Algorithms and Uncertainty, Winter 2018/19	Lecture 3 (4 pages)

Online Set Cover: Fractional Algorithm

Instructor: Thomas Kesselheim

Last time, we introduced the online set cover problem and its fractional relaxation. Today, we will consider only the relaxed problem and devise online algorithms. So, our goal is to solve the following kind of linear program online.

$$\begin{aligned} \text{minimize} \sum_{S \in \mathcal{S}} c_S x_S & \\ \text{subject to} \sum_{S \colon e \in S} x_S \geq 1 & \qquad \text{for all } e \in U \\ x_S \geq 0 & \qquad \text{for all } S \in \mathcal{S} \end{aligned}$$

We have to maintain a feasible solution x to the linear inequalities. In the t-th step, we get to know the t-th constraint. Possibly, the solution x we had so far is infeasible now. In this case, we may only *increase* variables.

Recall the dual of the set cover LP

$$\begin{aligned} \text{maximize} \sum_{e \in U} y_e & \\ \text{subject to} \sum_{e \in S} y_e \leq c_S & \qquad \text{for all } S \in \mathcal{S} \\ y_e \geq 0 & \qquad \text{for all } e \in U \end{aligned}$$

We will use a primal-dual algorithm. That is, besides maintaining a primal solution x, we will also maintain a dual solution y. In step t, variable y_t is added to the dual LP and we can only set its value. We will make use of the following lemma.

Lemma 3.1. *If for all times t*

(a) The primal increase is bounded by β times the dual increase, that is

$$P^{(t)} - P^{(t-1)} \leq \beta(D^{(t)} - D^{(t-1)}) \ , \ \textit{where } P^{(t)} = \sum_{S \in \mathcal{S}} c_S x_S^{(t)} \textit{ and } D^{(t)} = \sum_{e \in U} y_e^{(t)}$$

(b) $\frac{1}{\gamma} y^{(t)}$ is dual feasible,

Then the algorithm is $\beta\gamma$-competitive.

1 Fractional Ski Rental

To understand the design principle of primal-dual online algorithms, we will consider the fractional variant of Ski Rental first. Recall the LP relaxation

$$\begin{aligned} \text{minimize } B x_{\text{buy}} + \sum_t x_{\text{rent},t} & \\ \text{subject to } x_{\text{buy}} + x_{\text{rent},t} \geq 1 & \qquad \text{for all } t \\ x_{\text{buy}}, x_{\text{rent},t} \geq 0 & \qquad \text{for all } t \end{aligned}$$

and its dual

$$
\begin{aligned}
\text{maximize } & \sum_t y_t \\
\text{subject to } & \sum_t y_t \leq B \\
& y_t \leq 1 && \text{for all } t \\
& y_t \geq 0 && \text{for all } t
\end{aligned}
$$

Note that the optimal dual solution is always to set $y_1 = \ldots = y_B = 1$ and $y_{B+1} = \ldots = y_m = 0$. Our primal-dual algorithm will use exactly this dual solution. So, in Lemma 3.1, Property (b) holds with $\gamma = 1$. We now have to figure out how to update the primal solution so as to keep β as small as possible.

The dual does not increase in steps $t > B$. Therefore, to maintain Property (a), we may not increase the primal objective in these steps.

In step $t \leq B$, we increase x_{buy} some way. We have to set $x_{\text{rent},t}$ to fill the gap between x_{buy} and 1. So,

$$x_{\text{buy}}^{(t)} + x_{\text{rent},t}^{(t)} = 1 \ .$$

At the same time, the increase of the primal objective function has to be bounded by $\beta \cdot y_t = \beta$. That is

$$B(x_{\text{buy}}^{(t)} - x_{\text{buy}}^{(t-1)}) + x_{\text{rent},t}^{(t)} = \beta \ .$$

In combination, these two equalities give us

$$(B-1)x_{\text{buy}}^{(t)} - Bx_{\text{buy}}^{(t-1)} = \beta - 1 \ ,$$

or equivalently

$$x_{\text{buy}}^{(t)} = \frac{\beta - 1}{B - 1} + \frac{B}{B-1} x_{\text{buy}}^{(t-1)} \ .$$

