Lecture 15: Online Primal-Dual Algorithms

Lecturer: *Matt Weinberg* Last updated: November 2, 2021

Today, we will learn how to use primal-dual LP setup to design online algorithms. The lecture is based on Thomas Kesselheim's notes[1].

1 Online Primal-Dual Setup

The idea of online primal-dual algorithm is to simultaneously maintain a feasible primal solution x and a feasible dual solution y. At t-th arrival, the algorithm updates both the primal and the dual while maintaining feasibility. Since weak-duality implies that a feasible dual gives a bound on the optimal primal value, we get the following lemma (which is a slight generalization because we allow a scaled dual to be feasible).

Lemma 1. *For a minimization problem, if*

(a) in every step t the primal increase is bounded by β times the dual increase, that is

$$P^{(t)} - P^{(t-1)} \leq \beta(D^{(t)} - D^{(t-1)}) \ ,$$

where $P^{(t)}$ = primal objective and $D^{(t)}$ = dual objective at time t, and

(b) $\frac{1}{\gamma}$ times the dual solution is dual-feasible,

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

Proof. To see this, observe that at all times t, we have $P^{(t)} \leq \beta D^{(t)}$ by a telescoping sum. Moreover, because $\frac{1}{\gamma}$ times the dual solution is dual feasible, we have that $OPT^{(t)} \geq D^{(t)}/\gamma$. Chaining both inequalities together yields $P^{(t)} \leq OPT^{(t)} \cdot \beta\gamma$. $\square$

2 Online Matching

As a warm-up to online primal-dual setup, we analyze the greedy algorithm for the online matching problem[2]. In online matching, edges of a graph are revealed one-by-one and the algorithm, which starts with $M = \emptyset$, has to immediately and irrevocably decide whether to include the t-th edge e into matching M. The algorithm wants to maximize the size of M.

We will prove that the greedy algorithm, which selects the next edge $e = (u, v)$ into M if both end points u, v are currently unmatched, gives a 2 competitive ratio (i.e., always maintains a matching size of at least half of the optimal offline matching size).

[1] http://tcs.cs.uni-bonn.de/lib/exe/fetch.php?media=teaching:ss20:vl-aau:lecturenotes03.pdf

[2] Pedantically, the analysis in this section for online matching should be called "online dual-fitting" instead of "online primal-dual" because the primal algorithm does not use dual variables to make its decisions, but the dual variables are only used for analysis. Since both ideas rely on weak-duality, we won't distinguish.

Let's start by writing an LP relaxation for the max-matching problem, where we denote by $N^{(t)}(u)$ the edges incident to vertex u till time t.

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E^{(t)}} x_e^{(t)} & \\ \text{subject to} \quad & \sum_{e \in N^{(t)}(u)} x_e^{(t)} \leq 1 & \text{for all vertices } u \\ & x_e^{(t)} \geq 0 & \text{for all } e \end{aligned}$$

Its dual program is given by:

$$\begin{aligned} \text{minimize} \quad & \sum_u y_u^{(t)} & \\ \text{subject to} \quad & y_u^{(t)} + y_v^{(t)} \geq 1 & \text{for all edges } (u,v) \text{ till time } t \\ & y_u^{(t)} \geq 0 & \text{for all vertices } u \end{aligned}$$

Consider the primal solution $x^{(0)}$ being all 0 in the beginning. We set $x_e^{(t)} = x_e^{(t-1)} + 1$ if e is the t-th edge and gets selected by the greedy algorithm, and otherwise set $x_e^{(t)} = x_e^{(t-1)}$. For the dual, we start with $y^{(0)} = 0$. On arrival of t-th edge $e = (u,v)$, if both u and v are currently unmatched by the greedy algorithm then we set $y_u^{(t)} = y_v^{(t)} = 1$ and for every other vertex u we set $y_u^{(t)} = y_u^{(t-1)}$. On the other hand, if on arrival of t-th edge $e = (u,v)$ either of vertex u or v is already matched, we set $y_u^{(t)} = y_u^{(t-1)}$ for all vertices. Next we show that such a setting of primal and dual variables satisfies the conditions in Lemma 1 (after making changes corresponding to maximization vs. minimization problem) with $\beta = 1/2$ and $\gamma = 1$.

First, note that the primal is always feasible because we only set $x_e = 1$ if the edge can be selected in the greedy matching. Next, to prove dual feasibility, we show the following invariant: all the vertices u that have been matched by the greedy algorithm till time t satisfy that $y_u^{(t)} = 1$. This invariant is clearly true at $t = 0$. Since we only increase $y^{(t)}$ (compared to $y^{(t-1)}$), we only need to check the invariant for the new t-th edge $e = (u,v)$. Here the invariant holds because if both u, v are currently unmatched then we select it into matching and set $y_u^{(t)} = y_v^{(t)} = 1$. Given the invariant, dual feasibility immediately follows because for any edge (u,v) that does not satisfy $y_u^{(t)} + y_v^{(t)} \geq 1$, we should have included it into the greedy matching.

