CMSC 858F: Network Design Foundation
Fall 2011
Submodular Cover, Submodular Tree Coverage and Budgeted Maximum Coverage

Instructor: Mohammad T. Hajiaghayi
Scribe: Rajesh Chitnis

September 7, 2011

1 Overview

In this lecture we give approximation algorithms for the Set Cover, Submodular Set Cover, Submodular Tree Coverage and the Budgeted Maximum Coverage problems.

2 Approximation algorithm for Set Cover via LP-Rounding

In this section we will see an f-approximation algorithm for the Set Cover problem where f is the maximum number of times any element of the universe appears in the sets.

Set Cover
Input : A universe $\mathbb{U} = \{u_1, u_2, \ldots, u_m\}$, a collection $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets of $\mathbb{U}$ and a cost function $c : \mathbb{S} \rightarrow \mathbb{R}^+ \cup \{0\}$
Goal : To find a minimum cost $\mathbb{S}' \subseteq \mathbb{S}$ such that $\mathbb{U} = \bigcup_{S \in \mathbb{S}'} S$

Consider the following integer program which solves the Set Cover problem.

$$\min \quad \sum_{j=1}^{n} c(S_j)x_j$$

subject to

$$\forall\ i \in [m], \quad \sum_{u_i \in S_j} x_j \geq 1$$

$$\forall\ j \in [n], \quad x_j \in \{0, 1\}$$

We relax the above integer program to get a linear program (which we can solve in polynomial time) by relaxing the second constraint to $x_j \geq 0 \quad \forall j \in [n]$.

We solve the above LP optimally in polynomial time. Let the solution be $\{x'_1, x'_2, \ldots, x'_n\}$. Consider the following rounding procedure for all $1 \leq j \leq n$:

- If $x'_j \geq \frac{1}{f}$ then set $y_j = 1$.
- Else set $y_j = 0$.

We claim that $\{y_1, y_2, \ldots, y_n\}$ is a feasible solution for the Set Cover instance. Since f is the maximum frequency every $u_i \in \mathbb{U}$ belongs to at most f of the sets in $\mathbb{S}$. For every $u_i \in \mathbb{U}$, as $\sum_{u_i \in S_j} x_j \geq 1$ at least one of the x'_js should be at least $\frac{1}{f}$ and so the corresponding y_j will be 1.

Also $y_j \leq f.x'_j \quad \forall j \in [n]$. Solution of IP is greater equal the (fractional) solution of the LP and hence we get an f-approximation algorithm. The same LP can also be used to give a $\ln(n)$-approximation algorithm.

CPLEX is a software package which is often used to solve linear programs. The language AMPL is a modeling language used to support lots of LP solvers including CPLEX.

3 Some Definitions

Definition 1 *Let* U *be a finite set and let* $f : 2^U \rightarrow \mathbb{Z}^+ \cup \{0\}$ *be a non-negative integer valued function*

- f *is* non-decreasing *if* $f(S) \leq f(T)$ *for all* $S \subseteq T \subseteq U$.
- f *is* submodular *if* $f(S)+f(T) \geq f(S \cup T)+f(S \cap T)$ *for all* $S, T \subseteq U$. *Another equivalent definition which is more useful in practice is the following :* f *is submodular if* $f(A \cup \{a\}) - f(A) \leq f(B \cup \{a\}) - f(B)$ *for all* $B \subseteq A \subseteq U$ *and* $a \in U \setminus A$.
- f *is* subadditive *if* $f(S) + f(T) \geq f(S \cup T)$.
- f *is* polymatroid *if it is non-decreasing, submodular, integer valued and* $f(\emptyset) = 0$.

