Online Stochastic Matching
CMSC 858F: Algorithmic Game Theory
Fall 2010

Barna Saha, Vahid Liaghat

Abstract

This summary is mostly based on the work of Saberi et al. [1] on online stochastic matching problem proposed by Feldman et al. [2] as a model of display ad-allocation. We are given a bipartite graph; one side of the graph corresponds to a fixed set of bidders with budget limit of one and the other side represents the set of possible adword types. At each time step, an adword is sampled independently from the given distribution and it needs to be matched upon its arrival to an available bidder. The goal is to maximize the size of the matching.

Algorithms with a competitive ratio better than $1 - 1/e$ are known under the assumption that the expected number of arriving adwords of each type is integral. In [1] Saberi et al. present an online algorithm for the general (non-integral) problem with a competitive ratio of 0.702. A key idea of the algorithm is to collect statistics about the decisions of the optimum offline solution using Monte Carlo sampling and use those statistics to guide the decisions of the online algorithm. They also show that no online algorithm can have a competitive ratio better than 0.823.

1 Previous Work

Bipartite matching problems are central in algorithms and combinatorial optimization and arise naturally in several applications such as resource allocation, scheduling, and online advertising. We study a natural variation of bipartite matching problem motivated in the context of online advertising: suppose we are given a bipartite graph $G(Y, Z, E)$ where Y is the set of stochastic nodes (or adwords) and Z is the set of non-stochastic nodes (or bidders). At times $t = 1, 2, \ldots, b$, an adword of type $y \in Y$ is chosen independently at random from a given distribution. The algorithm can assign the adword to at most one of the available bidders that are adjacent to it. Furthermore, each bidder can be matched to at most one adword. The goal is to maximize the expected number of matched bidders at time b.

The online matching problem was first studied by Karp, Vazirani, and Vazirani [3] in the adversarial model, i.e. when the adwords are chosen by an adversary instead of a random process. They gave a simple and elegant randomized algorithm that achieves a competitive ratio of $1 - 1/e$ and further showed that no algorithm can achieve a better ratio.

A different model is the online, stochastic one called the iid model, where adwords Y arrive online according some *known* probability distribution (with repetition). In this iid model, the greedy algorithm achieves an approximation ratio of $1 - 1/e$ [4, 5]. Feldman et al.[2] (and later [6]) used a very interesting combinatorial algorithm to show that it is possible to do better than $1 - 1/e$ when the arrival rate of every adword, that is the expected number of times it appears, is integral. This assumption, even though not very restrictive for the display ad allocation, is somewhat unnatural. For example, when the distribution is uniform, it requires $b/|Y|$ to be an integer. They also showed that there is no $1 - o(1)$ approximation algorithm for this setting. Later Bahmani and Kapralov [6] improved the upper and lodwer bounds of Feldman et al. to 0.902 and 0.699 respectively in the same setting. Also, they showed that for d-regular graphs, a simple randomized algorithm achieves a competitive ratio of $1 - O(1/\sqrt{d})$.

Another stochastic model is the random order model where we assume that the distribution is unknown, but adwords arrive in a random order. This has proved be an important analytical construct for other problems such as secretary-type problems where worst cases are inherently difficult. It is known that in this case even the greedy algorithm has a (tight) competitive ratio of $1 - 1/e$ [4]. Further, no deterministic algorithm can achieve approximation ratio better than 0.75 and no randomized algorithm better than 0.83 [4].

A close line of work to the online matching is the online b-matching and the Adwords problem [7, 8]. Mehta et al. [7] developed a $1 - 1/e$ online algorithm in the adversarial case. Recently, Devanur and Hayes [8] improved the competitive ratio to $1 - \epsilon$ in the stochastic case where the sequence of arrivals is a random permutation or it consists of iid samples.

We present the first algorithm for the stochastic iid version of this problem in its general form (i.e. the arrival rates are not necessary integral) by Saberi et al [1] which improves the $1 - 1/e$ competitive ratio.

2 Problem Definition & Overview

Let $G(Y, Z, E)$ be a bipartite graph where Y is the set of stochastic nodes (or adwords) and Z is the set of non-stochastic nodes (or bidders). There is a rate r_y associated to every type of ball $y \in Y$. The online stochastic matching problem is as follows: at times $t = 1, 2, \ldots, b$, an adword of type $y \in Y$ is chosen randomly and with probability proportional to r_y. The algorithm can assign this adword to at most one of the available bidders that are adjacent to it; each bidder can be matched to at most one adword. The goal of the algorithm is to maximize the expected number of matched bidders at time b.

