CS 388R: Randomized Algorithms, Fall 2021 October 19th, 2021

Lecture 15: Online Bipartite Matching

Prof. Eric Price *Scribe: Youguang Chen, Yihan Zhou*

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

1 Overview

In the last lecture, we discussed the matching problems in bipartite graphs, especially the perfect matching in d-regular bipartite graphs. We introduced the online bipartite matching problem at the end of last lecture.

In this lecture, we will study different approximation algorithms for online bipartite matching:

- a greedy deterministic algorithm gives $\frac{1}{2}-$approximation and no deterministic algorithm can provide a better guarantee,
- an intuitive randomized algorithm gives $(\frac{1}{2}+o(1))-$approximation in the worst case,
- a randomized algorithm given by Karp, Vazirani, and Vazirani [KVV90] achieves an expected approximation ratio of $1-\frac{1}{e}$.

2 Introduction

Given a bipartite graph $G=(U,V,E)$ with n left vertices U and n right vertices V. We can think U as "customers", V as "merchants", and edges E exist between merchants and the customers they would like to advertise to. The online matching for this bipartite graph is the following:

- The merchants V are known in advance while the customers U come in one at a time (the corresponding edges are only revealed after a customer arrives).
- Once a customer $u \in U$ and its edges come in, we have to find which merchant from V to match u. Note that we can't go back and make changes.

Our goal is to match as many vertices as possible in this online setting, i.e. we want our online algorithm to perform as close to an offline algorithm, with full information, as possible. To measure the performance of an online algorithm A in comparison to the optimal (offline) algorithm, we use *competitive ratio.*

Definition 1 (Competitive Ratio)**.** *Given an online bipartite matching algorithm A, the approximation ratio is defined as*

$$R(A)=\liminf_{I}\frac{\mathbb{E}[|A(I)|]}{|OPT(I)|}, \tag{1}$$

where I is the input instances (bipartite graph and arrival order of customers), $A(I)$ is the matching of algorithm A, and thus $OPT(I)$ is the matching of the optimal offline algorithm, i.e. maximum matching.

3 Deterministic greedy algorithm

Let us consider the following simple greedy algorithm at first. We fix some ranking on the merchants and whenever a customer (left vertex) u arrives, match u to the highest priority unmatched merchant in u's neighborhood list $N(u)$.

Proposition 2. *The competitive ratio of the greedy algorithm is at least $\frac{1}{2}$.*

Proof. Given a bipartite graph $G = (U, V, E)$, let M_G be the matching of the online greedy algorithm, OPT be the optimal offline matching (i.e. maximum matching). For any vertex $u \in U \cup V$, define $\alpha_u := \mathbb{1}\{u \text{ is matched in } M_G\}$. It is easy to observe that $2|M_G| = \sum_{u \in U} \alpha_u + \sum_{v \in V} \alpha_v$.

Now we claim that $\forall (u, v) \in OPT$, $\alpha_u + \alpha_v \geq 1$. On the contrary, suppose $\alpha_u = \alpha_v = 0$ and $(u, v) \in OPT$, this means that u and v are not matched in M_G but there is an edge connecting them, which contradicts the greedy algorithm property. Thus,

$$|OPT| = \sum_{(u,v) \in OPT} 1 \leq \sum_{(u,v) \in OPT} \alpha_u + \alpha_v \leq \sum_{u \in U} \alpha_u + \sum_{v \in V} \alpha_v = 2|M_G|. \tag{2}$$

By Equation 1, the competitive ratio of the greedy algorithm is at least $\frac{1}{2}$. $\square$

We can show that any deterministic algorithm for online bipartite matching has the competitive ratio upper bounded by $\frac{1}{2}$.

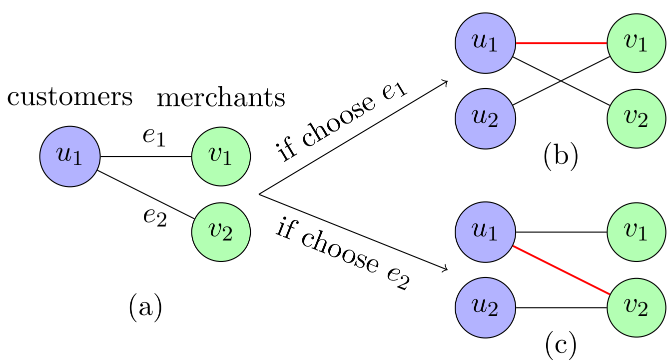

Figure 1: Adversary input for deterministic algorithm.

