Online Algorithms

LECTURE 7

Inscribed by: Amin Jalali

1. Review

Recall from previous lectures that we have already covered the following models:

Adversarial Model

Random Order Model where unlike the adversarial model, the order is random and we have a near 1 approximation of the optimal value as the result. This is in fact equivalent to *sampling without replacement.*

Today, we will discuss a model where we perform *sampling with replacement* and, as we will see, it will lead to better approximation results.

2. i.i.d. model with unknown distribution

Recall the bipartite matching problem where an offline vertex set $\mathcal{L}$ is given. Moreover, the vertices in $\mathcal{R}$ are sampled (independently and identically) from a given probability distribution with finite support $\hat{\mathcal{R}}$ to which the algorithm is blind.

- Each $j \in \hat{\mathcal{R}}$ is identified by its neighbors in $\mathcal{L}$
- The probability of picking vertex j is $p(j)$ with $\sum_{j\in\hat{\mathcal{R}}} p(j) = 1$

This probability distribution corresponds to the so called *distribution graph* with left hand nodes $\mathcal{L}$ and right hand nodes $\hat{\mathcal{R}}$ with corresponding probabilities from $p(\cdot)$. We will have the *actual graph* via m i.i.d. samples of this distribution graph.

Remark. We assume that m, the number of vertices that arrive, is known to the algorithm.

Notice that in this analysis, both OPT and ALG are random variables. In fact, the value of OPT depends on the arriving vertices. Thus, we will look for a guarantee of the form

$$\mathbb{E}[\mathrm{ALG}] \;\geq\; \gamma \cdot \mathbb{E}[\mathrm{OPT}] \tag{2.1}$$

as well as a statement in the concentration form as

$$\mathrm{ALG} \geq \gamma \cdot \mathrm{OPT} \;\text{ holds with high probability.} \tag{2.2}$$

Definition 1. Define the *expected instance* similar to distribution graph where each node j has a *supply* of $p(j) \cdot m$ which is equal to the expected number of the times we see j in the online process.

Definition 2. Denote the deterministic value $\overline{\mathrm{OPT}}$ as the OPT value over the *expected instance* such that

$$\overline{\mathrm{OPT}} \;\geq\; \mathbb{E}[\mathrm{OPT}]\,.$$

This definition allows us to only prove the following sufficient condition to ensure (2.1),

$$\mathbb{E}[\mathrm{ALG}] \geq \gamma \cdot \overline{\mathrm{OPT}}\,. \tag{2.3}$$

With these definitions we have the following *expected LP*,

$$\begin{aligned} \overline{\mathrm{OPT}} = \max \quad & \textstyle\sum_{i,j} x_{ij} & \\ \text{s.t} \quad & \textstyle\sum_j x_{ij} \leq 1 & \text{for all } i \in \mathcal{L} \\ & \textstyle\sum_i x_{ij} \leq m \cdot p(j) & \text{for all } j \in \hat{\mathcal{R}} \\ & x_{ij} \geq 0 & \text{for all } i, j\,. \end{aligned} \tag{2.4}$$

Lemma 3 (Main result)**.** $\overline{\mathrm{OPT}} \geq \mathbb{E}[\mathrm{OPT}]\,.$

Proof. In OPT, define the following set of indicator variables

$$X_{ij} = \begin{cases} 1 & \text{if some copy of } j \text{ is matched to } i \\ 0 & \text{otherwise}\,. \end{cases}$$

X_{ij} is a random variable because it depends on the sample we got. Moreover, define $x_{ij} = \mathbb{E}[X_{ij}]\,$. This implies

$$\begin{cases} \forall i: & \sum_j X_{ij} \leq 1 \Rightarrow \sum_j x_{ij} \leq 1 \\ \forall j: & \sum_i X_{ij} \leq \#\text{of times } j \text{ appears in } \mathcal{R} \Rightarrow \sum_i x_{ij} \leq p(j) \cdot m\,. \end{cases}$$

Thus, $\{x_{ij}\}_{i,j}$ is a feasible solution to the LP in (2.4) and we have

$$\sum_{i,j} x_{ij} = \mathbb{E}[\sum_{i,j} X_{ij}] = \mathbb{E}[\mathrm{OPT}] \leq \overline{\mathrm{OPT}}\,.$$

□

Theorem 4. *Greedy is* $(1 - \frac{1}{e})$*−competitive in the i.i.d. model with* unknown *(to the algorithm) distribution.*

Proof. Consider dividing $\mathcal{L}$ into the set of matched ($\mathcal{U}$) and unmatched vertices.

Notice that the probability that a new vertex in $\mathcal{R}$ can be matched is equal to the probability of that vertex having neighbor in $\mathcal{U}$. Following this definition, we present an auxiliary algorithm.

