CSC2421 Topics in Algorithms: Online and Other Myopic Algorithms Fall 2019

Allan Borodin

November 20, 2019

Announcements

- There is no lecture on Wednesday, November 27 The final lecture will take place December 4.
- Today we will start with Carolyn discussing results in Chapter 6
- I will finish the lecture by continuing with some topics in stochastic online online algorithms.

We begin where we left off discussing bipartite matching in i.i.d. input model and, in particular, the Feldman et al bipartite matching algorithm in the i.i.d. input model.

Bipartite matching in the i.i.d. model

For the unweighted bipartite maximum matching (BMM) we observed that Ranking can be viewed as a deterministic algorithm in the ROM model that achieves expected competitive ratio $1 - \frac{1}{e}$. It then follows immediately that deterministic Fixed Rank algorithm achieves (at least) the same $1 - \frac{1}{e}$ ratio in the unknown (and hence also known) i.i.d. model for the BMM problem. It is currently unknown if there is a better deterministic approximation for the BMM problem in the ROM model. However, there are significantly better approximations for both the unweighted (and even edge weighted) bipartite matching problems in the known i.i.d. model.

For the edge weighted case, the i.i.d. approximations are therefore much better than what can be achieved in the ROM model since in the ROM model we know that the ratio $\frac{1}{e}$ is asymptotical the best we can do.

For the unweighted (or offline vertex weighted) case the known results are better than what is known for the ROM model so far, we do not know what is the best possible ROM approximation nor do we know what is the best i.i.d. approximation.

The known i.i.d. model for the BMM problem

In the known i.i.d. model, an adversary first chooses a *type graph* $G = (L, R, E)$ and a distribution $p : L \to [0, 1]$ on the LHS nodes. In this case, the nodes in L are also referred to as types. The type graph together with the distribution is given to the algorithm in advance.

In the known i.i.d. model, an actual input instance $\hat{G} = (\hat{L}, R, \hat{E})$ is a random variable and is generated from G as follows. The right hand side R is the same in G and $\hat{G}$, but the left-hand-side of $\hat{G}$ consists of n i.i.d. samples from p. Thus, say a given node $\hat{\ell} \in \hat{L}$ has type $\ell \in L$, then the neighbors of $\hat{\ell}$ in $\hat{G}$ are the same as the neighbors of ℓ in G. The graph $\hat{G}$ is presented to the algorithm in the vertex arrival model (the order of vertices is the same as the order in which they were generated).

Note that a particular type ℓ can be absent altogether or can be repeated a number of times in $\hat{G}$. We refer to $\hat{G}$ as the *instance graph*. Feldman et refer to $\hat{G}$ as the *realization graph*.

Known i.i.d. distributions with integral types

A known i.i.d. problem is said to have *integral types* if the expected number of times a particular type occurs is integral. We will denote the number of times type ℓ occurs in an instance by the random variable Z_ℓ. Then the condition of integral types is that $\mathbb{E}[Z_\ell] = p(\ell) \cdot n \in \mathbb{Z}$.

While the parameters $|L|, |R|$, and n can all be different, the most common setting is $n = |L|$. This assumption together with integral types implies that without loss of generality one can take p to be the uniform distribution on L (by duplicating types as necessary). An additional common assumption is that $|L| = |R|$. In that case we talk about a single parameter $n = |L| = |R| = m$.

The first algorithm to beat the $1 - 1/e$ barrier in the known i.i.d. model is due to Feldman et al. [2009]. Their algorithm achieved a competitive ratio of .73. We rushed the explanation of Feldman et al so we will repeat this first i.i.d. result for the BMM problem.

The Feldman et al algorithm for BMM in the i.i.d. model

The first algorithm to beat the $1 - 1/e$ barrier in the known i.i.d. model is due to Feldman et al. [2009]. The algorithm has a preprocessing stage followed by the online stage. In the preprocessing stage, the algorithm solves the following modification of the standard network flow problem for biparite matching: add two new nodes s and t, add directed edges from s to r for each $r \in R$, and add directed edges from ℓ to t for each $\ell \in L$, orient the rest of the edges in G from RHS to LHS (these edges will be called the graph edges). Each outgoing edge from s, as well as each incoming edge into t, has capacity 2. The rest of the edges have capacities 1. We denote this flow network by $\widetilde{G}$.

The algorithm of Feldman et al. finds an integral optimal solution to this network flow problem. The subgraph induced by the graph edges with positive flow on them has maximum degree 2. The last step of the preprocessing stage is to apply a so-called *blue-red-decomposition* to this subgraph to obtain a blue semi-matching M_b and a red matching M_r as will be described.

The Feldman et al algorithm for BMM in the i.i.d. model continued

In the online stage, the algorithm receives online nodes in the i.i.d. fashion and matches them as follows: if a node of type i arrives for the first time, the algorithm tries to match it to $M_b(i)$. If $M_b(i) = \perp$ or $M_b(i)$ has been previously matched, the algorithm leaves the current node unmatched. If a node of type i arrives for the second time, the algorithm tries to match it to $M_r(i)$. Otherwise, a node of type i is left unmatched.

