign}|-1\}$. If such a good rank does not exist, set $i_g = -1$. We distinguish between two cases. Note that $c = \frac{5}{2} \cdot T + 5\lambda > 5\lambda$.

Case 1: $i_g = I_{sign}(\ell+1) + 1$. For the sake of simplicity, we abuse the notation and set $\widehat{S_{I_{sign}(0)}} = 0$ such that $\widehat{R_{I_{sign}(\ell)}} = \widehat{S_{I_{sign}(\ell)}} - \widehat{S_{I_{sign}(\ell-1)}}$ also holds for $\ell = 1$. Using Lemma 3 we have

$$\begin{aligned} \sum_{i=I_{sign}(\ell+1)+1}^{I_{sign}(\ell)-1} |M_i| &= \sum_{i=i_g}^{I_{sign}(\ell)-1} |M_i| \underset{\text{Lem. 3}}{\leq} 2\lambda \cdot \left(\widehat{S_{i_g}} - \widehat{S_{I_{sign}(\ell)}}\right) \\ &\underset{i_g \notin I_{sign}}{\leq} 2\lambda c \cdot \widehat{R_{I_{sign}(\ell)}} = 2\lambda \cdot c \cdot \left(\widehat{S_{I_{sign}(\ell)}} - \widehat{S_{I_{sign}(\ell-1)}}\right) \\ &\underset{\text{Lem. 3}}{\leq} 5\lambda \cdot c \cdot \sum_{i=I_{sign}(\ell)}^{I_{sign}(\ell-1)-1} |M_i| = 5 \cdot c \cdot D_\ell \qquad (1) \end{aligned}$$

Case 2: $i_g \neq I_{sign}(\ell+1)+1$**.** In this case $\widehat{S_{I_{sign}(\ell+1)+1}} \leq T \cdot \widehat{S_{I_{sign}(\ell)}}$. Thus

$$\begin{aligned}
\sum_{i=I_{sign}(\ell+1)+1}^{I_{sign}(\ell)-1} |M_i| &\leq S_{I_{sign}(\ell+1)+1} \underset{\text{Lem. 2}}{\leq} \lambda \cdot \widehat{S_{I_{sign}(\ell+1)+1}} \\
&\leq \lambda \cdot T \cdot \widehat{S_{I_{sign}(\ell)}} \underset{\text{Lem. 2}}{\leq} 2\lambda \cdot T \cdot S_{I_{sign}(\ell)} = 2\lambda \cdot T \cdot \sum_{i=1}^{\ell} D_i \\
&\underset{\text{Cor. 1}}{\leq} 2\lambda \cdot T \cdot D_\ell \cdot \sum_{i=1}^{\ell} \left(\frac{5\lambda}{c}\right)^i \leq 2\lambda \cdot T \cdot D_\ell \cdot \frac{1}{1-\frac{5\lambda}{c}} \qquad (2)
\end{aligned}$$

Combining the inequalities 1 and 2, we have $\sum_{i=I_{sign}(\ell+1)+1}^{I_{sign}(\ell)-1} |M_i| \leq \max\left\{5\lambda \cdot c, \frac{2\lambda \cdot T}{1-\frac{5\lambda}{c}}\right\} \cdot D_\ell$ which simplifies to

$$\sum_{i=I_{sign}(\ell+1)+1}^{I_{sign}(\ell)-1} |M_i| \leq (2\lambda \cdot T + 25\lambda^2) \cdot D_\ell \qquad \text{for } \ell \in \{1, \dots, |I_{sign}|-1\} \qquad (3)$$

because $c = \frac{5}{2} \cdot T + 5\lambda$. If $0 \notin I_{sign}$ we can do the same arguments to bound $\sum_{i=0}^{I_{sign}(|I_{sign}|)-1} |M_i|$ by $(2\lambda \cdot T + 25\lambda^2) \cdot D_{|I_{sign}|}$. □