This recursion solves to

$$x_{\text{buy}}^{(t)} = \sum_{t'=0}^{t-1} \left(\frac{B}{B-1} \right)^{t'} \frac{\beta - 1}{B - 1} = \frac{1 - \left(\frac{B}{B-1}\right)^t}{1 - \left(\frac{B}{B-1}\right)} \frac{\beta - 1}{B-1} = (\beta - 1) \left(\left(\frac{B}{B-1} \right)^t - 1 \right) \ .$$

Recall that $x_{\text{buy}}^{(B)} = 1$ because we may not increase primal variables after step B. So, we have to have

$$(\beta - 1) \left(\left(\frac{B}{B-1} \right)^B - 1 \right) = 1 \ ,$$

which is equivalent to

$$\beta = \left(1 - \left(1 - \frac{1}{B} \right)^B \right)^{-1} \ .$$

Note that $\left(1 - \left(1 - \frac{1}{B}\right)^B\right)^{-1} \leq \left(1 - \frac{1}{\mathrm{e}}\right)^{-1} \approx 1.58$, so the algorithm is 1.58-competitive.

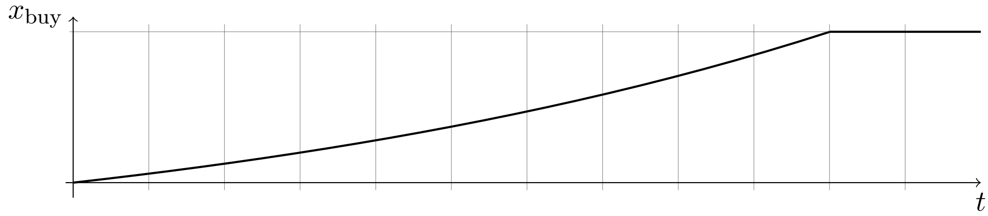

Figure 1: Increase of x_{buy} over time for $B = 10$.

2 Approach for Fractional Online Set Cover

When choosing $x^{(t)}$ and $y^{(t)}$, our primary goal is that they have similar objective-function values so that Property (a) in Lemma 3.1 holds with a small β.

So, let us figure out what we would like to do. Suppose we are in step t. We observe a new constraint $\sum_{S\colon e\in S} x_S \geq 1$ in the primal LP. In the dual, a new variable y_e arrives.

We have $\sum_{S\colon e\in S} x_S^{(t-1)} < 1$, otherwise we would not have to do anything. We will have to increase some variables to get a feasible $x^{(t)}$. Of course, $x^{(t)}$ will be more expensive than $x^{(t-1)}$. We reflect this additional cost in the value of $y_e^{(t)}$, all other dual variables remain unchanged.

Let us slowly increase x starting from $x^{(t-1)}$ and simultaneously y_e starting from 0. We do this in infinitesimal steps over continuous time.

We are at any point in time for which still $\sum_{S\colon e\in S} x_S^{(t-1)} < 1$. We increase x_S by dx_S. To account for the increased cost, we increase y_e by dy at the same time. The dual objective function increases by dy this way. This is at most $(\sum_{S\colon e\in S} x_S)dy$ because $\sum_{S\colon e\in S} x_S < 1$. Simultaneously, the primal objective function increases by $(\sum_{S\colon e\in S} dx_S)$. If we set $dx_S = (\frac{x_S}{c_S})dy$ for S such that $e \in S$, then these changes exactly match up.

Ideally, we would follow exactly this pattern. However, notice that we start from $x^{(0)} = 0$, so all increases would be 0. Therefore, let $\eta > 0$ be very small and set

$$dx_S = \frac{1}{c_S}(x_S + \eta)dy \quad .$$

This is a differential equation. We try a solution of the form $x_S = C_1 \mathrm{e}^{C_2 y} + C_3$. Then we have $\frac{dx_S}{dy} = C_2(x_S - C_3)$, so $C_3 = -\eta$, $C_1 = x_S^{(t-1)} + \eta$, $C_2 = \frac{1}{c_S}$. This way

$$x_S^{(t)} + \eta = \mathrm{e}^{\frac{1}{c_S} y_e^{(t)}} \left(x_S^{(t-1)} + \eta \right) \quad ,$$

where $y_e^{(t)}$ is the smallest value such that $x^{(t)}$ is a feasible solution to the first t constraints of the primal LP.

