Algorithms and Uncertainty, Winter 2017/18 Lecture 3 (4 pages)

Fractional Online Set Cover via LP Duality

Instructor: Thomas Kesselheim

Today, we will learn about a fundamental technique in the design of online algorithms. As our motivating example, we consider the set cover problem in its weighted variant. In the offline version, you are given a universe of m elements $U = \{1, \ldots, m\}$ and a family of n subsets of U called $\mathcal{S} \subseteq 2^U$. For each $S \in \mathcal{S}$, there is a cost c_S. Your task is to find a *cover* $\mathcal{C} \subseteq \mathcal{S}$ of minimum cost $\sum_{S \in \mathcal{C}} c_S$. A set $\mathcal{C}$ is a cover if for each $e \in U$ there is an $S \in \mathcal{C}$ such that $e \in S$. Alternatively, you could say $\bigcup_{S \in \mathcal{C}} S = U$.

We assume that each element of U is included in at least one $S \in \mathcal{S}$. So in other words $\mathcal{S}$ is a feasible cover. Otherwise, there might not be a feasible solution.

Note that the problem is NP-hard in the offline case, so this already limits our expectations. We will consider the online version, in which the universe U arrives online, one element at a time. Whenever an element is revealed, we get to know which sets $S \in \mathcal{S}$ it is contained in and have to make sure that it is covered, potentially by adding a set from $\mathcal{S}$ to $\mathcal{C}$. We may never remove sets from $\mathcal{C}$. Our goal is to eventually select sets so as to minimize $\sum_{S \in \mathcal{C}} c_S$.

1 LP Relaxation

We can state the set cover problem as an integer program as follows

$$\begin{aligned}
\text{minimize} \sum_{S \in \mathcal{S}} c_S x_S & & & \text{(minimize the overall cost)} \\
\text{subject to} \sum_{S \colon e \in S} x_S \geq 1 & & \text{for all } e \in \mathcal{U} & \quad \text{(cover every element at least once)} \\
x_S \in \{0, 1\} & & \text{for all } S \in \mathcal{S} & \quad \text{(every set is either in the set cover or not)}
\end{aligned}$$

We can relax the problem by exchanging the constraints $x_S \in \{0, 1\}$ by $0 \leq x_S \leq 1$. (These are the only constraints requiring integrality of the solution.) We get the following LP relaxation[1]

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

In the online problem, we know the variables and the objective function in advance. We get to know one constraint at a time and we have to maintain a feasible solution and we are not allowed to reduce the values of the variables. So the difficulty is that we do not know what constraints will come later when we choose which variables to increase.

For the time being, let us only consider the fractional problem, that is, the problem without the integrality constraints. We will first devise an algorithm to solve this problem online and later on use this algorithm to also derive solutions to the integral problem.

[1] We could also include that $x_S \leq 1$ for all S but this will not change the optimal solution as values greater than 1 do not make sense.

2 LP Duality

We will use LP duality for our algorithm. It is not necessary to know LP duality in its generality. What we need to know is that the dual LP gives us a *lower bound* on all feasible solutions. The dual of the Set Cover LP relaxation is

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

Lemma 3.1 (Weak Duality)**.** *Let x and y be feasible solutions to the primal and dual program respectively. Then $\sum_{S \in \mathcal{S}} c_S x_S \geq \sum_{e \in U} y_e$.*

Proof. We have $\sum_{e \in U} y_e \leq \sum_{e \in U} \left(\sum_{S \colon e \in S} x_S \right) y_e = \sum_{S \in \mathcal{S}} x_S \sum_{e \in S} y_e \leq \sum_{S \in \mathcal{S}} x_S c_S$. □

Example 3.2. *Consider $U = \{1, 2, 3\}$, $\mathcal{S} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$, $c_S = 1$ for all $S \in \mathcal{S}$. The optimal set cover solution has cost 2 because we need to take two sets. However, setting $x_{\{1,2\}} = x_{\{1,3\}} = x_{\{2,3\}} = \frac{1}{2}$ for all $S \in \mathcal{S}$ is a feasible solution to the LP relaxation of cost $\frac{3}{2}$. It is optimal because setting $y_1 = y_2 = y_3 = \frac{1}{2}$, we also have a solution to the dual LP of cost $\frac{3}{2}$. This means there cannot be a cheaper solution to the primal LP.*

3 Algorithmic Approach

Our algorithm will use LP duality. Namely, we will not only maintain (feasible) primal solutions $x^{(t)}$ but also (possibly infeasible) dual solutions $y^{(t)}$. Both start from $x^{(0)} = 0$ and $y^{(0)} = 0$.

