CS 7880: Algorithms for Big Data (Fall'22) **Northeastern University**

Lecture 21

November 29, 2022

Instructor: Soheil Behnezhad *Scribe: Amir Azarmehr*

Disclaimer: *These notes have not been edited by the instructor.*

1 A Random-Order Streaming Algorithm for Maximum Matching

In this lecture we will cover an algorithm that almost-$\frac{2}{3}$-approximates the maximum matching in the random-order semistreaming model. This means that the edges of the graph appear in a random order, as opposed to an adversarial order, and the algorithm will use $\mathcal{O}(n \operatorname{polylog} n)$ space to approximate the maximum matching. The algorithm we will discuss is deterministic and uses $\mathcal{O}\left(n \log n \operatorname{poly} \frac{1}{\varepsilon}\right)$ space, to $\left(\frac{2}{3} - \varepsilon\right)$-approximate the maximum matching with high probability. More formally, we will prove the following:

Theorem 1. *When the edges of a graph G arrive in a random-order stream, there is an algorithm that uses $\mathcal{O}(n \log n \operatorname{poly} \frac{1}{\varepsilon})$ space with high probability, and returns a matching of size at least $\left(\frac{2}{3} - 3\varepsilon\right)\mu(G)$.*

First, we will reiterate the definition of edge-degree constrained subgraph (EDCS) and a related theorem from the last lecture. Then we will prove a key lemma about edge-degree bounded subgraphs and underfull edges, and move on to describing the algorithm. The results in this lecture are due to Bernstein [Ber20].

> **Remark.** At the time that Theorem 1 was proved by Bernstein [Ber20], 2/3-approximation was considered a barrier for this problem. This barrier was (only slightly) broken in a subsequent paper of Assadi and Behnezhad [AB21] which obtains a $(2/3 + \varepsilon_0)$-approximation for a small absolute constant $\varepsilon_0 > 0$ using $\widetilde{O}(n)$ space and a single pass (also under random-arrival of edges).

1.1 Preliminaries

Definition 2. Given a graph G, a subgraph $H \subseteq G$ is an EDCS(G, β, λ), if it has the following two properties:

(P1) $\forall (u, v) \in E_H$, we have $d_H(u) + d_H(v) \leq \beta$,

(P2) and $\forall (u, v) \in E_G \setminus E_H$, we have $d_H(u) + d_H(v) \geq (1 - \lambda)\beta$.

> **Remark.** A maximal matching is an EDCS with parameters $\beta = 2$ and $\lambda = \frac{1}{2}$.

What follows is an important property of EDCS, i.e. that it approximates maximum matching. For the proof, refer to the previous lecture.

Theorem 3. *For any (possibly non-bipartite) graph G, and $\varepsilon < \frac{1}{2}$, let $\lambda < \frac{\varepsilon}{64}$, and $\beta \geq 8\lambda^{-2} \log \frac{1}{\lambda}$. Then for any subgraph $H \subseteq G$ that is an EDCS(G, β, λ), it holds that $\mu(H) \geq \left(\frac{2}{3} - \varepsilon\right)\mu(G)$.*

We will now use this theorem to prove a lemma that is at the heart of the algorithm.

Definition 4. A graph H has bounded edge-degree β, if for all edges $(u,v) \in E_H$ it holds that $d_H(u) + d_H(v) \leq \beta$.

Definition 5. Given a graph G, and a subgraph $H \subseteq G$ that has bounded edge-degree β, an edge $(u,v) \in E_G \setminus E_H$ is (H, β, λ)-underfull if $d_H(u) + d_H(v) < (1-\lambda)\beta$.

Remark. An EDCS is a bounded edge-degree subgraph such that the rest of the graph has no underfull edges.

Lemma 6. *For any graph G, and $\varepsilon < \frac{1}{2}$, let $\lambda < \frac{\varepsilon}{128}$, and $\beta \geq 16\lambda^{-2}\log\frac{1}{\lambda}$. Let $H \subseteq G$ be of bounded edge-degree β, and let X be all the (H, β, λ)-underfull edges. Then $\mu(H \cup X) \geq \left(\frac{2}{3} - \varepsilon\right)\mu(G)$.*

Proof. Let M be a maximum matching in G, and let $X_M = X \cap M$, i.e. the underfull edges of the matching. Note that $H \cup M$ is a subgraph of G, and it also includes the maximum matching M. So, it holds that $\mu(H \cup M) = \mu(G)$.

Claim 7. *$H \cup X_M$ is an* EDCS$(H \cup M, \beta + 2, 2\lambda)$.