3 Algorithm

Let us now use the algorithmic approach above to design an algorithm for fractional online set cover.

For our algorithm, we set $\eta = \frac{1}{n}$ and initialize all $x_S = 0$. In the t-th step, when a new element e arrives, we introduce the primal constraint $\sum_{S:e\in S} x_S \geq 1$ and a dual variable y_e. We initialize $y_e = 0$ and update it as follows. For each S with $e \in S$ increase x_S from $x_S^{(t-1)}$ to $x_S^{(t)}$ by

$$x_S^{(t)} + \eta = \mathrm{e}^{\frac{1}{c_S} y_e^{(t)}} \left(x_S^{(t-1)} + \eta \right) \quad ,$$

where $y_e^{(t)}$ is the smallest value such that $x^{(t)}$ is a feasible solution

Theorem 3.2. *The algorithm is $O(\log n)$-competitive for fractional online set cover.*

Proof. We will verify the conditions of Lemma 3.1 with $\beta = 2$ and $\gamma = \ln(n+1)$.

We start by property (a). Consider the t-th step, let element e arrive in this step. We have to relate $P^{(t)} - P^{(t-1)} = \sum_S c_S (x_S^{(t)} - x_S^{(t-1)})$ to $D^{(t)} - D^{(t-1)} = y_e^{(t)}$. For set S such that $e \in S$, we have

$$x_S^{(t)} + \eta = \mathrm{e}^{\frac{1}{c_S} y_e^{(t)}} \left(x_S^{(t-1)} + \eta \right) \quad ,$$

and therefore

$$x_S^{(t-1)} + \eta = \mathrm{e}^{-\frac{1}{c_S} y_e^{(t)}} \left(x_S^{(t)} + \eta \right) \quad .$$

This lets us write the increase of x_S as follows

$$x_S^{(t)} - x_S^{(t-1)} = \left(x_S^{(t)} + \eta \right) - \mathrm{e}^{-\frac{1}{c_S} y_e^{(t)}} \left(x_S^{(t)} + \eta \right) = \left(1 - \mathrm{e}^{-\frac{1}{c_S} y_e^{(t)}} \right) \left(x_S^{(t)} + \eta \right) \leq \frac{1}{c_S} \left(x_S^{(t)} + \eta \right) y_e^{(t)} \quad .$$

This way, we can bound the primal increase by

$$P^{(t)} - P^{(t-1)} \leq \sum_{S: e \in S} c_S \frac{1}{c_S} \left(x_S^{(t)} + \eta \right) y_e^{(t)} = \sum_{S: e \in S} x_S^{(t)} y_e^{(t)} + \sum_{S: e \in S} \eta y_e^{(t)} \leq 2 y_e^{(t)} = 2(D^{(t)} - D^{(t-1)}) \quad ,$$

because $\sum_{S: e \in S} x_S^{(t)} = 1$ (otherwise we would have increased variables by too much) and $\sum_{S: e \in S} \eta \leq n\eta = 1$.

Now, we turn to property (b). Consider a fixed S. Let element e arrive in step t. By our algorithm if $e \in S$ then

$$y_e = c_S \ln(x_S^{(t)} + \eta) - c_S \ln(x_S^{(t-1)} + \eta) \quad ,$$

otherwise $x_S^{(t)} = x_S^{(t-1)}$.

So, when computing $\sum_{e \in S} y_e$, we might as well take the sum over all T steps as follows

$$\sum_{e \in S} y_e = \sum_{t=1}^{T} \left(c_S \ln(x_S^{(t)} + \eta) - c_S \ln(x_S^{(t-1)} + \eta) \right) = c_S \ln \left(\frac{x_S^{(T)} + \eta}{x_S^{(0)} + \eta} \right) \quad .$$

Furthermore, $x_S^{(0)} \geq 0$ because variables are never negative and $x_S^{(T)} \leq 1$ because it does not make sense to increase variables beyond 1. So

$$\sum_{e: e \in S} y_e \leq c_S \ln \left(\frac{1 + \eta}{\eta} \right) = c_S \ln(n+1) = \gamma c_S \quad .$$

□