Here is the blue-red decomposition:

---

procedure BlueRedDecomposition($G = (L, R, E)$)

Color edges of the cycles alternating blue and red.

Color edges of the odd-length paths alternating blue and red, with more blue than red.

For the even-length paths that start and end with nodes in R, alternate blue and red.

For the even length paths that start and end with nodes in L, color the first two edges b
then alternate red, blue, red, blue, etc.

return (semi-matching formed by blue edges, matching formed by red edges).

Comments on BMM in the i..i.d. model

Note that the Feldman et al algorithm is not "greedy" in the sense that it can leave online nodes unmatched even when there are available unmatched offline nodes.

Any online algorithm for the unweighted BMM can be made greedy and doing so cannot worsen the competitive ratio. In practice, converting an algorithm so as to make it greedy will substantially improve the number of nodes matched. (I am posting a paper in this regard that presents an experimental study comparing different algorithms in the known i.i.d. model. The paper also provides a brief explanation of known i.i.d. algorithms for BMM.)
The Feldman et al algorithm achieved an expected (with respect to the distribution) approximation ratio $\alpha = \frac{1-\frac{2}{e^2}}{\frac{4}{3}-\frac{2}{3e}} \approx .67$ and this ratio is tight for their algorithm. In fact, Feldman et al prove a stronger positive resulti, namely:

$\forall \epsilon > 0$, with probability at least $1 - e^{\Omega(n)}$, as long as $OPT = \Omega(n)$, their algorithm ALG achieves $\frac{ALG}{OPT} - \epsilon \geq \alpha$.

Improvements on the Feldman et al i.i.d. ratio

Subsequently a number of papers build on the idea of using two matchings to derive improved approximation ratios. Here is a partial history of results all based on the "best of two matchings" idea initiated in Feldman et al.

- Bahmani and Kapralov [2010] For integral types $.699 - \epsilon$.
- Manshadi et al [2011] For arbitrary types, .705 and .707 for integral types. Furthermore, they establish the currently best inapproximation .823 for known i.i.d. arrivals implying the same for the ROM model (which is the best inapproximation known for the ROM model).
- Jaillet and Lu [2014] For integral types, ,729 and .706 for arbitrary types. They also have a .725 ratio for offline vertex weighted graphs.
- The current best ratio is .,7299 due to Brubach et al [2016]. They also have the best ratio .705 for the edge weighted case.

Prophet Inequalities

We have seen that knowing distributional information can fascilitate better approximations. We now consider a stochastic version of the secretary problem. Specifically, suppose we have n random real valued variables $X_1, X_2, \ldots, X_n$ where each X_i is defined by an adversarially chosen distribution [1] denoted by F_i. Abusing notation, we will use X_i to refer to the distribution. The X_i are independent but not necessarily identical. In the *prophet inequalities* problem, the X_i arrive online (i.e., in a sequence determined adversarially) at which point a weight value w_i is drawn from the distribution X_i. Just as in the classical secratary problem, we must either accept w_i or discard it permanently. The goal is construct an algorithm $\mathcal{A}$ which maximizes the expectation of the chosen variable which we will call w^*. Thus, like the secretary problem, we are looking for a *stopping rule*.

[1] The distributions can be continuous so that F_i is the cumulative distribution function or F_i can be discrete so that each possible element has a given probability. For definitness, we will assume a discrete probability distribution but the results and analysis can be modified to apply to continuous distributions.

Defining the competitive ratio for prophet inequalities

We know that if the values (and not the distributions) are determined adversarially and not known in advance then the approximation can be arbitarilty bad. What approximation can be achieved if we know the distributions for each X_i but not the values w_i until they are drawn from the distribution?

As in our previous discussion of i.i.d. distributions, the approriate benchmark is $\mathbb{E}[\max_i w_i]$ rather than comparison against the maximum w_i in the sequence. That is, we wish to bound the ratio $\frac{\mathbb{E}[w^*]}{\mathbb{E}[\max w_i]}$.

NOTE: In the adversarial setting of the secretary problem (i.e. where the adversary selects the weights of each online candidate), the problems of maximizing the probability of selecting the winner and maximizing the competitive ratio are equivalent. However, in the i.i.d. setting this equivalence is no longer applicable. Most work in the i.i.d. setting focuses on the distributional competitive ratio and not the probability of choosing the best solution.

The optimal stopping rule for prophet inequalities

It turns out that there is again (like the secretary problem) a simple uniform stopping rule. Let $T = \frac{\mathbb{E}[\max_i w_i]}{2}$. The uniform stopping rule is to accept the first $w_i > T$ and then stop. This threshold stopping rule turns out to be the best online algorithm.

This last statement deserves some clarification. When we say "online" in the context of a stochastic algorithm, we mean that the distributions are adversarial but known in advance while the drawn values occur online.

Aside: For any sequence of distributions with finite support, there is an online stopping rule that can be efficiently determined by an offline dynamic programming algorithm. The main point is that the algorithm must know the entire sequence of distributions in advance.