Hypothetical Algorithm: Given j, match it to i with probability $\frac{X_{ij}}{p(j)\cdot m}$. This is clearly worse than a greedy approach and we will use this fact later. Now, consider j which arrives with probability $p(j)$. Thus, for a given i, the probability that i gets matched in one step is

$$\sum_j p(j) \cdot \frac{X_{ij}}{p(j)\cdot m} = \frac{X_{ij}}{m}.$$

Therefore, the probability that this hypothetical algorithm matches something to $\mathcal{U}$ is equal to $\sum_{i\in\mathcal{U},\, j} \frac{X_{ij}}{m}$. This, from definition of X_{ij}, gives,

$$\mathrm{ALG}(t) = |\mathcal{L}\backslash\mathcal{U}| \geq \sum_{i\in\mathcal{U},\, j} X_{ij}$$

which together with $\overline{\mathrm{OPT}} = \sum_{i,\, j} X_{ij}$ gives $\overline{\mathrm{OPT}} - \mathrm{ALG}(t) \leq \sum_{i\in\mathcal{U},\, j} X_{ij}$. Thus, we have

$$\begin{aligned}\mathbb{E}\left[\mathrm{ALG}(t+1) \mid \mathrm{ALG}(t)\right] &= \mathrm{ALG}(t) + \mathbb{P}[\text{Greedy matches a vertex in } \mathcal{U}] \\ &\geq \mathrm{ALG}(t) + \mathbb{P}[\text{Hypothetical algorithm matches a vertex in } \mathcal{U}] \\ &\geq \mathrm{ALG}(t) + (\textstyle\sum_{i\in\mathcal{U},\, j\in\hat{\mathcal{R}}} X_{ij})/m \\ &\geq \mathrm{ALG}(t) + (\overline{\mathrm{OPT}} - \mathrm{ALG}(t))/m\end{aligned}$$

and results in

$$\begin{aligned}\mathbb{E}\left[\overline{\mathrm{OPT}} - \mathrm{ALG}(t+1) \mid \mathrm{ALG}(t)\right] &\leq \overline{\mathrm{OPT}} - \mathrm{ALG}(t) - (\overline{\mathrm{OPT}} - \mathrm{ALG}(t))/m \\ &= (\overline{\mathrm{OPT}} - \mathrm{ALG}(t))(1 - 1/m)\end{aligned}$$

which finally yields

$$\mathbb{E}\left[\overline{\mathrm{OPT}} - \mathrm{ALG}\right] \leq \overline{\mathrm{OPT}}(1-\tfrac{1}{m})^m \leq \tfrac{\overline{\mathrm{OPT}}}{e} \Rightarrow \mathbb{E}[\mathrm{ALG}] \geq (1-\tfrac{1}{e})\overline{\mathrm{OPT}}$$

□

Suggested Exercise. Generalize the presented approach to (Integral) Budgeted Allocation problem without the assumption $b_{ij} \ll B_i$.

3. B-Matching

Consider the B-Matching problem in which each vertex $i \in \mathcal{L}$ can be matched B_i times and each $j \in \mathcal{R}$ only once. We assume $B_i \geq K$.

Remark. We will show that as B_i increases, the guarantee goes to 1. However, on the contrary to the guarantees we saw before, where we get the improvement only for large enough B_i's, the guarantee kicks in from the beginning in here.

Assumptions: For the sake of simplicity assume $|\hat{\mathcal{R}}| = m$, and $p(j) = 1/m$ for all $j \in \hat{\mathcal{R}}$. In this case, the expected instance is simply an integral matching problem. Further, suppose that there exists a perfect matching in the expected instance; i.e. each j is matched to $M(j)$ and each $i \in \mathcal{L}$ is matched B_i times. This implies $\overline{\mathrm{OPT}} = \sum_i B_i = m$ and we need this to make sure that all the budgets get exhausted.

Pure Random Algorithm $\mathcal{P}$: This algorithm,

- knows the expected instance and the perfect matching,
- is non-adaptive; makes all the decisions ahead of time; makes its choice even if the vertex i has exhausted its budget (corresponding to throwing it away),
- always matches j to $M(j)$, but gets credit only if it makes at most B_i matches.

In fact, for all steps,

$$\mathbb{P}[i \text{ is matched in 1 step}] = \frac{B_i}{m} = \frac{\#\text{ of } j\text{'s matched to } i}{\text{total number of } j\text{'s}}$$

which is a uniform distribution.

This algorithm is in fact a *Balls and Bins process* and is independent of the graph and

- each i corresponds to a bin with capacity i, with $\sum_i B_i = m$,
- in each round we throw a ball into bin i with probability $\frac{B_i}{m}$,
- repeat m times.

