CS 590 Sublinear Algorithms Mar. 16, 2021

Lecture : Matching Algorithms in the Streaming Model

Lecturer: Elena Grigorescu *Scribe: Rohan Garg*

Today we will discuss matching algorithms in the streaming model. We will only consider the insertion model where edges only appear and are not deleted. Recall, that in the *semi-streaming model* edges arrive in a stream (in adversarial order) and the storage space of our algorithm is bounded by $O(n \cdot polylog(n))$. We are generally interested in single-pass algorithms but there is also interest in algorithms that take a constant number or polylogarithmic number of passes over the stream.

For matching algorithms in the streaming model, our goal is to have a max-size matching or max-weight matching for unweighted or weighted graphs respectively. Often, we will try and look at offline algorithms and extend them to the streaming or online models. For our first algorithm, we will show that the classical greedy 2-approximation algorithm can be easily extended to the streaming model setting. This algorithm is briefly introduced in the seminal paper by Karp et. al. [6].

There are two variants to the max-matching problem:

1. Maximum Cardinality Matching (MCM)
2. Maximum-Weighted Matching (MVM)

Recall: A *maximal* matching is a matching that cannot be extended by an extra edge. Note, that this is not the same as a maximum matching. Indeed, a maximum matching is always a maximal matching, but not vice-versa.

A quick implication of this is that our algorithms should always output a maximal-matching. Otherwise, we can extend the matching easily by an edge. Now, we present the first algorithm.

1 Maximum Cardinality Matching

Theorem 1 *There exists a 2-approximation algorithm for MCM in the semi-streaming model using $O(n \log n)$ bits of space.*

Proof Consider the following algorithm: Set $M \leftarrow \emptyset$. On edge $e = (u, v)$, if neither u nor v is matched in M, then add e to M. Output M at the end of the stream. ■

Let us now analyze the complexity of the algorithm as well as its approximation guarantee.

Theorem 2 *The returned matching is a maximal matching.*

Proof By construction. ■

Theorem 3 *The algorithm uses $O(n \log n)$ space.*

Proof The maximum matching M cannot have size more than $n/2$ and so we never use more space than $O(n \log n)$. ■

Theorem 4 *The returned matching is a 2-approximation.*

Proof Let M^* be the optimal matching for G. We want to show that $|M| \geq \frac{1}{2}|M^*|$. The crucial observation here is that any edge $e = (u, v) \in M$ that we take, "blocks" at most two edges of M^*. In other words, we decided to match u and v and maybe it was better to match u to some other vertex u' and vertex v to some other vertex v'. M' got two edges in the matching, and our matching, M, blocked those two edges in favor of edge e. In the worst case, this applies to all edges in the matching M. So, this gives us the desired bound of $|M| \geq \frac{1}{2}|M^*|$. ■

Now we will prove the same theorem using an alternative method. This will be useful for us for future proofs. We will use the following mapping (function) $\sigma : M^* \to M$.

1. For any edge $e \in M^* \cap M$, $\sigma(e) = e \in M$
2. For any edge $e \in M^* \backslash M$, there should be an edge f incident on one of the vertices of e as otherwise the algorithm would have added e to M as well; in this case, let $\sigma(e) = f$ (breaking the ties arbitrarily).

For both cases above, we say the edge $e \in M^*$ is *charged* to the edge $\sigma(e) \in M$. Notice that any edge in M can be charged at most twice as it is incident on at most two edges of M^* and we only charge an edge to an incident edge. Alternatively, for any edge $e \in M$

$$|\sigma^{-1}(e)| \leq 2$$

$$\implies \frac{|M^*|}{2} \leq |M|$$

2 Maximum Weighted Matching

Let's consider the Greedy algorithm for weighted matching. Consider the line graph on three vertices where the middle edge has weight 1 and the outer edges have weight 100. The greedy algorithm may return the matching of weight 1 (middle edge) where the optimal matching has weight 200 (outer edges). So, the greedy algorithm can be arbitrarily worse than the optimal.

Crouch and Stubbs give a $(4 + \epsilon)$ approximation. Feigenbaum et. al give a $(6 + \epsilon)$ approximation [5]. The current best is a $(2 + \epsilon)$ approximation by Paz and Schwartzmann [7]. Now, we present the result by Crouch and Stubs [4]. This will be via a reduction argument.

Theorem 5 *If an $\alpha-$approximation algorithm in the semi-streaming model for MCM exists, then there exists a $(2+\epsilon)\alpha$-approximation algorithm for the MVM problem in the semi-streaming model with only a $\log n$ blow up in space.*

Proof We will use a *Turing reduction.* Consider the following algorithm:

Let $E_i = \{e \in E \mid w(e) \geq (1+\epsilon)^i\}$. Let

$$W = \max_e w(e)$$

Let $K = \lfloor log_{i+\epsilon} W \rfloor$. Let $G_i = (V, E_i)$. Note: $E_1 \subseteq E_2 \subseteq \cdots \subseteq E_k$.

In parallel, for each weight threshold $(1+\epsilon)^i$, run the α-approximate MCM on E_i to compute C_i of G_i. Notice in this step we are running a *max-cardinality* matching algorithm that ignores the weights of the edges. Then, $M \leftarrow \emptyset$. For $i =$ *k to 1*, if $e \in C_i$ has none of its endpoints matched in M, then add e to M. Otherwise, discard e. Lastly, output M. ■

This algorithm is a $(4+\epsilon)$ approximation for MWM if we couple this with the greedy MCM algorithm presented earlier in the notes. Let's try and analyze this algorithm.

