CS880: Approximation and Online Algorithms	**Scribe:** Taylor Kemp
Lecture 16: Online Set Cover	**Date:** 10-25-2019

16.1 Online Set Cover

In the online version of set cover we have the following problem. We are given m sets $\{S_i\}$ with associated costs c_i. During each step a new element j will arrive, along with a list of sets it belongs to. The algorithm is then tasked with maintaining a low cost collection of sets that cover all elements seen so far.

Definition 16.1.1 *Let σ be defined as the sequence of elements or requests. We define the competitive ratio as*

$$c.r. = \max_{\sigma} \frac{\mathbb{E}[Alg(\sigma)]}{Opt(\sigma)} \tag{16.1.1}$$

Let us first consider developing a linear program for the offline version of set cover.

Primal: Covering

$$\begin{aligned} min \sum_i c_i x_i \\ s.t. \sum_{i:S_i \ni j} x_i \geq 1 \forall j \\ x_i \geq 0 \end{aligned}$$

Dual: Packing

$$\begin{aligned} max \sum_j y_j \\ s.t. \sum_{j:j \in S_i} y_j \leq c_i \forall i \\ y_j \geq 0 \end{aligned}$$

In the primal we are trying to minimize the number of sets we include $\sum_i c_i x_i$. We also consider that all elements j are covered $\sum_{i:S_i \ni j} x_i \geq 1$. In the dual we are considering a packing problem where we are trying to maximize the number of elements y_j where we require that each set can contain at most c_i elements $\sum_{j:j \in S_i} y_j \leq c_i$. We will use these to motivate the following primal-dual algorithm.

Algorithm 1 Online Primal-Dual Algorithm

1: When element j arrives
2: **if** j not covered **then**
3:     raise the value of y_j gradually
4:     raise the value of x_i where $S_i \ni j$ as a function $x_i = f(\sum_{j \in S_i} y_j)$

There are then a couple properties of the function f that must be satisfied. First $f(0) = 0$ and $f(c_i) = 1$. We choose these values because if the set S_i contains no elements observed so far. We do not need to include the set in our solution eg $f(0) = 0$. Additionally, looking at the dual, if a set S_i has been packed full eg $\sum_{j:j\in S_i} y_j = c_i$, then we want to include the set S_i in our solution, hence, $f(c_i) = 1$.

Claim 16.1.2 *The following hold true*

- *Primal Feasibility*
- *Dual Feasibility*
- *At any point of time, ∂ Primal Cos $\leq \alpha \cdot \partial$ Dual Cost*

The first follows from the fact that at each step of the algorithm, for every element j, at least one x_i with $j \in S_i$ is at least one. This is due to $x_i = f(\sum_{j\in S_i} y_j)$ and $f(c_i) = 1$. The second follows from the fact that for every element j we observe, we do not raise the value y_j beyond tightness in the constraint $\sum_{j:j\in S_i} y_j$. Finally, we consider the last claim.

$$\frac{\partial}{\partial y_j}\text{Primal Cost} = \sum_{i:S_i \ni j} c_i \cdot \frac{\partial x_i}{\partial y_j} \leq \alpha \tag{16.1.2}$$

In order to ensure that $c_i \cdot \frac{\partial x_i}{\partial y_j} \leq \alpha \cdot x_i$ we define f in the following manner

$$f(y_i) = \frac{1}{\lambda} e^{\frac{\alpha y_i}{c_i}} + \beta \tag{16.1.3}$$

Given that we want $f(0) = 0$ and $f(c_i) = 1$. We can derive the following values for α, β.

$$0 = f(0) = \frac{1}{\lambda} e^0 + \beta \implies \beta = -\frac{1}{\lambda} \tag{16.1.4}$$

$$1 = f(c_i) = \frac{1}{\lambda} e^{\alpha} - \frac{1}{\lambda} \implies \alpha = log(\lambda + 1) \tag{16.1.5}$$

From this we now have the following equation for x_i. Where $y_i = \sum_{j:j\in S_i} y_j$

$$f(y_i) = \frac{1}{\lambda}(e^{\frac{log(\lambda+1) y_i}{c_i}} - 1) \tag{16.1.6}$$

Considering the derivative we can obtain the following.

$$\frac{\partial x_i}{\partial y_j} = \frac{1}{\lambda} \frac{log(\lambda+1)}{c_i} e^{\frac{log(\lambda+1)\sum_{j\in S_i} y_j}{c_i}} \tag{16.1.7}$$

$$= \frac{log(\lambda+1)}{c_i}(x_i + \frac{1}{\lambda}) \tag{16.1.8}$$

Using this we will now determine the rate of change of the primal cost.

$$\frac{\partial \text{Primal Cost}}{\partial y_j} = \sum_{i:S_i \ni j} c_i \frac{\partial x_i}{\partial y_j} \tag{16.1.9}$$

$$= log(\lambda + 1) \sum_{i:S_i \ni j} (x_i + \frac{1}{\lambda}) \tag{16.1.10}$$

$$= log(\lambda + 1) \cdot (1 + \frac{|\{i : S_i \ni j\}|}{\lambda}) \tag{16.1.11}$$

If we consider $d = max_j |\{i : S_i \ni j\}|$ as the maximum frequency. Then we have the following relationship.

$$\frac{\partial \text{Primal Cost}}{\partial y_j} \leq log(\lambda + 1) \cdot (1 + \frac{d}{\lambda}) \tag{16.1.12}$$

Using this, we have that setting $\lambda = d$ gives us a competitive ratio of $2log(d + 1)$.