Without loss of generality, we assume that $\sum_{y \in Y} r_y = b$, thus the expected number of adwords of type y in the sequence is r_y. Also, we assume that $r_y \leq 1$; if a node has a rate greater than 1, we can easily split it into a set of identical nodes with rates at most 1.

We will study two classes of algorithms: non-adaptive and adaptive. A non-adaptive algorithm is equivalent to an ordering of the neighbors $N(y)$ of every node $y \in Y$. If $z_1, z_2, \ldots, z_{|N(y)|}$ is such an ordering for y, then the k-th time an adword of type y arrives, the algorithm will allocate it to bidder z_k if it is available. If $k > |N(y)|$ or z_k is full then the adword will not be allocated. On the other hand, adaptive algorithms can choose the assignment of every adword when it arrives.

We will present the two algorithms in [1] and will compare them to the optimum offline solution. Given the sequence of arrived adwords $\omega = (y_1, y_2, \ldots, y_b)$, one can compute the optimum allocation, $\mathbf{OPT}(\omega)$, in polynomial time by solving a maximum matching problem. Fix a particular maximum matching algorithm and let $F(\omega) : E \to \{0, 1\}$ be the vector indicating which edges are used in the optimum allocation given ω. Clearly, $\mathbf{OPT}(\omega) = 1^T F(\omega)$ and the competitive ratio of an online algorithm $\mathbf{ALG}$ is defined as $\frac{\mathbb{E}[\mathbf{ALG}]}{\mathbb{E}[\mathbf{OPT}]}$. In our case, both $\mathbf{ALG}$ and $\mathbf{OPT}$ are concentrated around their expected values, therefore the above competitive ratio is fairly robust (see Feldman et al. [2] for a more detailed discussion).

One of the key ideas in designing the two algorithms in [1] is to approximately compute the distribution imposed by the optimum offline solution and use that distribution to guide the decisions of the online algorithm. Using Monte Carlo sampling, one can compute $f(y, z)$, the probability that the optimum offline algorithm allocates an adword of type y to a bidder z, for every y and z. Without loss of generality, we can assume f is fractional matching.

In section 3 we will present the first algorithm in [1] for the special case when $\forall_{y \in Y} r_y = 1$. The algorithm writes f as a distribution over integral matchings and samples two matchings M_1 and M_2 from it. Then, in the online phase, it will use these two matchings for allocating the arriving adwords to the bidders. The analysis of the algorithm is much shorter and simpler than both [2, 6]. All these algorithms are non-adaptive, in the sense that they decide the allocation of all the adword types in advance before they appear. We present a simple example to show that no non-adaptive algorithm can achieve a competitive ratio better than $1 - 1/e$ when the arrival rates are non-integral.

In section 4 we will present the second result of [1] which is an adaptive algorithm that obtains a competitive ratio of 0.702 for arbitrary rates. Unlike the non-adaptive algorithms, the adaptive algorithm decides the allocation of each arriving adword based on the current state of the bidders. In particular, when an adword arrives the algorithm samples two neighbor bidders from a *joint* distribution and tries to match it to the first bidder; if the bidder is already matched the algorithm tries the second bidder. This is the first algorithm that beats the $1 - 1/e$ ratio in the general form. The adaptivity of the algorithm imposes a lot of dependencies in the distribution of matched bidders and because of that our analysis is somewhat intricate and for the proof we will only refer to [1].

Saberi et. al [1] presented an example that gives an upper bound of 0.823 on the competitive ratio of any deterministic or randomized online algorithm. For analyzing this example, they used the expected size of a maximum matching of a random bipartite graph recently computed by [9] in the context of Random SAT and cuckoo hashing. In section 5 we will only present a simpler example in the same paper for the problem with integral rate. This example shows an upper bound of 0.86 on any online algorithm in that settings.