Proposition 3. *For any deterministic algorithm of online bipartite matching, the competitive ratio is at most $\frac{1}{2}$.*

Proof. Consider a 2 customers and 2 merchants matching problem, suppose the first customer u_1 is arrived and it has connections to both of the merchants (see Figure 1(a)). If the deterministic algorithm chooses e_1 (i.e. customer u_1 is matched with merchant v_1), the adversary can construct the input for the 2nd customer u_2 as the graph shown in Figure 1(b). Otherwise, if customer u_1 is matched with merchant v_2, the adversary can construct the input for the 2nd customer u_2 as the graph shown in Figure 1(c). In both cases, u_2 can not be matched, but the optimal matching can match both of the two customers, therefore the competitive ratio for any deterministic algorithm is no more than $\frac{1}{2}$. $\square$

4 Intuitive randomized algorithm

In section 3, we introduced a deterministic greedy algorithm which fixes a global ranking on the merchants. What if instead, whenever a customer arrives, we randomly pick a merchant in the customer's neighborhood? This can be viewed as randomly assigning each customer its own ranking of the merchants.

Unfortunately, this intuitive randomized algorithm achieves an expected approximation ratio in the worst case approaching $\frac{1}{2}$ as n grows large, so it does not improve on the deterministic algorithm by much.

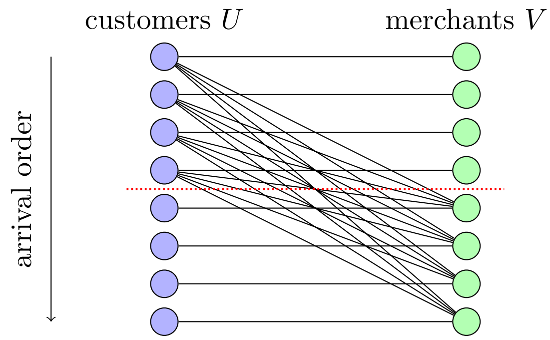

Figure 2: An example of the adversary input with 8 customers and 8 merchants for the naive randomized algorithm.

Proposition 4. *The competitive ratio of the intuitive randomized algorithm is at most* $\frac{1}{2} + o(1)$.

Proof. We first construct a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$. Assume that the ith arrived customer (left vertex) is u_i. Customer u_i and merchant v_i are connected for all $i \in [n]$. In addition, the first $\frac{n}{2}$ customers are fully connected to the second half of merchants $v_{n/2+1}, v_{n/2+2}, \cdots, v_n$. A sketch of such construction for $n = 8$ is presented in Figure 2.

It can be observed that there is a perfect matching in such construction and thus $|OPT| = n$. For any customer u_i ($i \leq \frac{n}{2}$), define $E_i :=$ # of unmatched merchants connected to u_i. It's easy to observe that at most $i - 1$ of the second half merchants are matched before u_i arrives. Thus

$E_i \geq \frac{n}{2} + 1 - (i-1) = \frac{n}{2} - i + 2$. Therefore, for the first half of customers, we expect

$$\begin{aligned}
\mathbb{E}[\#\text{ of } u_i \textbf{ not} \text{ matched with } v_i\, (i \leq n/2)] &= \sum_{i=1}^{\frac{n}{2}} 1 - \mathbb{P}[u_i \text{ matched with } v_i] \\
&= \sum_{i=1}^{\frac{n}{2}} 1 - \frac{1}{E_i} \\
&\geq \sum_{i=1}^{\frac{n}{2}} 1 - \frac{1}{\frac{n}{2} - i + 2} \\
&= \frac{n}{2} - \sum_{j=2}^{\frac{n}{2}+1} \frac{1}{j} \\
&\geq \frac{n}{2} - \log\big(\frac{n}{2} + 1\big). \qquad (3)
\end{aligned}$$

Let M_I be the matching found by the intuitive randomized algorithm, then

$$E[|M_I|] \leq n - \mathbb{E}[\#\text{ of } u_i \textbf{ not} \text{ matched with } v_i\, (i \leq n/2)] \leq \frac{n}{2} + \log\big(\frac{n}{2} + 1\big). \qquad (4)$$