The ratio for prophet inequalities

Theorem

Let $\mathcal{A}$ be the online algorithms using the above stopping rule. For every set of independent distributions $X_1, X_2, \ldots, X_n$, $\mathcal{A}$ has competitive ratio at least $\frac{1}{2}$. Furthermore this is an optimal stopping rule in the sense that for any $\epsilon > 0$, there are distributions such that the expected value of any online algorithm is at most $(\frac{1}{2} - \epsilon) \cdot \mathbb{E}[max_i w_i]$.

For the negative example, let X_1 be a deterministic distribution with value 1 and let X_2 be the distribution such that $w_2 = \frac{1}{\epsilon}$ with probility ϵ and 0 with probability $1 - \epsilon$ for some arbitrarily small $\epsilon > 0$. An online algorithm can accept $w_1 = 1$ and obtain that value or it can decide to reject w_1 and the obtain $\mathbb{E}[X_2] = 1$. So that any online algorithm will obtain value 1. On the other hand.

Fact

For the above distributions, $\max\{X_1, X_2\} = (2 - \epsilon)$.

The proof of the prophet inequalities competitive ratio

Here first is the proof of the fact establishing the negative result. Let Y be any random variable and let Z be a random $\{0, 1\}$ indicator variable with $p = Prob[Z]$. Then

$$\mathbb{E}[Y] = \mathbb{E}[Y|Z] \cdot p + \mathbb{E}[Y|\bar{Z}] \cdot (1 - p)$$

The claim follows by setting $Y = \max\{X_1, X_2\}$, and letting Z be the indicator variable for $X_1 \geq X_2$ so that $p = prob[Z] = (1 - \epsilon)$.

For the positive result, we need to lower bound $\mathbb{E}[\mathcal{A}]$ and upper bound $T = \mathbb{E}[Y]/2$ for $Y = \max_i w_i$.

The proof for the positive result is in the text.

Prophet secretary problem

Esfandiari et al [2017] consider the prophet secretary problem, a natural combination of the prophet inequalities and secretary problems. As in the prophet inequalities, we have n random variables $X_1, \ldots, X_n$ drawn from n adversarially chosen independent (but not necessarily identical) distributions $F_1, \ldots, F_n$. But now rather than these X_i being presented in an adversarial online order, they are input in random order (i.e. the ROM model as in the secretary problem). To be more precise, the i^{th} input seen by the algorithm is $(X_{\pi(i)}, F_{\pi(i)})$ where π is a random permutation of $[1, n]$.

For the prophet secretary problem, Esfandiari et al [2017] show no single uniform threshold can achieve a competitive ratio better than the $\frac{1}{2}$ prophet inequalities ratio. Using n different non-adaptive thresholds, they achieve competitive ratio $1 - \frac{1}{e}$. They also show that no prophet secretary algorithmn can have a ratio better than .73.

Stocahstic probing and stochastic rewards

A different type of stochastic optimization involves *stochastic probing*. In stochastic probing, input items come with a probability, and to reveal the actual item, an algorithm= must probe the item.

In some probing variants, once the item is probed, the algorithm is *committed* to the item. In other variants there may be some associated number of probes that are allowed,

Lets consider the BMM problem in the *stochastic-commit* model and call this the *stochastic-rewards* problem as it is sometimes referred to.

In the stochastic rewards problem, when an online vertex u arrives all adjacent edges $e = (u, v)$ are labelled with a probability $p_e = p_{(u,v)}$ which denotes the probability that the edge exists. Furthermore, u has a patience constraint t_u.

Stochastic rewards continued

In order to attempt to match u to some available offline v, an online algorithm must probe the edges adjacent to u in some order. The patience constraint bounds the number of times edges adjacent to u can be probed.

Once a probed edge actually exists, the online algorithm is commtted to using that edge in the matching. The offline vertices may or may not be weighted.

When the patience parameter $t_u = 1$ for all online vertices, this is called the stochastic rewards problem without patience.

What is the benchmark against which the online =algorithm is competing? One urealistic benchmark is that the adversary-offline algorithm get to see the instance graph. Then consider an adversarial type graph with one online node having paience 1 and n offline vertices such that each edge has probability $1/n$. Then the graph will have an edge in the instance graph with probability $1 - [(1 - \frac{1}{n}]^n \approx 1 - \frac{1}{e}$ which is then the expected offline reward whereas the online algorithm only has expected reward $1/n$.

A better benchmark

One often uses an optimal LP fractional solution to some apprpriate relaxation of the matching problem to obtain an upper bound on a benchmafrk solution.

It seems like a good "natural" extension to the non-stochastc benchmark (i.e. an optimal solution) is the following: The adversary sets the probabilities of the edges and the benchmark can reveal edges in any order. In particular, it can look at the online nodes in any order. The only offline constraint is that the benchmark is also limited by the patience constraints of each edge.

The stochastic rewards without patience originated in Mehta and Panigraph [2012]. (Their LP benchmark that can be argueded is too powerful.)

I am following Brubach et al [2019] who consider the stochastic rewards with patience problem with offline vertices weighted. They articulated the "natural benchmark" above and use a more restricted LP to upper bound such a benchmark.

End of Lecture 10

We ended on slide 17. We will start the final lecture by restating the stochastic rewards problem.