The algorithms will crucially use the optimum offline solution for making decisions. In particular, define

$$f = \sum_{\omega} F(\omega)\mathbb{P}(\omega) \tag{1}$$

where $\mathbb{P}(\omega)$ is the probability of the sequence $\omega = (y1, y2, \ldots, yb)$. By definition, f is a convex combination of matchings and therefore it is in the convex hull of the matchings of G. We will refer to f as the fractional matching defined by **OPT**. For each edge $e = (y, z) \in E$, f_e is the probability that an adword of type y is allocated to bidder z by the optimum offline algorithm. Similarly we define the fractional degree of a node to be $f_v = \sum_{e \sim v} f_e$ for $v \in Y \cup Z$.

Proposition 1 *(Proposition 2.1 [1]) The vector f is a fractional matching in G. i.e.*

$$f_y \leq r_y \leq 1, y \in Y, \text{ and } f_z \leq 1, z \in Z \tag{2}$$

Moreover, for $e = (y, z)$, we have $f_e \leq 1 - e^{-r_y} + o(1/b)$.

Proof. Given ω, let $N_y(\omega)$ be the number of adwords of type y in ω. Clearly $\sum_{e \sim y} F_e(\omega) \leq N_y(\omega)$. Taking expectations from both sides results in the first inequality in 2. Similarly, the second inequality in 2 can be proved by noting that in any instance of the problem, z can be matched to at most one adword. Finally, for $e = (y, z)$, we have

$$f_e \leq \mathbb{P}(N_y(\omega) \geq 1) = 1 - (1 - \frac{r_y}{b})^b \leq 1 - e^{-r_y} + o(1/b)$$

. $\square$

Throughout the paper, we will assume that b is sufficiently large so that $o(1/b)$ is negligible. We will need to compute f_e for every edge e. Obviously, f_es can be computed by enumeration in time $O(|Y|^b)$. It is also easy to see that $\mathbb{E}[\mathbf{OPT}]$ and $f(e)$ for all $e \in E$, can be approximated with great accuracy using Monte Carlo method. **OPT** is an integral random variable which is in interval $[0, b]$, hence its variance is upper-bounded by b^2. Therefore, $\mathbb{E}[\mathbf{OPT}]$ can be estimated with error of $o(1/b)$, by averaging over $O(b^3)$ independent samples of the process. A similar argument shows that with $O(|E|^2 b^4)$ samples of ω in equation 1, with high probability, one can compute the vector f with accuracy within $o(1/b|E|)$. In the rest of the paper, for simplicity of notation, we will assume that we have estimated f accurately and ignore $o(.)$ terms.

Since f is a fractional matching, it can be written as a convex combination of at most $|Y| + |Z| + 1$ integral matchings. The following corollary is therefore trivial but very essential for the two algorithms:

Corollary 1 *(Corollary 2.1 [1])It is possible to efficiently and explicitly construct (and sample from) a distribution μ on the set of matchings in G such that*

$$\sum_{M \ni e} \mu(M) = f_e, \forall_{e \in E}$$

3 A Non-Adaptive Algorithm

In this section, we will analyze a simple non-adaptive algorithm for the special case where all rates are one, i.e., $r_y = 1, \forall_{y \in Y}$ given in [1]. This is the setting studied in Feldman et al. [2]. This algorithm and its analysis is simpler and more intuitive than [2]. It also gives a slightly better competitive ratio. This non-adaptive algorithm has some similarities with the online algorithm that Feldman et al. Both algorithms start by computing two matchings M_1 and M_2 offline; we use the first matching, only for the first arrived adword of each type and the second one only for the second arrivals. In particular, when the first adword of type y arrives it will be allocated to the bidder matched to y in M_1, and when the second adword arrives, we will allocate it via M_2. If the corresponding bidders are already full, the adwords will be dropped. Note that the probability that there are more than two adwords of each type y in the sequence of arrivals is very small.

On the other hand, the method used for constructing these matchings is different from [2]. Feldman et al. find M_1 and M_2 by decomposing the solution of a maximum 2-flow of G into two disjoint matchings (since all the rates are one, the expected graph is simply G). However, we will sample our matchings from the distribution μ defined by the optimum solution f.

Algorithm 1 ***The Online Non-adaptive Algorithm***

- **Offline Phase**

  1. Compute the fractional matching f, and the distribution μ using Corollary 1.
  2. Sample two matchings M_1 and M_2 from μ independently; set M_1 (M_2) to be the first (second) priority matching.