Since $|OPT| = n$, the competitive ratio of the intuitive randomized algorithm is at most $\frac{1}{2} + o(1)$. □

5 Ranking algorithm

In this section we introduce the randomized online bipartite algorithm with ranking developed by [KVV90]. This algorithm has an approximation ratio better than $\frac{1}{2}$. At the beginning of the algorithm, we assign a "rank" to each merchant. This rank would incorporate consistency in our algorithm so it would work better than the naive algorithm in the previous section. Let σ be a permutation on $[n]$. The KVV algorithm runs as follows:
(1) Randomly generate a permutation σ on all the merchants.
(2) When a customer comes in, assign they to the merchant v with minimal rank $\sigma(v)$.

Before we analyse this algorithm, let's first introduce some notations. Let $A(G, \pi, \sigma)$ denote the KVV algorithm on input graph G (offline), customer arrival order π and merchant rank permutation σ. The three parameters G, π and σ determine the output of the algorithm. Let U denote the set of all customers and V denote the set of all merchants. We use u to denote a customer and v to denote a merchant unless mentioned otherwise. WLOG, when analysing the algorithm, we assume there always exists a perfect matching M (the worst approximation ratio always appear in a graph with perfect matching). We say a customer u is matched to merchant $M(u)$ and a merchant v is matched to a customer $M(v)$ by this perfect matching. We introduce a few lemmas to facilitate our proof.

Lemma 5. *Let $H = G - \{v\}$ where $v \in V$. Let π_H and σ_H denote ordering π and σ restricted to H respectively. Then $A(H, \pi_H, \sigma_H) = A(G, \pi, \sigma)$ + augmenting path of increasing (σ, π).*

Proof. After the merchant v is removed, the customer $M(v)$ will not be matched to v, but some other merchants with higher rank, let's call it u. Then the customer $M(u)$ with rank higher than

$M(v)$ will be forced to match with a merchant with rank higher than u. This process will continue when no more matching is possible and this gives an augmenting path consisting of increasing order merchants and customers. $\square$

Lemma 6. *Let $u \in U$ and $v = M(u)$. If v is unmatched in A, then u is matched by A to some v' such that $\sigma(v') < \sigma(v)$.*

Proof. This lemma is trivial by the matching mechanism of the algorithm. $\square$

5.1 Original proof

Now we present a critical lemma that will essentially prove the correctness of this algorithm. The lemma presented here is correct but its proof in the original paper [KVV90] is **incorrect**. The mistake was found after 17 years in 2007. For pedagogical purpose, we will give the original **wrong** proof and fix it with a revision.

Lemma 7. *Let x_t denote the probability that the vertex of rank $\sigma(v) = t$ is matched by A, then*

$$1 - x_t \le \frac{1}{n} \sum_{s \le t} x_s.$$

Now we give the **incorrect** proof below.

Proof. Let $v \in V$ such that $\sigma(v) = t$ and $u = M(v)$. The graph G and the customer ordering is fixed, so is the perfect matching M. The merchant ordering is generated uniformly. Therefore, v is distributed uniform over V and thus u is distributed uniform over U. Let $R_{t-1} \subseteq U$ be customers served by merchants v' with $\sigma(v') < t$. Then $\mathbb{E}(|R_{t-1}|) = \sum_{s \le t-1} x_s$. If v is unmatched, then u is matched to merchants with smaller rank. This means if v is unmatched, $u \in R_{t-1}$. Then

$$\begin{aligned} 1 - x_t &= \mathbb{P}(v \text{ unmatched}), \\ &\le \mathbb{P}(u \in R_{t-1}), \\ &= \mathbb{E}\left(\frac{|R_{t-1}|}{n}\right), \\ &= \frac{1}{n} \sum_{s \le t-1} x_s. \end{aligned} \tag{5}$$

$\square$

Given Lemma 3, we proceed to prove the approximation ratio. Let $S_t := \sum_{i=1}^{t} x_t$ denote the sum of x_t's. By Lemma 3, we have

$$1 - (S_t - S_{t-1}) \le \frac{S_t}{n}.$$

Rearrange and we get

$$S_t \cdot \frac{n+1}{n} \ge S_{t-1} + 1.$$

Rearrange again and

$$\begin{aligned}
S_t &\geq (S_{t-1}+1)\left(\frac{n}{n+1}\right), \\
&\geq \left(\frac{n}{n+1}\right) + \left(\frac{n}{n+1}\right)^2 + \cdots + \left(\frac{n}{n+1}\right)^t, \\
&= \left(\frac{n}{n+1}\right) \frac{1-\left(\frac{n}{n+1}\right)^t}{1-\left(\frac{n}{n+1}\right)}, \\
&= n\left(1-\left(1-\frac{1}{n+1}\right)^t\right).
\end{aligned}$$

