COL758: Advanced Algorithms **Spring 2019**

Lecture 7: January 24

Lecturer: Naveen Garg *Scribes: Sourabh Basutkar*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

7.1 Covering LP

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \\ & x \geq 0 \end{aligned}$$

An example of this type of LP is the **Set Cover Problem**.

7.2 Set Cover Problem

The set cover problem is defined as follows

Input: A set of elements $\mathbb{U} = \{e_1, e_2, \ldots, e_n\}$. A set $\mathbb{S} = \{S_1, S_2, \ldots, S_m\}$ where $\forall\ i\ S_i \subseteq U$.
Output: A set $\mathbb{O} \subseteq S$ of minimum size such that $\bigcup_{S_i \in \mathbb{O}} S_i = \mathbb{U}$.

Example:

$$\begin{aligned} \mathbb{U} &= \{e_1, e_2, e_3, e_4, e_5\} \\ S_1 &= \{e_1, e_3, e_5\} \\ S_2 &= \{e_2, e_4, e_5\} \\ S_3 &= \{e_3, e_4\} \\ S_4 &= \{e_1, e_5\} \\ S_5 &= \{e_1, e_2, e_4\} \end{aligned}$$

The Optimal Set Covers in the above instance are $\{S_1, S_2\}$,$\{S_1, S_5\}$

A variant of the above problem is the Min Cost Set Cover where every set has a cost associated with it and the problem is to find the set cover with minimum total cost.

Integer Program for Set Cover

- Firstly we have a variable $x_j \in \{0,1\}$ for every set S_j which is set to 1 if the set is picked and 0 otherwise.
- Since the problem constraints that a possible set cover must cover all elements it implies that for every element at least one of the set it belongs to must be selected.
- The objective is to minimize the number of sets covered.

The following Integer program accounts for all the above points.

$$\begin{aligned} \text{minimize} \quad & \sum_j x_j \\ \text{subject to} \quad & \sum_{j: e_i \in S_j} x_j \geq 1 \quad \forall e_i \in U \\ & x_j \in \{0,1\} \quad \forall S_j \end{aligned}$$

LP for Set Cover

Below is the LP for Set Cover after relaxing the Integer Program.

$$\begin{aligned} \text{minimize} \quad & \sum_j x_j \\ \text{subject to} \quad & \sum_{j: e_i \in S_j} x_j \geq 1 \quad \forall e_i \in U \\ & x_j \geq 0 \quad \forall S_j \end{aligned}$$

Note that the above LP doesn't require the constraint $x_j \leq 1$ as for any feasible solution with a variable $x_j > 1$ we can set $x_j = 1$ and it still is a feasible solution with a better objective function since the constraints only demand $\sum_{j: e_i \in S_j} x_j \geq 1$.

Similarly the LP for the Min Cost Set Cover is

$$\begin{aligned} \text{minimize} \quad & \sum_j c_j x_j \\ \text{subject to} \quad & \sum_{j: e_i \in S_j} x_j \geq 1 \quad \forall e_i \in U \\ & x_j \geq 0 \quad \forall S_j \end{aligned}$$

Dual for Set Cover

Below is the Dual for Set Cover LP.

$$\begin{aligned} \max \quad & \sum_i y_i \\ \text{subject to} \quad & \sum_{i:e_i \in S_j} y_i \leq 1 \quad \forall S_j \\ & y_i \geq 0 \quad \forall e_i \in U \end{aligned}$$

and the Dual for the Min Cost Set Cover

$$\begin{aligned} \max \quad & \sum_i y_i \\ \text{subject to} \quad & \sum_{i:e_i \in S_j} y_i \leq c_j \quad \forall S_j \\ & y_i \geq 0 \quad \forall e_i \in U \end{aligned}$$

An interpretation for the above Dual is that all the sets S_j have a volume c_j and we are assigning each element some volume and the problem is to find an assignment for which the total volume of all the elements is maximised constrained by the rule that for no set the total volume of elements in it is greater than the volume of the set.

7.3 Greedy Algorithm

Below is a greedy algorithm for the Min Cost Set Cover Problem

Algorithm 1: Greedy Set Cover Algorithm

Data: U,$\{S_1, S_2, \ldots, S_m\}$
Result: A set cover S

$S \leftarrow \phi$;
$X \leftarrow \phi$;
while $X \neq U$ **do**
  $j = \underset{j}{\text{argmin}} \frac{c_j}{|S_j - X|}$;
  $S \leftarrow S \cup \{S_j\}$;
  $X \leftarrow S \cup S_j$;
end

The above algorithm basically picks the set with the least cost to number of uncovered elements ratio till a set cover is achieved.[1]

[1] The same algorithm and analysis follows for trivial Set Cover with no costs by setting $c_j = 1$

Analysis

<u>**Claim**</u>: The Greedy Set Cover Algorithm is an $\mathbf{O}(\log\ \mathbf{n})$ approximation.