Therefore, denoting by X_i the number of balls thrown into bin i, we have $\mathbb{E}[X_i] = B_i$ which is not considering the capacities. In fact, we are looking for $\mathbb{E}[\min\{X_i, B_i\}]$ instead. Considering

$$\mathbb{P}[X_i = \ell] = \binom{m}{\ell} \cdot \left(\frac{B_i}{m}\right)^{\ell} \cdot \left(1 - \frac{B_i}{m}\right)^{(m-\ell)}$$

we can calculate the desired expectation from

$$\mathbb{E}[\min\{X_i, B_i\}] = \sum_{\ell=1}^{B_i} \ell \cdot \mathbb{P}[X_i = \ell] + \sum_{\ell=B_i+1}^{m} B_i \cdot \mathbb{P}[X_i = \ell]\,.$$

Observe that $\mathbb{E}[\min\{X_i, B_i\}]$ is monotonically decreasing in m considering the fact that $\mathbb{E}[X_i] = B_i$ is fixed. Thus, we can look at the limit as

$$\lim_{m\to\infty} \mathbb{E}[\min\{X_i, B_i\}] = B_i - \sqrt{\frac{B_i}{2\pi}} = B_i(1 - \frac{1}{\sqrt{2\pi B_i}}) \geq B_i(1 - \frac{1}{\sqrt{2\pi K}})$$

which implies

$$\mathbb{E}[\mathcal{P}] = \sum_i B_i(1 - \frac{1}{\sqrt{2\pi K}}) = \overline{\mathrm{OPT}}(1 - \frac{1}{\sqrt{2\pi K}}) \geq \overline{\mathrm{OPT}}(1 - \epsilon)$$

for $\epsilon \geq 1/\sqrt{2\pi K}$ that is equivalent to

$$K \geq \frac{1}{\sqrt{2\pi\epsilon^2}}\,. \tag{3.1}$$

Remark. Compare this result to what we had before as

$$\frac{B_i}{b_i^{\max}} \geq \frac{o(n\log(mn))}{\epsilon^2}\,.$$

However, we started out with *unknown* distribution. As we will see in the sequel, we can get similar guarantees by designing an algorithm that does not know about the distribution.

3.1. Main Algorithm

We will define the algorithm inductively. Suppose we could magically start using the *Pure Random algorithm* ($\mathcal{P}$) after the tth step and denote such an algorithm by *hybrid algorithm*; i.e.

$$\mathcal{H}^t = \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{t-1},\ ?\ , \mathcal{P}_{t+1}, \mathcal{P}_{t+2}, \ldots \mathcal{P}_m\,.$$

Consider the following procedure: given j in the tth step, for any choice of i that is an unmatched neighbor of j, evaluate the expected number of matches in the remaining time for $\mathcal{H}^t$. Match j to i that maximizes this; i.e. match j to

$$\arg\max_{i:\, i\sim j} \left\{\mathbb{E}[\mathcal{H}^t] \mid \mathcal{A}_t = i, \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{t-1}\right\}$$

where $\mathcal{A}_t = i$ is equivalent to matching j to i at step t. This defines $\mathcal{A}_t$ and hence the algorithm. Considering,

$$\mathcal{H}^t = \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{t-1}, \mathcal{A}_t, \mathcal{P}_{t+1}, \mathcal{P}_{t+2}, \ldots, \mathcal{P}_m$$
$$\mathcal{H}^{t-1} = \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{t-1}, \mathcal{P}_t, \mathcal{P}_{t+1}, \mathcal{P}_{t+2}, \ldots, \mathcal{P}_m\,.$$

we have $\mathbb{E}\left[\mathcal{H}^t\right] \geq \mathbb{E}\left[\mathcal{H}^{t-1}\right]$ from the definition where $\mathcal{A}_t$ is always trying to do better than pure random algorithm. Listing the values from every step yields

$$\mathbb{E}[\text{ALG}] = \mathbb{E}\left[\mathcal{H}^m\right] \geq \mathbb{E}\left[\mathcal{H}^{m-1}\right] \geq \ldots \geq \mathbb{E}\left[\mathcal{H}^0\right] = \mathbb{E}[\mathcal{P}]. \tag{3.2}$$

Notice that even though we do not know the distribution, the *adaptiveness* of our algorithm leads to getting $\mathbb{E}[\text{ALG}]$ in the end.

However, the remaining question is how we can perform the aforementioned magical step. In fact, we can estimate the expected number of matches by its equivalence to a balls and bins procedure. To perform this calculation, we need the remaining capacity, the probability of match in one step, and the number of remaining steps.

Suggested Exercise. Extend the presented algorithm and analysis to Budgeted Allocation Problem with integral parameters. Here is the sketch of analysis: denote by X_i the sum of b_{ij}'s where $\mathbb{E}[X_i] = B_i$ and for each time step we have $\mathbb{E}[X_i^t] = B_i/m$. Then, $\mathbb{E}[\min\{X_i, B_i\}]$ is the smallest when $b_{ij} \in \{0, b_i^{\max}\}$. The algorithm is as follows: evaluate profit in the current step plus the expected remaining profit (assuming the condition on b_{ij}'s). The final guarantee is in the form

$$\text{ALG} \geq \text{OPT}(1 - \frac{1}{\sqrt{2\pi K}})$$

where $B_i/b_i^{\max} \geq K$. Observe that the analysis works for non-uniform distributions.

For this, look at the remaining budget plus to what we have in every step because in this case b_{ij}'s are not equal to one as above.