Lemma 3.3. *If for all times t*

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

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

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

The the algorithm is $\alpha\beta$-competitive.

Proof. First, observe that by a telescoping-sum argument, we have $P^{(t)} = \sum_{t'=1}^{t} (P^{(t')} - P^{(t'-1)}) \leq \alpha \sum_{t'=1}^{t} (D^{(t')} - D^{(t'-1)}) = \alpha D^{(t)}$.

Let x^* be an optimal offline solution. Then, by weak duality, we know $\sum_{S \in \mathcal{S}} c_S x_S^* \geq \sum_{e \in U} y_e$ for any dual feasible y, in particular $y = \frac{1}{\beta} y^{(t)}$. So, $\sum_{S \in \mathcal{S}} c_S x_S^* \geq \frac{1}{\beta} \sum_{e \in U} y_e$.

Combined with $P^{(t)} \leq \alpha D^{(t)}$, we get $c^T x^{(t)} \leq \alpha \cdot b^T y^{(t)} \leq \alpha \beta c^T x^*$. This means exactly that the online solution $x^{(t)}$ is within an $\alpha\beta$ factor of the offline solution x^*. □

When choosing $x^{(t)}$ and $y^{(t)}$, our primary goal is that they have similar objective-function values so that Property (a) in Lemma 3.3 holds with a small α.

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

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

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

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

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.

4 Algorithm for Fractional Online Set Cover

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 $x_S = 0$. Whenever 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. While $\sum_{S: e \in S} x_S < 1$ do: For each S with $e \in S$ increase x_S by $dx_S = \frac{1}{c_S}(x_S + \eta) dy_e$.

Theorem 3.4. *The algorithm is $O(\log n)$-competitive for fractional online set cover.*

Proof. We will verify the conditions of Lemma 3.3 with $\alpha = 2$ and $\beta = \ln(n+1)$.

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

$$x_S^{(t)} - x_S^{(t-1)} = \int_0^{y_e^{(t)}} \frac{d}{dy} \left(\mathrm{e}^{\frac{y}{c_S}} \left(x_e^{(t-1)} + \eta \right) \right) dy = \int_0^{y_e^{(t)}} \frac{1}{c_S} \left(\mathrm{e}^{\frac{y}{c_S}} \left(x_e^{(t-1)} + \eta \right) \right) dy \ .$$

For $y \leq y_e^{(t)}$, $\mathrm{e}^{\frac{y}{c_S}} \left(x_e^{(t-1)} + \eta \right) \leq x_e^{(t)} + \eta$ because $x_e^{(t)} + \eta$ is exactly the value that we reach for $y = y_e^{(t)}$. So

$$\int_0^{y_e^{(t)}} \frac{1}{c_S} \left(\mathrm{e}^{\frac{y}{c_S}} \left(x_e^{(t-1)} + \eta \right) \right) dy \leq \int_0^{y_e^{(t)}} \frac{1}{c_S} \left(x_e^{(t)} + \eta \right) dy = \frac{1}{c_S} \left(x_e^{(t)} + \eta \right) y_e^{(t)} \ .$$

This way, we can bound the primal increase by

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

because $\sum_{S:e\in S} x_e^{(t)} = 1$ (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

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

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

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

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

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

□

References

- N. Buchbinder, J. Naor: The Design of Competitive Online Algorithms via a Primal-Dual Approach. Foundations and Trends in Theoretical Computer Science 3(2-3): 93-263 (2009)