We use Lemma 4 to show that $w(M)$ is bounded in terms of $\sum_{\ell=1}^{|I_{sign}|} r_{I_{sign}(\ell)} \cdot D_\ell$:

$$\sum_{i=0}^{t} r_i \cdot |M_i| \geq \sum_{\ell=1}^{|I_{sign}|} r_{I_{sign}(\ell)} \cdot D_\ell \qquad (4)$$

$$\sum_{i=0}^{t} r_i \cdot |M_i| \leq (1 + 2\lambda \cdot T + 25\lambda^2) \cdot \sum_{\ell=1}^{|I_{sign}|} r_{I_{sign}(\ell)} \cdot D_\ell. \qquad (5)$$

Putting Everything Together Using Lemma 1 we have $\frac{1}{2\lambda} \cdot D_\ell \leq \widehat{R_{I_{sign}(\ell)}} \leq \frac{5}{2} \cdot D_\ell$ for all $\ell \in \{1, \dots, |I_{sign}|\}$ which with (4) and (5) gives $\frac{1}{2\lambda \cdot (1+2\lambda \cdot T + 25\lambda^2)} \cdot w(M) \leq \sum_{\ell=1}^{|I_{sign}|} r_{I_{sign}(\ell)} \cdot \widehat{R_{I_{sign}(\ell)}} \leq \frac{5}{2} \cdot w(M)$. Recall that we set $T = 8\lambda^2 - 2\lambda$. Now, folding in the factor of $\frac{1}{8}$ from the partitioning and rescaling the estimator gives an $O(\lambda^4)$-estimation on the weight of an optimal weighted matching.

2.1 Applications

Since every edge insertion and deletion supplies the edge weight, it is straightforward to determine the rank for each edge upon every update. Using the following results for unweighted matching, we can obtain estimates with similar approximation guarantee and space bounds for weighted matching.

Random Order Streams For an arbitrary graph whose edges are streamed in random order, Kapralov, Khanna and Sudan [20] gave an algorithm with polylog n approximation guarantee using polylog n space with failure probability $\delta = 1/\text{polylog } n$. Since this probability takes the randomness of the input permutation into account, we cannot easily amplify it, though for $\log W \leq \delta$, the extension to weighted matching still succeeds with at least constant probability.

Adversarial Streams For graphs of bounded arboricity, Esfandiari et al. [13] gave an algorithm with constant approximation guarantee using $\tilde{O}(n^{2/3})$ space.

Dynamic Streams Matching in trees can be easily sketched by counting the number of distinct elements of the degree vector initialized to $-\mathbf{1}^n$. We briefly sketch how to extend the algorithm by Esfandiari et al. [13] to dynamic streams: The only part that is not straightforwardly adapted is the small matching up to some threshold k maintained by the greedy algorithm in insertion-only streams. For this, we summarize entries of the adjacency matrix as well as entries of the Tutte-matrix [29], where each non-zero entry (i, j) of the adjacency matrix is replaced by the variable x_{ij} if $i < j$ and $-x_{ij}$ if $i > j$. Lovász [25] showed that the maximum rank of T over all choices of the indeterminates is twice the size of the maximum matching. Furthermore, Lovász also noted that T has maximum rank with high probability if the indeterminates are chosen independently and uniformly at random from $\{1, \ldots, \text{poly}(n)\}$. This shows that computing the size of the maximum matching is a special case of computing the rank of a matrix; we simply maintain the Tutte-matrix with randomly chosen values for each entry. Uniformly random bits would require $O(n^2)$ space, which can be averted by using Nisan's pseudorandom generator for bounded space computation [28,18].

Given a positive integer k and a stream over updates to a matrix A, an algorithm for the *rank decision problem* outputs 1 if $\text{rank}(A) \geq k$ and 0 otherwise. Clarkson and Woodruff [8] proposed an algorithm operating in fully dynamic streams using $O(k^2 \log n)$ space. Setting $k = n^{2/5}$ then gives us the following.