Proof: The following proof will use the above algorithm to create a feasible solution to the Primal problem and the Dual problem and show that they are only $\mathbf{O}(\log\ \mathbf{n})$ factor apart.

Say the algorithm picks the sets $S_{j_1}, S_{j_2}, \ldots, S_{j_k}$ in the same order. We set the Primal and Dual variables as follows.

$$\begin{array}{llll}
\text{Init:} & x_j = 0 & y_i = 0 & \forall\ i, j \\
\text{Step 1:} & x_{j_1} = 1 & y_i = \dfrac{c_{j_1}}{|\overline{S_{j_1}}|} & \forall\ i : e_i \in \overline{S_{j_1}} \\
\text{Step 2:} & x_{j_2} = 1 & y_i = \dfrac{c_{j_2}}{|\overline{S_{j_2}}|} & \forall\ i : e_i \in \overline{S_{j_2}} \\
\vdots & & & \\
\text{Step k:} & x_{j_k} = 1 & y_i = \dfrac{c_{j_k}}{|\overline{S_{j_k}}|} & \forall\ i : e_i \in \overline{S_{j_k}}
\end{array}$$

Where $\overline{S_{j_p}} = S_{j_p} \setminus \{\bigcup_{q=1}^{p-1} S_{j_q}\}$

The increment in the primal objective function at p^{th} step is c_{J_p}. And the increment in dual objective function is $\sum_{i:e_i \in \overline{S_{j_p}}} y_i$. Since $\underset{e_i \in \overline{S_{j_p}}}{\forall}\ y_i = \frac{c_{j_p}}{|\overline{S_{j_p}}|}$ the increment in dual objective solution evaluates to c_{J_p}.

$$\sum_j c_j x_j = \sum_i y_i$$

The Primal variables form a feasible solution by termination condition of the algorithm and the definition of the LP.

But for the Dual variables the constraints might be violated. Consider for a set S_p the elements $e_1, e_2, \ldots, e_{|S_p|}$ which were covered in the same order. When e_1 was covered by some set S_{q_1} from the condition of greedy algorithm we have that

$$y_1 = \frac{c_{q_1}}{|\overline{S_{q_1}}|} \leq \frac{c_p}{|S_p|}$$

. And similarly for e_2 we have that

$$y_2 = \frac{c_{q_2}}{|\overline{S_{q_2}}|} \leq \frac{c_p}{|S_p| - 1}$$

and so on.

$$y_{|S_p|} = \frac{c_{q_{|S_p|}}}{|\overline{S_{q_{|S_p|}}}|} \leq \frac{c_p}{1}$$

Note that q_2 can be the same as q_1 but the property still holds as $\frac{c_p}{|S_p|} < \frac{c_p}{|S_p|-1}$.

$$\begin{aligned}\sum_{i:e_i \in S_p} y_i &\leq \frac{c_p}{|S_p|} + \frac{c_p}{|S_p|-1} + \dots \frac{c_p}{1} \\ &\leq c_p(1 + \int_1^{|S_p|} \frac{1}{x}\ dx) \\ &\leq c_p(1 + \ln(|S_p|)) \\ &\leq c_p(1 + \ln(n))\end{aligned}$$

Where n is the total number of elements.
Hence by scaling all the dual variables by $(1 + \ln(n))$ we can create a feasible dual solution.

$$\overline{y_i} = \frac{y_i}{(1 + \ln(n))}$$

.

The objective function value of this new dual solution is

$$\sum_i \overline{y_i} = \frac{1}{(1 + \ln(n))} \sum_i y_i = \frac{1}{(1 + \ln(n))} \sum_j c_j x_j$$

Since the optimal LP solution lies between feasible primal solutions and feasible dual solutions

$$\sum_i \overline{y_i} \leq OPT \leq \sum_j c_j x_j$$

And from the fact that the Optimal Integral Solution lies between the optimal LP solution and Greedy Integral solution we have that

$$\begin{aligned}\sum_i \overline{y_i} &\leq OPT \leq OPT_{Integral} \leq \sum_j c_j x_j \\ \frac{1}{(1 + \ln(n))} \sum_j c_j x_j &\leq OPT \leq OPT_{Integral} \leq \sum_j c_j x_j\end{aligned}$$

Finally, we have that

$$\sum_j c_j x_j \leq (1 + \ln(n)) OPT_{Integral}$$

■