16.1.1 Secondary Approach

We now consider an alternative approach

Claim 16.1.3 *The following holds.*

- *Primal is feasible*
- *Complimentary Slackness* $x_i > 0 \implies \sum_{j \in S} y_j \geq c_i$ *e.g.* $f(y) = 0 \forall y < c_i$

Let us consider choosing $f(y) = 0$ if $y < c_i$ and $f(y) = \frac{1}{d} e^{\frac{y}{c_i} - 1}$ if $y \geq c_i$ Then primal feasibility follows from construction, and complimentary slackness follows from the fact that if $\sum_{j \in S_i} y_j \geq c_i$, $x_i > 0$ and if $\sum_{j \in S_i} y_j < c_i$, then $x_i = 0$.

Claim 16.1.4 *Dual constraints are violated by a factor of log(d)+1 at most.*

Let us consider $x_i = 1$. Then we have the following relationship

$$1 = f(y) = \frac{1}{d} e^{\frac{y}{c_i} - 1} \implies \tag{16.1.13}$$

$$y = c_i(log(d) + 1) \tag{16.1.14}$$

Hence we have that the dual constraints are violated by at most a factor of $log(d) + 1$.

Claim 16.1.5 *Primal Cost* $\leq$ *2* $\cdot$*DualCost*

Let us consider $\hat{x}_i = \min(x_i, \frac{1}{d}), \forall i$. From this we can then obtain the following

$$\sum_i c_i \hat{x}_i \leq \sum_i \hat{x}_i (\sum_{j \in S_i} y_j) \tag{16.1.15}$$

$$= \sum_j y_j (\sum_{i:j \in S_i} \hat{x}_i) \tag{16.1.16}$$

$$\leq \sum_j y_j d \cdot \frac{1}{d} = \text{Dual Cost} \tag{16.1.17}$$

Finally, let us consider the rate of change of $x_i - \hat{x}_i$.

$$\frac{\partial}{\partial y_j}(x_i - \hat{x}_i) = \frac{1}{c_i}\frac{1}{d}e^{(\sum_{j:j \ni S_i} \frac{y_j}{c_i})-1} = \frac{x_i}{c_i} \tag{16.1.18}$$

From this we have that the total rate of increase in the primal cost is given by

$$(x_i - \hat{x}_i) = \sum_{i:S_i \ni j} c_i \frac{x_i}{c_i} \leq 1 \tag{16.1.19}$$

This implies a $2(log(d)+1)-$approximation

16.1.2 Online Rounding

We now consider implementing online rounding to provide integral solutions for x_i. We will first consider $r_i =$ min of $log(n)$ draws uniform on $[0,1]$. We will then include S_i in the solution as soon as $x_i \geq r_i$. Then at step t if an element j arrives and is uncovered we will pick the cheapest i containing j. The expected cost of such a solution is given by

$$\mathbb{E}[\text{cost of solution}] \leq 2log(n) \cdot LP + n \cdot \frac{1}{n^2} \cdot Opt \tag{16.1.20}$$

This implies an $O(log(n)log(d))$ approximation. If we consider the probability that an element j is uncovered we have $pr[\text{j uncovered}] = \prod_{i:S_i \ni j}(1-x_i)^{log(n)} \leq e^{-log(n)} = \frac{1}{n}$

16.2 Online Routing

Let us consider the following graph $G = (V, E)$ with edge capacities c_e and source sink pairs (s_i, t_i). Our goal is then to find a path P_i from s_i to t_i. We define the load on an edge e as $|\{i : P_i \ni e\}|$. We then want to minimize the maximum congestion $max_e(\frac{l_e}{c_e})$. One such idea we can propose is to greedily choose each route in order to minimize the congestion in the current graph. This idea, however, has an unbounded competitive ratio. For example consider a circular graph oriented with nodes $1, 2, \cdots n$ around the circle. Consider at each iteration we observe two nodes $i, i+1$ with load two. In this case we will place a load on every edge in the graph. But after we have observed every choice of edge between i and $i+1$, we will have a load of n on each edge instead of 2. Hence, we have that the greedy solution has an unbounded competitive ratio.