4 Submodular Set Cover (generalization of Set Cover)

In the Submodular Set Cover problem we are given the following :

- A universe $U = \{u_1, u_2, \ldots, u_n\}$
- A non-decreasing submodular function $f : 2^U \rightarrow \mathbb{Z}^+ \cup \{0\}$
- A cost c_i for each element $u_i \in U$

The cost of a set is the sum of the costs of its elements. Also we say that a set $S \subseteq U$ is *spanning* if $f(S) = f(U)$. The SUBMODULAR SET COVER asks us to find a spanning set S^* of minimum cost.

4.1 Reduction from Set Cover to Submodular Set Cover

In this subsection we sketch a reduction from SET COVER to SUBMODULAR SET COVER. Consider an instance of the SET COVER problem. It is given by a universe $\mathbb{U} = \{u_1, u_2, \ldots, u_m\}$, a collection $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets of $\mathbb{U}$ and a cost function $c : \mathbb{S} \to \mathbb{R}^+ \cup \{0\}$. We now build an instance of SUBMODULAR SET COVER. The sets in the set cover problem become the elements for the submodular set cover instance. So we have

- $U = \{S_1, S_2, \ldots, S_n\}$ and
- c_i is the cost of S_i which is given by $c(S_i)$
- $f : 2^U \to \mathbb{Z}^+ \cup \{0\}$ is given as follows : Let $S \subseteq U$. So $S = \{S_{i_1}, S_{i_2}, \ldots, S_{i_j}\}$. We define $f(S) = |\bigcup_{k=1}^{j} S_{i_k}|$

It remains to check that the function f defined is submodular and both the instances are equivalent.

4.2 Approximation Algorithm for Submodular Set Cover

Theorem 1 *(Wolsey 1982) [1]*
If f *is a* polymatroid *function then there is an* $H\Big(\max_{j \in U} f(\{j\})\Big) = O\Big(\log(\max_{j \in U} f(\{j\}))\Big)$ *factor approximation algorithm for the* SUBMODULAR SET COVER *problem.*

The algorithm is greedy and is as follows :

1. Start with $C = \emptyset$
2. Add gradually an element u_j which minimizes $\dfrac{c_j}{f(C \cup \{j\}) - f(C)}$.

The analysis is somewhat similar to the analysis done in the previous lecture for the $\log(n)$ greedy approximation algorithm for SET COVER.

5 Other variants of Set Cover

Consider the following generalization of Set Cover :

2-SET COVER

Input : A universe $\mathbb{U} = \{u_1, u_2, \ldots, u_m\}$, a collection $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets of $\mathbb{U}$ and a cost function $c : \mathbb{S} \to \mathbb{R}^+ \cup \{0\}$
Goal : To find a minimum cost $\mathbb{S}' \subseteq \mathbb{S}$ such that each element of $\mathbb{U}$ occurs at least twice in $\bigcup_{S \in \mathbb{S}'} S$.

Can we try to approximate this using Theorem 1 ? We need to define an appropriate function f. One natural definition which comes to mind is the following : If $S = \{S_1, S_2, \dots, S_j\}$ then $f(S)$ is the number of elements which belong to at least two of the $S_j's$. But this f is not submodular (check). Also f applied to any singleton element will be 0 and hence Theorem 1 will not be useful. Consider the following function :

$$f(\{S_1, S_2, \dots, S_j\}) = \sum_{i=1}^{m} \beta(u_i)$$

where $\beta(u_i)$ is defined as follows :

- $\beta(u_i) = 0$ if u_i does not belong to any of $S_1, S_2, \dots, S_j$.
- $\beta(u_i) = 1$ if u_i belongs to exactly one of $S_1, S_2, \dots, S_j$.
- $\beta(u_i) = 2$ if u_i belongs to at least two of $S_1, S_2, \dots, S_j$.

We can show that this newly defined f is submodular and hence using Theorem 1 we get an M-approximation algorithm for the 2-Set Cover problem where M is the maximum size of a set.

Exercise: Use Theorem 1 to get an approximation algorithm for the Capacitated Set Cover problem where each set has an additional capacity constraint on how many elements it can actually cover.