- **Online Phase**

1. When the first adword of type y arrives, allocate it through the first priority matching, M_1.
2. Similarly, when an adword of type y arrives for the second time, allocate it through the second priority matching, M_2.

In the rest of this section, we analyze Algorithm 1, and show that its approximation ratio is 0.684. Let X_z be the random variable indicating the event that bidder z is matched with an adword during the run of the algorithm. We analyze the competitive ratio of the algorithm by comparing $\mathbb{E}[X_z]$ with f_z:

$$\frac{\mathbb{E}[\mathbf{ALG}]}{\mathbb{E}[\mathbf{OPT}]} = \frac{\sum_{z\in Z}\mathbb{E}[X_z]}{\sum_{z\in Z} f_z} \geq \min_{z\in Z}\frac{\mathbb{E}[X_z]}{f_z}$$

Consider any $z \in Z$, and with a slight abuse of notation let $M_1(z)$ denote the stochastic node matched to it in M_1. More precisely, if $(y,z) \in M_1$, define $M_1(z) = \{y\}$, and if z is not saturated in M_1, define $M_1(z) = \phi$; similarly define $M_2(z)$. Note that z is saturated by M_1 (or M_2) with probability f_z, but if $M_1(z) = M_2(z)$, bidder z will only be used for the first arrived ball and effectively it is not saturated by M_2. Given M_1 and M_2, $\mathbb{E}[X_z|M_1, M_2]$ can be computed similar to [[2], section 4.2.2] by considering the following cases:

$$\mathbb{E}[X_z|M_1,M_2] = \begin{cases} 0 & \text{if } M_1(z) = M_2(z) = \phi \\ 1 - 1/e & \text{if } M_1(z) \neq \phi, \{M_1(z) = M_2(z)\} \\ 1 - 1/e & \text{if } M_1(z) \neq \phi, M_2(z) = \phi \\ 1 - 2/e & \text{if } M_1(z) = \phi, M_2(z) \neq \phi \\ 1 - 1/e^2 & \text{if } M_1(z) \neq \phi, M_2(z) \neq \phi, M_1(z) \neq M_2(z) \end{cases}$$

By substituting this into $\mathbb{E}[X_z]$ we get:

$$\mathbb{E}[X_z] = f_z(2 - 3/e) - f_z^2(1 + 2/e^2 - 3/e) - (1/e - 2/e^2)\sum_{e\sim z} f_e^2$$

It remains to prove a lower bound on the value of the last equation. Lemma 3.1 in [1] shows that the value of last equation is at least $0.684 \times f_z$ resulting the first theorem:

Theorem 1 *Assuming all the rates are* 1*, the solution of Algorithm 1 is within* 0.684 *of the optimum offline solution.*

4 The Adaptive Algorithm

In the analysis of the non-adaptive algorithm presented in the previous section, we assumed that the arrival rates of all adwords are integral and in particular, they are at least one. This is a crucial assumption. If the rates r_y's are not bounded from below, the probability of receiving a second adword of the same type can become arbitrary low and the competitive ratio of the algorithm can get very close to $1 - 1/e$. This is the case for all non-adaptive algorithms: In Proposition 5.1 in [1], Saberi et al. show that no non-adaptive algorithm can achieve a competitive ratio better than $(1 - 1/e)$ when the sampling rates are not necessarily integral.

In this section, we will analyze a simple adaptive algorithm that will have a better performance when the sampling rates are arbitrary. The algorithm is very simple: when an adword of type y arrives, it samples two neighboring bidders z_1 and z_2 from a *joint* distribution. If z_1 is available then y is matched to z_1. Otherwise, the algorithms will check z_2 and match y to it if it is available.

The joint distribution from which z_1 and z_2 are chosen, is determined in advance for every adword type y and it has the following properties: *(i)* The probability that z_1 is equal to z is equal to $f(y,z)$. The same is true for z_2. Recall that rates are normalized such that $\sum_{y\in Y} r_y = b$ and thus f is a fractional matching. *(ii)* Given (i), the joint distribution is such that the probability of $z_1 = z_2$ is minimized. In what follows, we will present one simple joint distribution with these properties.

