1 Weighted Set Cover via LP Dual Fitting

Weighted vertex cover is a special case of the weighted set cover problem. We have previously seen an approximation algorithm for weighted set cover, where the approximation ratio involved the function

$$H(m) = \sum_{j=1}^{m} \frac{1}{j} \leq 1 + \ln(m). \quad (1)$$

We have run into the function $H$ before, in the analysis of random treaps. In this lecture we show how this result can be obtained by an analysis technique known as LP dual fitting.

Recall that in the weighted set cover problem, we are given a set $U$ of $n$ elements along with a set $S$ of subsets of $U$ with nonnegative weights $w : S \rightarrow \mathbb{R}_+$ such that $\bigcup S = U$; that is, the union of all the sets in $S$ covers $U$. The goal is to choose a subcollection $J \subseteq S$ of minimum total weight $\sum_{S \in J} w_S$ such that $\bigcup J = U$. The decision version of this problem is NP-complete.

Recall that our greedy approximation algorithm chooses sets according to a “minimum weight per new element covered” heuristic. The algorithm constructs $J$ inductively according to this heuristic. The variable $T$ keeps track of the set of elements not yet covered by $\bigcup J$.

Algorithm 1 Greedy algorithm for set cover

1: Initialize $J \leftarrow \emptyset$ and $T \leftarrow U$
2: while $T \neq \emptyset$ do
3: \hspace{1em} $S \leftarrow \arg\min_{S \in S} \{w(S)/|T \cap S| \mid S \in S, \ T \cap S \neq \emptyset\}$
4: \hspace{1em} $J \leftarrow J \cup \{S\}$
5: \hspace{1em} $T \leftarrow T \setminus S$
6: end while
7: return $J$

Line 3 greedily selects the set minimizing the added weight per new element covered. This quantity might be called the cost-effectiveness of the set.

It is clear that the algorithm can be implemented in polynomial time and produces a valid set cover. We wish to show that it achieves an approximation ratio of $\alpha$, where

$$\alpha = H(\max_{S \in S} |S|),$$
where the function $H$ is defined in (1).

To analyze the approximation ratio, we will use the LP relaxation of set cover and its dual.

\[
\begin{align*}
\text{minimize} & \quad \sum_{S \in \mathcal{S}} w_S x_S \\
\text{subject to} & \quad \sum_{j \in S} x_S \geq 1 \quad \text{for } j \in U \quad (2) \\
& \quad x_S \geq 0 \quad \text{for } S \in \mathcal{S} \\
\end{align*}
\]

\[
\begin{align*}
\text{maximize} & \quad \sum_{j \in U} y_j \\
\text{subject to} & \quad \sum_{j \in S} y_j \leq w_S \quad \text{for } S \in \mathcal{S} \quad (3) \\
& \quad y_j \geq 0 \quad \text{for } j \in U. \\
\end{align*}
\]

It will be helpful to add some extra lines to the program that do not influence the choice of sets to put into $J$, but merely record some extra data relevant to the analysis. Specifically, we compute a vector $z$ indexed by elements of $U$. The vector $z$ is not a feasible solution of the dual LP, but at the end we will scale it down by a factor of $\alpha$ to obtain $y = z/\alpha$ that is feasible for the dual LP. The scale factor $\alpha$ will be an upper bound on the approximation ratio. This method is sometimes called *dual fitting*.

**Algorithm 2** Greedy algorithm for set cover

1: Initialize $J \leftarrow \emptyset$ and $T \leftarrow U$
2: while $T \neq \emptyset$ do
3: $S \leftarrow \arg \min_S \left\{ w_S/|T \cap S| \mid S \in \mathcal{S}, \ T \cap S \neq \emptyset \right\}$
4: $J \leftarrow J \cup \{S\}$
5: for $j \in T \cap S$ do
6: $z_j \leftarrow w_S/|T \cap S|$
7: end for
8: $T \leftarrow T \setminus S$
9: end while
10: $y \leftarrow z/\alpha$
11: return $J$

The following loop invariant is easily shown to hold initially and to be preserved by the body of the while loop:

\[
\sum_{j \in U} z_j = \sum_{S \in J} w_S.
\]

Also, note that each $z_j$ is assigned exactly once in line 6, at the time when $j$ becomes covered. The weight of the set $S$ chosen in line 3 is apportioned equally among all the new points covered, and the value assigned to $z_j$ is the portion borne by $j$. 
We will show below (Lemma 1) that the vector $y$ created in line 10 is feasible for the dual LP (3). From this it follows that the approximation ratio is bounded above by $\alpha$:

$$\sum_{S \in I} w_S = \sum_{j \in U} z_j = \alpha \sum_{j \in U} y_j \leq \alpha \sum_{S \in \text{OPT}} w_S,$$

where the last line follows from weak duality.

**Lemma 1.** The vector $y$ computed in line 10 of Algorithm 2 is feasible for the dual linear program (3).

*Proof.* Clearly $y_j \geq 0$ for all $j$, so we only need to show that $\sum_{j \in S} y_j \leq w_S$ for every set $S$; equivalently,

$$\sum_{j \in S} z_j \leq \alpha w_S$$

for every set $S$. Let $m = |S|$ and denote the elements of $S$ by $s_0, s_1, \ldots, s_{m-1}$, where the numbering corresponds to the order in which nonzero values were assigned to the variables $z_{s_j}$ in line 6. This is also the order in which the elements were first covered by a set chosen in line 3.

At the time $s_0$ was covered and the value $z_{s_0}$ assigned, all of the elements of $S$ still belonged to $T$. At that time, the cost-effectiveness of $S$ (weight of $S$ divided by number of new elements that would be covered by choosing $S$) was judged to be $w_S/m$. The algorithm chose a set with the same or better cost-effectiveness, and $z_{s_0}$ was set equal to the cost-effectiveness of the chosen set; thus

$$z_{s_0} \leq \frac{w_S}{m}. \quad (4)$$

In general, for any $k < m$, at the time $s_k$ was covered and the value $z_{s_k}$ assigned, all of the elements $s_k, s_{k+1}, \ldots, s_{m-1}$ still belonged to $T$. At that time, the cost-effectiveness of $S$ was judged to be at most $w_S/(m - k)$. The algorithm chose a set with the same or better cost-effectiveness, and $z_{s_k}$ was set to the cost-effectiveness of the chosen set; thus

$$z_{s_k} \leq \frac{w_S}{m - k}. \quad (5)$$

Summing the bounds (5) for $k = 0, \ldots, m - 1$, we see that

$$\sum_{j \in S} z_j \leq \left(\frac{1}{m} + \frac{1}{m - 1} + \cdots + \frac{1}{2} + 1\right) w_S = H(m)w_S \leq \alpha w_S,$$

as desired. \qed