Then we can calculate the approximation ratio

$$\begin{aligned}
\text{Approximation ratio} &= \frac{\mathbb{E}(\text{number of matched customers})}{n}, \\
&= \frac{S_n}{n}, \\
&\geq \left(1-\left(1-\frac{1}{n+1}\right)^n\right), \\
&\geq 1-\frac{1}{e}, \\
&\approx 63\%.
\end{aligned}$$

5.2 Revised proof

We present a revision of the proof in this subsection. The proof of Lemma 3 is wrong because equation (5) does not hold. Although u is uniformly distributed, it is dependent on the permutation σ. The set R_{t-1} is also dependent on σ so u and R_{t-1} are not independent. To fix this issue, we first propose the following lemma.

Lemma 8. *Let $u \in U$ and $v = M(u)$. Let σ' be a permutation that $\sigma'(v) = t$ and $\sigma^{(i)}$ be the permutation σ' with v moved to rank i. If v is unmatched in σ', then u is matched to $v^{(i)}$ in $\sigma^{(i)}$ such that $\sigma^{(i)}(v^{(i)}) \leq t$ for any choice of i.*

Proof. By lemma 2, u is matched in σ'. Then u is also matched in $\sigma^{(i)}$ because deleting an unmatched merchant won't affect the matching and inserting a merchant will only increase the matching. We go from the permutation $\sigma^{(i)}$ to σ' by first removing v from $\sigma^{(i)}$ and then inserting v to rank t. By Lemma 1, removing v from $\sigma^{(i)}$ will match u to some merchant k such that $\sigma^{(i)}(k) > \sigma^{(i)}(v^{(i)})$. Then we add v back to rank t and get σ'. Since v is unmatched in σ', adding it back won't change the matching. Note that $\sigma'(k) \geq \sigma^{(i)}(k) - 1$ because the rank of $\sigma^{(i)}(k)$ either decreases by 1 or stays the same. By Lemma 2, we have $t \geq \sigma'(k)$. This gives $t \geq \sigma'(k) \geq \sigma^{(i)}(k) - 1 \geq \sigma^{(i)}(v^{(i)})$ as desired. □

Now we show the correct proof of Lemma 3. We choose uniformly random permutation σ and uniformly random merchant v. Now v and σ are independent. Let σ' be the permutation σ with

v moved to rank t. Let $u = M(v)$. By Lemma 4, if v is unmatched in σ', then u is matched to merchant $\tilde{v}$ in σ such that $\sigma(\tilde{v}) \leq t$. Now we have

$$\begin{aligned}
1 - x_t &= \mathbb{P}(v \text{ is unmatched in } \sigma'), \\
&\leq \mathop{\mathbb{P}}_{v,\sigma}(\sigma(\tilde{v}) \leq t), \\
&= \mathop{\mathbb{E}}_{\sigma}\left(\mathop{\mathbb{P}}_{v}(\sigma(\tilde{v}) \leq t|\sigma)\right), \\
&= \mathop{\mathbb{E}}_{\sigma}\left(\mathop{\mathbb{P}}_{u}(\sigma(\tilde{v}) \leq t|\sigma)\right), \\
&= \mathop{\mathbb{E}}_{\sigma}\left(\frac{|R_t|}{n}\right), \\
&= \frac{\mathbb{E}(|R_t|)}{n}, \\
&= \frac{1}{n}\sum_{s \leq t} x_s.
\end{aligned}$$

The set R_t is the same as the previous definition. The issue is now fixed because u is only dependent on v and is independent of σ. The set R_t is only dependent on σ and is independent of v. With the correct proof of Lemma 3, the rest of the analysis follows.

References

[KVV90] Richard Karp, Umesh Vazirani, and Vijay Vazirani. An Optimal Algorithm for On-line Bipartite Matching. *Proceedings of STOC 1990*, 352–359, 1990.