Taking the claim to be true, we can prove the lemma as follows:

$$\mu(H \cup X) \geq \mu(H \cup X_M) \geq \left(\frac{2}{3} - \varepsilon\right)\mu(H \cup M) = \left(\frac{2}{3} - \varepsilon\right)\mu(G)$$

Where the first inequality holds because $H \cup X \supseteq H \cup X_M$. And the second inequality follows from Claim 7 together with Theorem 3. ☐

Proof of Claim 7. Let $\tilde{H} = H \cup X_M$. We will simply check the two properties of EDCS for every edge. To see that (P1) holds, note that by adding the edges of X_M to H, the degree of any edge increases by at most 2, because X_M is a matching. For any edge $e \in E_H$, because of the (P1) property in H, we have:

$$d_{\tilde{H}}(e) \leq d_H(e) + 2 \leq \beta + 2,$$

and for any edge $e \in X_M$, because it was (H, β, λ)-underfull, we have:

$$d_{\tilde{H}}(e) \leq d_H(e) + 2 \leq (1-\lambda)\beta + 2 \leq \beta + 2$$

To see that (P2) holds, note that any edge in $e \in (H \cup M) \setminus \tilde{H}$ is not in X, i.e. it is not an (H, β, λ)-underfull edge. So we have:

$$d_{\tilde{H}}(e) \geq d_H(e) \geq (1-\lambda)\beta \geq (1-2\lambda)(\beta+2)$$ ☐

1.2 The Algorithm

Definition 8. Let $e_1, \ldots, e_m$ be the edges of the stream. We use $G_{>i}$ to denote the subgraph of G consisting of the edges $\{e_{i+1}, \ldots, e_m\}$, G_{late} to denote $G_{>\varepsilon m}$, and G_{early} to denote $G \setminus G_{\text{late}}$.

The algorithm is going to approximate $\mu(G_{\text{late}})$. This is justified because we expect $\mu(G_{\text{late}})$ to be about $(1-\varepsilon)\mu(G)$ when $\mu(G)$ is large. The following statements will formalize this fact.

Claim 9. *Without loss of generality, we can assume $\mu(G) \geq 20 \cdot \log n \cdot \varepsilon^{-2}$.*

Proof. For any graph G, we have $m \leq 2n\mu(G)$. To see this, fix a maximum matching in G. For every edge, charge 1 to an adjacent matching edge. The total charge is m and every matching edge is charged at most $2n$ times (n times from each endpoint). Hence, $m \leq 2n\mu(G)$. Using this fact, we can run a simple algorithm that stores the whole graph and reports the maximum matching when $m \leq 20 \cdot n\log n \cdot \varepsilon^{-2}$. And assume $\mu(G) \geq 20 \cdot \log n \cdot \varepsilon^{-2}$ otherwise. ☐

Lemma 10. *Assuming $\mu(G) \geq 20 \cdot \log n \cdot \varepsilon^{-2}$, we have $\mu(G_{\text{late}}) \geq (1-2\varepsilon)\mu(G)$ with high probability.*

Proof. Fix a maximum matching M. Let X_i be the indicator variable of the i-th matching edge appearing in G_{early}. We have

$$\mathbb{E}\left[X_i\right] = \varepsilon,$$

and

$$\mathbb{E}\left[\sum_{i=1}^{\mu(G)} X_i\right] = \varepsilon\mu(G).$$

As $X_1, \ldots, X_{\mu(G)}$ are negatively associated we can use the Chernoff bound (for an extensive treatment of negative association see [Waj17]):

$$\Pr\left(\sum_{i=1}^{\mu(G)} X_i > 2\varepsilon\mu\right) \leq \exp\left(-\frac{1}{3}\mu(G)\right) \leq n^{-5} \qquad \square$$

Now to approximate $\mu(G_{\text{late}})$, the algorithm is going to operate in two phases. The first phase is going to stop at some point i before εm and return an edge-degree bounded β subgraph $H \subseteq G_{\text{early}}$. The second phase is going to store all the (H, β, λ)-underfull edges in $G_{>i}$, we call them X. The algorithm will at the end return the maximum matching in $H \cup X$. The subgraph H must be chosen in such a manner that X is small enough so that the algorithm can store it efficiently. We will describe how the first phase operates after the following theorem.

Lemma 11. *The two-phase algorithm described above achieves a $\left(\frac{2}{3} - 3\varepsilon\right)$-approximation.*

Proof. To see why this holds we simply apply Lemma 1.1 to $H \cup X$ and $H \cup G_{>i}$.

$$\mu(H \cup X) \geq \left(\frac{2}{3} - \varepsilon\right)\mu(H \cup G_{>i}) \geq \left(\frac{2}{3} - \varepsilon\right)\mu(G_{\text{late}}) \geq \left(\frac{2}{3} - \varepsilon\right)(1-\varepsilon)\mu(G) \geq \left(\frac{2}{3} - 3\varepsilon\right)\mu(G) \qquad \square$$

We will now describe the first phase in detail. The first phase processes the stream in sections of length $\alpha = \frac{\varepsilon m}{n\beta^2+1}$. At any point, it maintains a bounded edge-degree B subgraph H. When an edge arrives, it is added to H if it is (H, β, λ)-underfull (where β and λ are set as in Lemma 1.1). If it is added then any edges of degree larger than β are removed until no such edge remains, so that H remains of bounded edge-degree β. The first phase terminates when H remains unchanged for a whole section. Intuitively, each section is a random sample of the remaining part of the stream. When no underfull edges appear in this sample, we expect there are not many underfull edges left.