Suppose $e_1, \dots, e_k$ are the edges incident to y, and without loss of generality assume $f_{e_1} \geq \dots \geq f_{e_k}$. Also define a dummy edge e_{k+1}, that is connected to a dummy non-stochastic node z_{k+1}, with $f_{e_{k+1}} = r_y - f_y$. Note that $f_{e_k} + 1$ is the probability that **OPT** drops a ball of type y. We will construct two different partitions of the interval $I_y = [0, r_y]$, $\mathcal{I}_y$ and $\mathcal{J}_y$. In order to get property (i), $\mathcal{I}_y$ is divided between e_i's proportional to f_{e_i}'s. And to get property (ii), $\mathcal{J}_y$ is obtained by shifting the subintervals of $\mathcal{I}_y$ to the left by f_{e_1}. Formally, partitions $\mathcal{I}_y$ and $\mathcal{J}_y$ are defined as follows:

- Partition $\mathcal{I}_y$: let $I_{y,l} = [\sum_{j=1}^{l-1} f_{e_j}, \sum_{j=l}^{l} f_{e_j}]$, $1 \leq l \leq k+1$.
- Partition $\mathcal{J}_y$: let $J_{y,l} = [\sum_{j=2}^{l-1} f_{e_j}, \sum_{j=2}^{l} f_{e_j}]$, $2 \leq l \leq k+1$, and $J_{y,1} = [r_y - f_{e_1}, r_y]$.

The outline of the algorithm is presented in Algorithm

Algorithm 2 ***The Online adaptive Algorithm***

- **Offline Phase**

  1. Compute the fractional matching f.
  2. For each $y \in Y$ construct the two functions $z_1(.)$ and $z_2(.)$ by defining the corresponding partitions $\mathcal{I}_y$ and $\mathcal{J}_y$.

- **Online Phase**

  1. If an adword of type $y \in Y$ arrives, choose a number x uniformly at random from interval $[0, r_y]$.
  2. First try to match it with $z_1(x)$, if it was full, try to match it with $z_2(x)$.

In Theorem 4.1 of [1], Saberi et al. proved that this algorithm guarantees a competitive ration of 0.702.

Theorem 2 *For any graph G and arbitrary set of rates $\{r_y, y \in Y\}$, the competitive ratio of Algorithm 2 is at least 0.702.*

5 Upper Bounds for Online Algorithms

Saberi et al. proved different upper bounds by presenting three examples. The results of these examples are presented below:

- **Integral Setting:** There exists an instance of the online stochastic matching problem with integral rates for which no online algorithm can achieve an expected competitive ratio better than $1 - 1/e^2 \simeq 0.86$. [Proposition 5.2 [1]]

- **General Setting** There is an instance of the online stochastic matching problem with small rates, $r_y = o(1)$, for which no non-adaptive randomized algorithm can achieve a competitive ratio better than $1 - 1/e$. [Proposition 5.1 [1]]

- **General Setting** There is an instance of the online stochastic matching problem for which no algorithm can achieve a competitive ratio better than 0.823. [Proposition 5.3 [1]]

References

[1] V. H. Manshadi, S. Oveis Gharan, A. Saberi. *Online Stochastic Matching: Online Actions Based on Offline Statistics*. In SODA, 2011.

[2] J. Feldman, A. Mehta, V. S. Mirrokni, S. Muthukrishnan. *Online stochastic matching: Beating 1-1/e*. In FOCS, pages 117126, 2009.

[3] R. M. Karp, U. V. Vazirani, V. V. Vazirani. *An optimal algorithm for on-line bipartite matching*. In STOC, pages 352358. ACM, 1990.

[4] G. Goel, A. Mehta. *Online budgeted matching in random input models with applications to adwords*. In SODA, pages 982991, 2008.

[5] S. Alaei, A. Malekian. *Maximizing sequence-submodular functions*, manuscript. 2009.

[6] B. Bahmani, M. Kapralov. *Improved bounds for online stochastic matching*. In ESA, pages 170181, 2010.

[7] A. Mehta, A. Saberi, U. Vazirani, V. Vazirani. *Adwords and generalized online matching*. J. ACM, 54(5):22, 2007

[8] N. R. Devanur and T. P. Hayes. *The adwords problem: online keyword matching with budgeted bidders under random permutations*. In EC, pages 7178, 2009.

[9] M. Dietzfelbinger, A. Goerdt, M. Mitzenmacher, A. Montanari, R. Pagh, M. Rink. *Tight thresholds for cuckoo hashing via xorsat*. SIAM Journal on Computing, 2009.