Lemma 6 *If M^* is an optimal weighted matching of G and M is the output matching, then*

$$w(M^*) \leq 2(1+\epsilon) \cdot \alpha \cdot w(M)$$

First, we will need the following definitions. Let, $M_i = M \cap E_i$ and $M_i^* = M^* \cap E_i$.

Question: Can we compare the *size* of M_i with size of M_i^* (not weights)?

Lemma 7

$$|M_i| \geq \frac{1}{2\alpha}|M_i^*| \quad \forall 1 \leq i \leq k$$

Proof We want to compare the following three values:

$$|C_i| \text{ vs. } |M_i| \text{ vs. } |M_i^*|$$

C_i is an α-approximation greedy matching for threshold $(1+\epsilon)^i$. So,

$$|C_i| \cdot \alpha \geq OPT_{MCM}(G_i)$$

.

This is by definition of C_i. Further, $OPT_{MCM}(G_i) \geq OPT_{MVM}(G_i)$ with the assumption that all edge-weights are larger than 1. So,

$$|C_i| \geq \frac{1}{\alpha}|M_i^*|$$

So to compare C_i and M_i, notice that M_i is made by taken edges greedily that are larger than threshold i in decreasing order. Using the reasoning from the proof for the greedy 2-approximation algorithm, one edge in M_i blocks at most two edges in C_i. So,

$$|M_i| \geq \frac{1}{2}|C_i|$$

.

Chaining together these inequalities, we get:

$$|M_i| \geq \frac{1}{2\alpha}|M_i^*|$$

■

For the final lemma, we have:

Lemma 8 *There exists a mapping $\sigma : M^* \to M$ s.t.*

1. $\forall e \in M$, $|\sigma^{-1}(e)| \leq 2\alpha$
2. $\forall e^* \in M^*$, $w(e^*) \leq (1+\epsilon)w(\sigma(e^*))$

Proof

Starting at the maximum value of i, we charge edges in M_i^* to edges in M_i. For any edge $e \in M^*$, we let $\sigma(e)$ be an edge in M that has fewer than 2α edges already charged to it. Lemma 7 ensures that such an edge always exists. This satisfies property 1. Further, since we process M^* in the decreasing order of i, edges of M_1^* are charged to edges of M_1, edges of $M_2^* \backslash M_1^*$ are charged to edges of M_2, and generally edges of $M_i^* \backslash M_{i-1}^*$ are charged to edges of M_i. This implies that for any edge e^*, we have

$$w(e^*) \leq (1+\epsilon)w(\sigma(e^*))$$

This completes the proof for lemma 7.

■

Using Lemma 7, we can conclude the following theorem from [4].

Theorem 9 *There is a semi-streaming $(4+\epsilon)$-approximation algorithm for maximum weight matching.*

Proof By Lemma 7, we have:

$$w(M^*) = \sum_{e \in M^*} w(e) \leq \sum_{e \in M^*} w(\sigma(e))(1+\epsilon) \leq 2\alpha(1+\epsilon) \cdot \sum_{e \in M} w(e) = 2\alpha(1+\epsilon)w(M)$$

■

3 Recent Work

For recent works in this line of research, see papers [1], [2], and [3] listed in the References below. These papers consider a different model of streaming where edges appear in the stream in random order. i.e. after seeing t of the total m edges in the stream, the subgraph containing these t edges looks as if the t out of m edges were sampled uniformly at random. There are many positive results for random order streams.

References

[1] Sepehr Assadi and Soheil Behnezhad. Beating two-thirds for random-order streaming matching. *CoRR*, abs/2102.07011, 2021. URL: https://arxiv.org/abs/2102.07011, http://arxiv.org/abs/2102.07011 arXiv:2102.07011.

[2] Aaron Bernstein. Improved bounds for matching in random-order streams. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 12:1–12:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. https://doi.org/10.4230/LIPIcs.ICALP.2020.12 doi:10.4230/LIPIcs.ICALP.2020.12.

[3] Aaron Bernstein and Aditi Dudeja. Online matching with recourse: Random edge arrivals. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14-18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference)*, volume 182 of *LIPIcs*, pages 11:1–11:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. https://doi.org/10.4230/LIPIcs.FSTTCS.2020.11 doi:10.4230/LIPIcs.FSTTCS.2020.11.

[4] Michael Crouch and Daniel M. Stubbs. Improved Streaming Algorithms for Weighted Matching, via Unweighted Matching. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*, volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 96–104, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2014/4690, https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.96 doi:10.4230/LIPIcs.APPROX-RANDOM.2014.96.

[5] Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2):207–216, December 2005. https://doi.org/10.1016/j.tcs.2005.09.013 doi:10.1016/j.tcs.2005.09.013.

[6] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 352–358, New York, NY, USA, 1990. Association for Computing Machinery. https://doi.org/10.1145/100216.100262 doi:10.1145/100216.100262.

[7] Ami Paz and Gregory Schwartzman. A (2+)-approximation for maximum weight matching in the semi-streaming model. *ACM Trans. Algorithms*, 15(2), December 2018. https://doi.org/10.1145/3274668 doi:10.1145/3274668.