Lecture 17: Dynamic matching approximation in polylog update time

Lecturer: *Anil, Pachara and Elena* Scribe: *Adithya Bhaskar*

1 Introduction

In the last lecture, we saw the following algorithm to approximate the size of the maximal matching in a graph $G \equiv (V, E)$:

- Let M be a maximal matching in G.
- Select a matching $M' \subseteq M$ where each edge is selected with probability $p = \sqrt{2} - 1$. Denote $V' \equiv V(M')$ the vertices in M' and $U \equiv V \setminus V(M)$ the vertices not matched by the original matching M.
- Let $H := G[V', U]$ be the bipartite subgraph induced between the vertices in V' and U. We do not construct H explicitly.
- Let $g := \mathbb{E}_\pi[|\mathrm{GMM}(H, \pi)| \quad |M']$ be the expected size of the greedy maximal matching over all possible permutations π of processing the edges.
- Output $\hat{\mu} := |M| + \max(0, g - |M'|)$.

This algorithm is good for us since it is cheap to maintain M as G changes.

We also saw last time that $\mathbb{E}[\hat{\mu}] \geq (2 - \sqrt{2})\mu(G)$. We will round out that conversation by now providing a bound from the other side.

Lemma 1. *With probability* 1*,*

$$\hat{\mu} \leq \mu(G)$$

where $\mu(G)$ *refers to the size of a maximum matching in* G *(note the use of maximum instead of maximal).*

Proof. Let S be a maximum matching in H, and denote by M'' the subset of edges in M' both of whose endpoints are matched by S. Note that one can replace these edges with the two that matched their endpoints, bringing the size of M up to $|M| + |M''|$. Therefore

$$\mu(G) \geq |M| + |M''| \tag{1}$$

Now, every edge in S is adjacent to exactly one vertex of M'. Thus, S leaves at most $|V(M')| - |S| = 2|M'| - |S|$ vertices unmatched. Hence,

$$|M''| \geq |M'| - (2|M'| - |S|) = |S| - |M'|$$

$$\Rightarrow \mu(G) \geq |M| + |S| - |M'|$$

Since S is a maximum matching in H, we also have $|S| \geq g$. This yields $\mu(G) \geq |M| + g - |M'|$. Of course, we also have $\mu(G) \geq |M|$. The result follows.

Recall from Monday's lecture that

$$\mathbb{E}_{M'}[\hat{\mu}] \geq \left(1 - p - \frac{2p}{1+p}\right)|M| + \frac{2p}{1+p}\mu(G) - \frac{2p}{1+p_1}L_1$$

where $L_1 = 0$ since there are no length-1 augmenting paths in a maximal matching. The choice $p = \sqrt{2} - 1$ gives the lower bound $\mathbb{E}_{M'}[\hat{\mu}] \geq (2 - \sqrt{2})\mu(G)$.

This completes the proof of approximation.

Adaptive adversaries. With adaptive adversaries, although we do not know if a maximal matching M can be maintained in polylog time, we can maintain a $(0.5-\varepsilon)$-approximate matching in polylog n. Now it no longer holds that $L_1 = 0$, and using Monday's sublinear time algorithm, we can get an estimator of size at least

$$(1-\varepsilon)\max\left(\frac{1}{2}\mu(G) + \frac{1}{2}|L_1|, \left(1 - p - \frac{2p}{1+p}\right)|M| + \frac{2p}{1+p}\mu(G) - \frac{2p}{1+p_1}L_1\right)$$

For $p = 0.3$, this is minimized for $L_1 = 0.084\mu(G)$ and results in a 0.542-approximation.

2 Doing better than a $\frac{2}{3}$-approximation for bipartite graphs

The main result of this section is the following.

Theorem 2. *There exists a dynamic algorithm that maintains a* $\left(\frac{2}{3} + 10^{-6}\right)$*-approximation of* $\mu(G)$ *with worst-case update time* $\mathcal{O}\left(\min(m^{1/4}, \sqrt{\Delta}) + \text{polylog } n\right)$*, where* Δ *denotes the maximum vertex degree in* G*.*

We can effectively ignore the $\sqrt{\Delta}$ term above, since the "marking algorithm" we are familiar with [Solomon, 2018] provides us a subgraph that has a maximum degree $\leq \sqrt{m}$ and is a $(1+\varepsilon)$-approximation to $\mu(G)$, and we can build atop it.

