Application of linear programming to set cover and related problems

Daniel Gulotta

February 14, 2006

1 Review from last time

1.1 Set cover problem

U is the universal set, $\mathcal{S}$ is a collection of subsets of U, and $c : \mathcal{S} \to \mathbb{N}$ is a cost function. The goal is to find a collection $S_1, S_2, ...S_k$ of elements of $\mathcal{S}$ such that $S_1 \cup S_2 \cup ... \cup S_k = U$ with minimal total cost.

1.2 Greedy algorithm for set cover

At step n, choose the set S_n that minimizes $c(S_n)/|S_n \setminus \cup_{k=1}^{n-1} S_k$. Halt if $\cup_{k=1}^{n} S_n = U$.

1.3 Linear program for set cover problem

Minimize $\sum_S c(S)x_S$ subject to $\sum_{S \ni e} x_S \geq 1$ for each i.

1.4 Dual linear program for set cover problem

Maximize $\sum_{e \in U} y_e$ subject to $sum_{e \in S} y_e \leq c(S)$ for each S.

2 Proof of greedy algorithm performance using linear programming

Last time it was shown that the solution chosen by the greedy algorithm is at most H_n times the optimal solution. This can also be proved using linear programming.

The y_e's in the dual program represent the cost of each element of the set. The greedy algorithm also assigned costs to the elements of the set. Usually, the costs found by the greedy algorithm do not satisfy all of the inequalities of the dual program. However, if all of the costs are divided by H_n, then the inequalities of the dual program are satisfied.

Proof. For any set $S \in \mathcal{S}$, consider the e_i, the ith element of S to be covered. Before this element is covered, at least $|S| - i + 1$ elements of S are uncovered, so S could be added for a cost of $\frac{c(S)}{|S|-i+1}$ per element. The cost per element of the set that is actually chosen can be no larger for this. After dividing by H_n, the dual linear programming solution satisfies

$$y_{e_i} \leq \frac{1}{H_n} \frac{c(S)}{|S| - i + 1} \tag{1}$$

The sum of the costs of all elements of S is at most

$$\sum_{i=1}^{|S|} y_{e_i} \leq \frac{c(S)}{H_n} \sum_{i=1}^{|S|} \frac{1}{|S| - i + 1} = \frac{c(S) H_{|S|}}{H_n} \leq c(S) \tag{2}$$

Therefore the solution $y_e = \frac{\text{price}(e)}{H_n}$ satisfies all of the inequalities of the dual linear programming problem. □

Note that n can be replaced by $\max_{S \in \mathcal{S}} |S|$ in the above bound. This means that if the size of the subsets is bounded, then the greedy solution is within a constant factor of the optimal one.

Since the difference between the optimal solution and the linear programming bound is always less than the difference between the greedy solution and the linear programming bound, the linear programming bound is at worst a logarithmic factor smaller than the optimal solution. It turns out that there are cases when the linear programming bound is off by a logarithmic factor.

Let k be an integer, and let $n = 2^k - 1$. Let U be projective $k-1$-space over $\mathbb{F}_2$, and let $\mathcal{S}$ consist of the complements all $k-2$-dimensional hyperplanes. Give each set a cost of one. Since each set has $2^{k-1} = \frac{n+1}{2}$ elements, the solution to the dual linear program is $x_i = \frac{2}{n+1}$. This gives a total cost of $\frac{2n}{n+1}$. However, any intersection of $k-1$ or fewer hyperplanes is nonempty, so $k = \log_2(n+1)$ complements of hyperplanes are needed to cover U. Therefore there exist instances of the set cover problem for which linear programming underetimates the solution by a factor of $\frac{n+1}{2n} \log_2(n+1) > \log_4 n \approx .72 H_n$.

3 Constrained set multicover

3.1 Setup

The set multicover problem is similar to the set cover problem, but each element i must be covered a specified number of times r_i. In the constrained set multicover problem, each set can be used only once. The linear program for

constrained set multicover is

$$\begin{aligned} \text{Minimize} \quad & \sum_{S \in \mathcal{S}} c(S) x_S && (3) \\ x_S & \geq 0 && (4) \\ \sum_{S \ni e} x_S & \geq r_e && (5) \\ -x_S & \geq -1 && (6) \end{aligned}$$

Because of the requirement that each set be used only once, the linear program now has negative coefficients.

These additional constraints also lead to new variables in the dual linear program.

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in U} r_i y_i - \sum_{S \in \mathcal{S}} z_S && (7) \\ \left(\sum_{e \in S} y_e \right) - z_S & \leq c(S) && (8) \\ e_i & \geq 0 && (9) \\ z_S & \geq 0 && (10) \end{aligned}$$

Essentially, the z_S's represent opportunity costs for choosing a particular set (and thus preventing it from being chosen again).

3.2 Greedy algorithm

Again, the greedy algorithm can be used to find an approximate solution to this problem. At each step, the set that minimizes cost divided by the number elements that need to be covered is chosen.

In order to measure the performance of the greedy algorithm, we assign a price to each element of U and each of the r_e instances that that element is chosen. For each $e \in U$, define α_e to be the price of e the last time it is chosen, and for each $S \in \mathcal{S}$, that was chosen define β_S to be $\left(\sum_{e \text{ covered by } S} \alpha_e \right) - c(S)$. β_S can be thought of as the discount received by S. If S was not chosen, then set $\beta_S = 0$. Then the total cost of the covering found by the greedy algorithm is $\sum_{i \in U} \alpha_i - \sum_{S \in \mathcal{S}} \beta_S$.

As in the ordinary set cover problem, dividing the prices found by the greedy algorithm by H_n gives a feasible solution to the dual linear programming problem.

Proof. Choose $S \in \mathcal{S}$. Let e_i be the ith element of S to be covered completely. If $\mathcal{S}$ is never chosen, then $c(S)/(|S|-i+1) \geq \alpha_{e_i}$. So $\sum_{e_i \in S} y_{e_i} - z_S = \frac{1}{H_n} \sum_i \alpha_i \geq c(S)$. If $\mathcal{S}$ is chosen, then the sum of the α's of the elements that still needed to be covered minus β_S is precisely $c(S)$. The sum of the α's of the remaining elements is at most $(H_n - 1)c(S)$. So the linear programming constraint is still satisfied. □