Finally, note that on each edge's arrival, the increase in primal objective $\sum_{e \in E^{(t)}} x_e^{(t)} - \sum_{e \in E^{(t-1)}} x_e^{(t-1)}$ is at least half of the increase in dual objective $\sum_u y_u^{(t)} - \sum_u y_u^{(t-1)}$. This is because the dual only increases on arrival of t-th edge (u,v) when both u and v are currently unmatched in the greedy solution, and then dual increases by 2 and the primal increases by 1. Thus we have shown a competitive ratio of 2 by Lemma 1.

3 Online Fractional Set Cover

Next we will apply the online-primal dual framework to an online variant of the set cover problem. Let's first recall the offline weighted set cover problem: You are given a universe of n elements $U = \{1, \dots, n\}$ and a family of m 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.

Today, we will consider an online variant of a relaxation of this problem where we are allowed to fractionally select sets and the elements to be covered are revealed one-by-one. 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 $x^{(t)}$ to the linear inequalities. In the t-th step, the t-th element arrives and therefore we get to know the t-th coverage constraint. Possibly, the solution $x^{(t-1)}$ we had so far is infeasible now. In this case, we may only *increase* variables to get to the solution $x^{(t)}$, which is feasible again.

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 $x^{(t)}$, we will also maintain a dual solution $y^{(t)} = (y_1, y_2, \dots, y_t)$. In step t, variable y_t is added to the dual LP and we can only set its value (i.e., we do not change $y_1, \dots, y_{t-1}$). We want to eventually use Lemma 1.

3.1 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 1 holds with a small β.

So, let us figure out what we would like to do. Suppose we are in step t. That is, element t arrives and we observe a new constraint $\sum_{S \colon t \in S} x_S \geq 1$ in the primal LP. In the dual, a new variable y_t arrives. Our current solution is $x^{(t-1)}$. It fulfills all constraints except maybe the new one. If we also have $\sum_{S \colon t \in S} x_S^{(t-1)} \geq 1$, then there is nothing to do because we can keep the old solution as the new one by setting $x^{(t)} = x^{(t-1)}$, $y_t = 0$.

In the case $\sum_{S \colon t \in S} x_S^{(t-1)} < 1$, we will have to increase some primal 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_t, all other dual variables remain unchanged.

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

We are at any point in time for which still $\sum_{S:\, t \in S} x_S < 1$. We increase x_S by dx_S. To account for the increased cost, we increase y_t by dy at the same time. The dual objective function increases by dy this way. This is at least $(\sum_{S:\, t \in S} x_S) dy$ because $\sum_{S:\, t \in S} x_S < 1$. Simultaneously, the primal objective function increases by $\sum_{S:\, t \in S} c_S dx_S$. If we set $dx_S = (\frac{x_S}{c_S}) dy$ for all S for which $t \in S$, 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 a very small constant and set

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

This is a differential equation. We try a solution of the form $x_S = C_1 e^{C_2 y} + C_3$. Then we have $\frac{dx_S}{dy} = C_2(x_S - C_3)$. So comparing with (1), we get $C_3 = -\eta$ and $C_2 = \frac{1}{c_S}$. Moreover, because for $y = 0$ we have $x_S = x_S^{(t-1)}$, we get $C_1 = x_S^{(t-1)} + \eta$, This way

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

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

3.2 Algorithm

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

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

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

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

Theorem 2. *The algorithm is $O(\log m)$-competitive for online fractional set cover.*

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

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

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

and therefore

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

This lets us write the increase of x_S as follows (the final inequality follows from $e^x \geq 1 + x$).

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

This way, we can bound the primal increase by

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

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

Now, we turn to property (b). Consider a fixed $S \in \mathcal{S}$. We will verify that the dual constraint for set S is fulfilled. By our algorithm if $t \in S$ then

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

otherwise $x_S^{(t)} = x_S^{(t-1)}$ and so $c_S \ln(x_S^{(t)} + \eta) - c_S \ln(x_S^{(t-1)} + \eta) = 0$.

This lets us write the sum $\sum_{t\in S} y_t$ as

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

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

$$\sum_{t:t\in S} y_t \leq c_S \ln \left(\frac{1+\eta}{\eta} \right) = c_S \ln(m+1) = \gamma c_S \enspace . \qquad \square$$

It is possible to extend Theorem 2 to an $O(\log n \cdot \log m)$ competitive algorithm for the online set cover problem where the algorithm has to select sets integrally. The idea is to do randomized rounding, try this as an exercise or see [1].

References

[1] 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)