Constructing a semi-dynamic algorithm is enough; we can translate it into a dynamic algorithm in the following manner. Suppose we have a query time of Q and an update time of U, where we output a matching size $\mu' = [(\alpha - \varepsilon)\mu(G), \mu(G)]$ Then, since we can tolerate an extra absolute error of $\mathcal{O}(\varepsilon)$, we can run the query once every εn updates to get a dynamic algorithm that has a worst-case update time $\mathcal{O}\left(\max\left(\frac{Q}{\varepsilon n}, U\right)\right)$.

In this section, we will achieve $U = \tilde{\mathcal{O}}(\sqrt{\Delta})$, $Q = \tilde{\mathcal{O}}(n\sqrt{\Delta})$, and $\alpha = \frac{2}{3} + \Omega(1)$.

In our quest, we will leverage the help of our good friend, the Edge-Constrained Degree Subgraph (EDCS):

Definition 1. *For any integers $\beta > \beta_- \geq 1$, a subgraph H of G is called a (β, β_-)-EDCS of G if*

1. $\deg_H(u) + \deg_H(v) \leq \beta \ \forall (u,v) \in H$, *and*
2. $\deg_H(u) + \deg_H(v) \geq \beta_- \ \forall (u,v) \in G \setminus H$

Two simple observations to boot: H is *sparse* since it has at most $n\beta$ edges, and if $\deg_H(u) + \deg_H(v) < \beta_-$ for some edge (u,v) in G, then $(u,v) \in H$. Further, recall from last week's lectures the following result.

Proposition 3. *If $\beta = \Theta(\sqrt{\Delta} poly(1/\varepsilon))$ and $\beta_- \geq (1-\varepsilon)\beta$, then H is a $\left(\frac{2}{3} - \mathcal{O}(1)\right)$-sparsifier for G. Moreover, the edges of H as well as a $(1-\varepsilon)$-approximate maximum matching of H can be maintained deterministically with worst-case update time $\mathcal{O}(\sqrt{\Delta} poly(1/\varepsilon))$.*

This already gets us pretty close to our goal. However, we are on the wrong side of $\frac{2}{3}$.

3 Jumping across two-thirds

Our motivation is as follows. In the "normal" case, our approximation ratio will not be as bad as our bound suggests, and we will already do better than $\frac{2}{3}$. For the instances where this does not hold, something drastic must happen. This drastic something should show up as some kind of structure or result we can exploit to do better.

We therefore characterize these tight instances as follows.

Lemma 4. *Let H be a $(\beta, (1-\varepsilon)\beta)$-EDCS of G. Let V_{low} denote the set of vertices with $\deg_H(\cdot) \in [0, 0.2\beta]$,m and V_{mid} those with $\deg_H(\cdot) \in [0.4\beta, 0.6\beta]$. Further define $H' := H[V_{low}, V_{mid}]$ as the subgraph induced by the edges between the two subsets. If $\mu(H) \leq \left(\frac{2}{3} + \Omega(1)\right)\mu(G)$, then*

1. $\mu(H') \geq \left(\frac{2}{3} - \mathcal{O}(1)\right)\mu(G)$
2. *For any matching M in H',*

$$\mu(G[V_{mid} \setminus V(M)]) \geq \left(\frac{1}{3} - 800\delta\right)\mu(G)$$

3. $V_{mid} < 8\mu(G)$

Here, $\delta \in (2\varepsilon, \frac{1}{60})$ is an arbitrary parameter.