Theorem 3. *Let G be a graph with bounded arboricity ν. Then there exists an algorithm estimating the size of the maximum matching in G within an $O(\nu)$-factor in the dynamic streaming model using a single pass over the data and $\tilde{O}(\nu \cdot n^{4/5})$ space or two passes over the data and $\tilde{O}(\nu \cdot n^{2/3})$ space.*

3 Lower Bound

Esfandiari et al. [13] showed a space lower bound of $\Omega(\sqrt{n})$ for any estimation better than 3/2. Their reduction (see below) uses the Boolean Hidden Matching Problem introduced by Bar-Yossef et al. [6], and further studied by Gavinsky et al. [15]. We will use the following generalization due to Verbin and Yu [31].

Definition 2 (Boolean Hidden Hypermatching Problem [31]). *In the* Boolean Hidden Hypermatching *Problem $BHH_{t,n}$ Alice gets a vector $x \in \{0,1\}^n$ with $n = 2kt$ and $k \in \mathbb{N}$ and Bob gets a perfect t-hypermatching M on the n coordinates of x, i. e., each edge has exactly t coordinates, and a string $w \in \{0,1\}^{n/t}$. We denote the vector of length n/t given by $(\bigoplus_{1\leq i\leq t} x_{M_{1,i}}, \ldots, \bigoplus_{1\leq i\leq t} x_{M_{n/t,i}})$ by Mx where $(M_{1,1}, \ldots, M_{1,t}), \ldots, (M_{n/t,1}, \ldots, M_{n/t,t})$ are the edges of M. The problem is to return 1 if $Mx \oplus w = 1^{n/t}$ and 0 if $Mx \oplus w = 0^{n/t}$, otherwise the algorithm may answer arbitrarily.*

Verbin and Yu [31] showed a lower bound of $\Omega(n^{1-1/t})$ for the randomized one-way communication complexity for $BHH_{t,n}$. For our reduction we require $w = 0^{n/t}$ and thus $Mx = 1^{n/t}$ or $Mx = 0^{n/t}$. We denote this problem by $BHH^0_{t,n}$. We can show that this does not reduce the communication complexity.

Lemma 5. *The communication complexity of $BHH^0_{t,4n}$ is lower bounded by the communication complexity of $BHH_{t,n}$.*

Let us now sketch the reduction from $BHH^0_{2,n}$ to approximate maximum matching to get the idea how to extend it to the general bound. Let x, M be the input for Alice and Bob. They construct a graph consisting of $2n$ nodes denoted by $v_{1,i}$ and $v_{2,i}$, for $i \in \{1, \ldots, n\}$. For each bit x_i of $x \in \{0,1\}^n$, Alice adds an edge $\{v_{1,i}, v_{2,i}\}$ iff $x_i = 1$ and sends a message to Bob. Bob adds an edge between $v_{2,i}$ and $v_{2,j}$ for each edge $\{x_i, x_j\} \in M$ and approximates the size of the matching. If all parities are 1 then the size of the maximum matching is $n/2$. If the parities are all 0 then the size is $3n/4$. Every streaming algorithm that approximates better than $3/2$ can distinguish between these two cases. The first observation is that the size of the matching is lower bounded by the number of ones in x. The second observation is that the added edges by Bob increase the matching iff the parities of all pairs are 0 and only the edges between the two 0 input bits of Alice increase the matching. Since it is promised that all parities are equal and the number of ones is exactly $n/2$ we can calculate the number of $(0,0)$ pairs. For our lower bound we show that this calculation is still possible if Bob adds a t-clique between the corresponding nodes of the hyperedge.