What follows is a complete description of the algorithm.

Phase 1:

- Start with $H = \emptyset$
- Repeat until stopped:
- Process a section of α edges one by one.
- For edge (u, v) being processed, add (u, v) to H if $d_H(u) + d_H(v) < (1-\lambda)\beta$.
- If the edge is added, check for any edges (u', v') such that $d_H(u') + d_H(v') > \beta$ and remove them.
- If no edges were added from the last section, terminate Phase 1.

Phase 2:

- Start with $X = \emptyset$

- Process the remaining edges one by one.
- For edge e being processed, add e to X if it is (H, β, λ)-underfull.
- In the end, return the maximum matching $\mu(H \cup X)$.

Now, as promised, it remains to show two things. First, that Phase 1 will terminate after processing at most εm edges. Second, that X will have "few edges".

Lemma 12. *The first phase will process at most $n\beta^2 + 1$ sections, and hence at most $(n\beta^2 + 1)\alpha = \varepsilon m$ edges.*

Proof. Consider the following potential function:

$$\Phi(H) = \left(\beta - \frac{1}{2}\right) \sum_u d_H(u) - \sum_{(u,v) \in E_H} d_H(u) + d_H(v)$$

It starts at zero value when H is empty. As proven in the previous lecture, it is upper-bounded by $n\beta^2$. And it increases by at least 1 every time an underfull edge is added to H, or an edge with degree larger than β is removed from H. Therefore, there will be at most $n\beta^2$ such changes made to H. As every section, except the last one, makes at least one change to H, there will be at most $n\beta^2 + 1$ sections. $\square$

Lemma 13. *There will be at most $\gamma = 5\log(n)\frac{m}{\alpha}$ underfull edges encountered in the second phase, i.e. $|X| < \gamma$, with high probability.*

Proof. Note that this lemma is where we use the fact that the stream is in random order. Let E_k be the event that the algorithm finishes after the k-th section, and there are more than γ underfull edges left. We will bound the probability of each E_k and then use the union bound to prove the lemma. Let $m' = m - (k-1)\alpha$ be the number of remaining edges at the beginning of section k. Let U be the number of underfull edges still unprocessed at the beginning of section k. Conditioning on U, the probability of all the edges in section k not being underfull is exactly:

$$\left(1 - \frac{U}{m'}\right)\left(1 - \frac{U}{m'-1}\right) \cdots \left(1 - \frac{U}{m' - \alpha + 1}\right)$$

Each term in the product above is less than $\left(1 - \frac{\gamma}{m}\right)$, so for any k we have

$$\Pr(E_k) \leq \left(1 - \frac{\gamma}{m}\right)^{\alpha} \leq \left(1 - \frac{5\log n}{\alpha}\right)^{\alpha} \leq \exp(-5\log n) \leq n^{-5}$$

As the number of sections is smaller than n^2, using the union bound we can conclude that $|X| > \gamma$ with probability at most $n^2 \cdot n^{-5} = n^{-3}$ $\square$

Considering that at any step in the first phase, H has at most $\mathcal{O}(n\beta)$ edges, and there are at most γ edges in X, the algorithm will use space $\mathcal{O}(n\beta + \gamma) = \mathcal{O}(n \log n \operatorname{poly} \frac{1}{\varepsilon})$ with high probability.

Putting together these remarks along with Lemma 11, we can conclude Theorem 1.

References

[Waj17] David Wajc. "Negative association: definition, properties, and applications". In: *Manuscript, available from https://goo.gl/j2ekqM* (2017).

[Ber20] Aaron Bernstein. "Improved bounds for matching in random-order streams". In: *47th International Colloquium on Automata, Languages, and Programming*. Vol. 168. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020, Art. No. 12, 13.

[AB21] Sepehr Assadi and Soheil Behnezhad. "Beating Two-Thirds For Random-Order Streaming Matching". In: *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*. Ed. by Nikhil Bansal, Emanuela Merelli, and James Worrell. Vol. 198. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 19:1–19:13. DOI: 10.4230/LIPIcs.ICALP.2021.19. URL: https://doi.org/10.4230/LIPIcs.ICALP.2021.19.