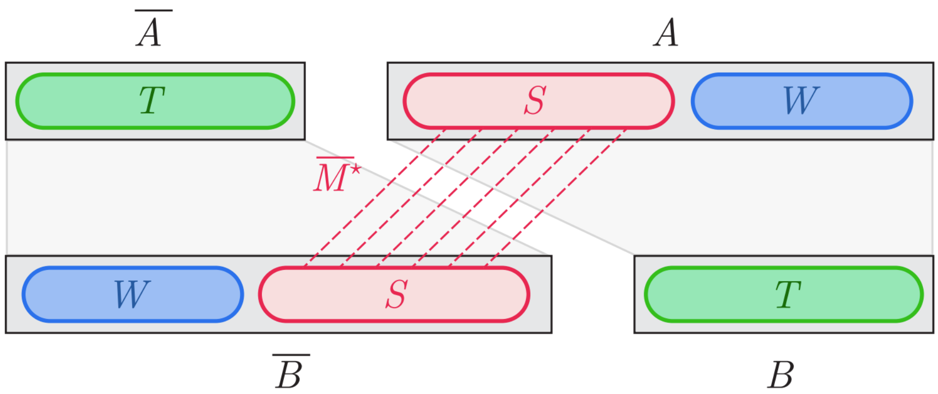

Figure 1: Characterization of a tight EDCS instance.

We will not prove this proposition but just highlight the idea. The approach is to find a witness A ($\subseteq L$, say) for the Extended Hall Theorem on G (see Figure 1). Fix a maximum matching M^* of G, and define $\bar{A} := L \setminus A$, $B := N_H(A)$, $\bar{B} = R \setminus B$, $S := V(\bar{M}^*), W := (A \cup \bar{B}) \setminus S$, and $T := \bar{A} \cup B$. The key insight then is to show that

1. Almost all vertices in S and T belong to V_{mid}.
2. Almost all vertices in W belong to V_{low}

3.1 The semi-dynamic algorithm

We maintain in the background $H := (\beta, (1-\varepsilon)\beta)$-EDCS$(G)$, and M_H, a $(1-\varepsilon)$-approximate maximum matching of H as previously described. Then the query algorithm proceeds as follows.

1. Get V_{low} and V_{mid}.
2. Find $H' := H[V_{\text{low}}, V_{\text{mid}}]$
3. Find a $(1-\varepsilon)$-maximum matching $M_{H'}$ in H'.
4. With $F := G[V_{\text{mid}} \setminus V(M_{H'})]$, define

$$g := |\text{GMM}(F, \pi)|$$

   for any arbitrary permutation π of edges in F. Here, GMM is the greedy maximal matching algorithm.

5. Return

$$\tilde{\mu}' := \max\left(|M_H|, |M_{H'}| + g\right)$$

Runtime. The runtimes of various steps are as follows.

- Steps 1 and 2 take $\mathcal{O}(n\beta)$.
- With the Hopkroft-Karp algorithm, step 3 takes $\mathcal{O}\left(\frac{\#\text{edges}}{\varepsilon}\right) = \mathcal{O}(n\beta\varepsilon))$.
- For the last two steps, we have compute $\tilde{g}$ in $\mathcal{O}(n/\varepsilon^3)$ such that

$$\mathbb{E}[g] - \varepsilon|V_{\text{mid}}| \leq \tilde{g} \leq \mathbb{E}[g]$$

Overall, the runtime is $\mathcal{O}_\varepsilon(n\beta) = \mathcal{O}_\varepsilon(n\sqrt{\Delta})$.

Approximation Guarantee. We only need to worry about the case where

$$\begin{aligned}(1-\varepsilon)\left(\frac{2}{3}+\delta\right)\mu(G) &\geq |M_{H'}| \\ &\geq (1-\varepsilon)\mu(H) \\ \Rightarrow \mu(H) &\leq \left(\frac{2}{3}+\delta\right)\mu(G)\end{aligned}$$

Applying our characterization in Lemma 4,

$$\begin{aligned}&|M_{H'}| + g - \varepsilon|V_{\text{mid}}| \\ \geq (1-\varepsilon)\left(\frac{2}{3}-\mathcal{O}(1)\right)\mu(G) &+ \frac{1}{2}\left(\frac{1}{3}-\mathcal{O}(1)\right)\mu(G) - \varepsilon\mathcal{O}(1)\mu(G) \\ &= \left(\frac{2}{3}+\frac{1}{6}-\mathcal{O}(1)\right)\mu(G)\end{aligned}$$

Which is a better-than-2/3 approximation. The paper chooses $\delta = 1.9 \cdot 10^{-6}$ and $\varepsilon < \delta/100$.

References

Shay Solomon. Local Algorithms for Bounded Degree Sparsifiers in Sparse Graphs. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 52:1–52:19, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-060-6. doi: 10.4230/LIPIcs.ITCS.2018.52. URL `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2018.52`.