Theorem 2. *Any randomized streaming algorithm that approximates the maximum matching size within a $1 + \frac{1}{3t/2-1}$ factor for $t \geq 2$ needs $\Omega(n^{1-1/t})$ space.*

Finally, constructing the Tutte-matrix with randomly chosen entries gives us

Corollary 2. *Any randomized streaming algorithm that approximates* $\operatorname{rank}(A)$ *of $A \in \mathbb{R}^{n \times n}$ within a $1 + \frac{1}{3t/2-1}$ factor for $t \geq 2$ requires $\Omega(n^{1-1/t})$ space.*

References

1. K. Ahn and S. Guha. Graph sparsification in the semi-streaming model. In *ICALP (2)*, pages 328–338, 2009.
2. K. Ahn, S. Guha, and A. McGregor. Analyzing graph structure via linear measurements. In *SODA*, pages 459–467, 2012.
3. K. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *PODS*, pages 5–14, 2012.
4. K. Ahn, S. Guha, and A. McGregor. Spectral sparsification in dynamic graph streams. In *APPROX-RANDOM*, pages 1–10, 2013.
5. S. Assadi, S. Khanna, Y. Li, and G. Yaroslavtsev. Tight bounds for linear sketches of approximate matchings. *CoRR*, abs/1505.01467, 2015.
6. Z. Bar-Yossef, T. S. Jayram, and I. Kerenidis. Exponential separation of quantum and classical one-way communication complexity. *SIAM J. Comput.*, 38(1):366–384, 2008.
7. R. Chitnis, G. Cormode, H. Esfandiari, M. Hajiaghayi, A. McGregor, M. Monemizadeh, and S. Vorotnikova. Kernelization via sampling with applications to dynamic graph streams. *CoRR*, abs/1505.01731, 2015.
8. K. Clarkson and D. Woodruff. Numerical linear algebra in the streaming model. In *STOC*, pages 205–214, 2009.

9. M. Crouch, A. McGregor, and D. Stubbs. Dynamic graphs in the sliding-window model. In *ESA*, pages 337–348, 2013.
10. M. Crouch and D. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *APPROX/RANDOM*, pages 96–104, 2014.
11. L. Epstein, A. Levin, J. Mestre, and D. Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM J. Discrete Math.*, 25(3):1251–1265, 2011.
12. L. Epstein, A. Levin, D. Segev, and O. Weimann. Improved bounds for online preemptive matching. In *STACS*, pages 389–399, 2013.
13. H. Esfandiari, M. Hajiaghayi, V. Liaghat, M. Monemizadeh, and K. Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. In *SODA*, pages 1217–1233, 2015.
14. J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2-3):207–216, 2005.
15. D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM J. Comput.*, 38(5):1695–1708, 2008.
16. A. Goel, M. Kapralov, and S. Khanna. On the communication and streaming complexity of maximum bipartite matching. In *SODA*, pages 468–485, 2012.
17. M. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams, 1998.
18. P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *FOCS*, pages 189–197, 2000.
19. M. Kapralov. Better bounds for matchings in the streaming model. In *SODA*, pages 1679–1697, 2013.
20. M. Kapralov, S. Khanna, and M. Sudan. Approximating matching size from random streams. In *SODA*, pages 734–751, 2014.
21. C. Konrad. Maximum matching in turnstile streams. *CoRR*, abs/1505.01460, 2015.
22. C. Konrad, F. Magniez, and C. Mathieu. Maximum matching in semi-streaming with few passes. In *APPROX-RANDOM*, pages 231–242, 2012.
23. Y. Li, H. Nguyen, and D. Woodruff. On sketching matrix norms and the top singular vector. In *SODA*, pages 1562–1581, 2014.
24. Y. Li, H. Nguyen, and D. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *STOC*, pages 174–183, 2014.
25. L. Lovász. On determinants, matchings, and random algorithms. In *FCT*, pages 565–574, 1979.
26. A. McGregor. Finding graph matchings in data streams. In *APPROX-RANDOM*, pages 170–181, 2005.
27. A. McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014.
28. N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.
29. W. Tutte. The factorization of linear graphs. *J. London Math. Soc.*, 22:107–111, 1947.
30. R. Uehara and Z. Chen. Parallel approximation algorithms for maximum weighted matching in general graphs. *Inf. Process. Lett.*, 76(1-2):13–17, 2000.
31. E. Verbin and W. Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In *SODA*, pages 11–25. SIAM, 2011.
32. M. Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, 2012.