6 Submodular Tree Coverage

This problem is also known as Polymatroid Steiner Tree. In the Submodular Tree Coverage problem we are given

- A general tree T with its set of leaves denoted by U.
- A function $f : 2^U \to \mathbb{Z}^+ \cup \{0\}$.
- Each edge e of the tree T has a cost given by c_e.

The goal is to find a subtree S of minimum cost such that $f(S_L) = f(U)$ where S_L is the set of leaves of S.

It is easy to see that Submodular Set Cover is a special case of the Submodular Tree Coverage problem when the tree T is a star.

Theorem 2 *(Calinescu and Zelikovsky 2005) [2]*
There is a polynomial $O\Big(\log^{1+\epsilon} n.\ \log k\Big)$*-approximation algorithm for* Polymatroid Steiner Tree *on trees with* n *nodes and* $k = max_{j \in U} f(\{j\})$.

The Submodular Tree Coverage problem is a generalization of the Group Steiner Tree problem and hence the known hardness for Group Steiner Tree carries over.

Theorem 3 SUBMODULAR TREE COVERAGE *problem has no* $O\left(\log^{1-\epsilon} n. \log k\right)$-*approximation algorithm under some complexity assumptions.*

7 Budgeted Maximum Coverage

BUDGETED MAXIMUM COVERAGE
Input : A collection of sets $\mathbb{S} = \{S_1, S_2, \ldots, S_m\}$ of costs $c_1, c_2, \ldots, c_m$ over a universe $\mathbb{U} = \{u_1, u_2, \ldots, u_n\}$ and weights $w_1, w_2, \ldots, w_n$ and a budget L
Goal : To find a collection $\mathbb{S}' \subseteq \mathbb{S}$ maximizing the total weight of the elements covered by $\mathbb{S}'$ such that the total cost of the elements of $\mathbb{S}'$ does not exceed the budget L.

We can show that the BUDGETED MAXIMUM COVERAGE is NP-hard via a reduction from SET COVER or KNAPSACK.

7.1 A $1 - \frac{1}{e}$-approximation algorithm for the Unit Cost case

In this subsection we will give a $1 - \frac{1}{e}$-approximation for the special case of BUDGETED MAXIMUM COVERAGE when all the costs are 1. Even this special case of the problem is NP-hard.

We give a greedy algorithm which at each step picks up a set which maximizes the weight of the uncovered elements that this set will cover. We will give a formal description and analysis of this algorithm in the next lecture. Now let us see why this algorithm will fail for the general cost case. Consider the following instance :

- $U = \{x_1, x_2\}$ and $L = p + 1$
- $S_1 = \{x_1\}$ and $S_2 = \{x_2\}$
- $w_1 = 1$ and $w_2 = p$
- $c_1 = 1$ and $c_2 = p + 1$

Our greedy algorithm would choose S_1 and then will not have enough budget to pick S_2. But the OPT is clearly to pick S_2. So we will get a weight of 1 instead of p and this p can be made arbitrarily large.

Normally, for hard-capacity constraint problems the greedy approach does not usually work as it only takes steps which are locally optimal.

7.2 An approximation algorithm for the the general cost case

In this subsection we give a $\frac{1}{2}(1 - \frac{1}{e})$-approximation algorithm for the general case : Pick the maximum of the following two quantities

- Output of the above greedy algorithm.
- Pick a set S which maximizes $w(S)$ and satisfies $c(S) \leq L$.

7.3 An improved approximation algorithm for the general cost case

In this subsection we give an improved approximation algorithm for the general cost case due to Khuller, Moss and Naor [3] which achieves a ratio of $1 - \frac{1}{e}$. Fix any integer $k \geq 3$. The algorithm is to output the maximum of the following two types of candidate solutions :

- Consider all subsets of $\mathbb{S}$ of size k which have cost at most L and complete each of them to a solution using the greedy heuristic.
- Consider all subsets of $\mathbb{S}$ of size at most k which have cost at most L.

References

[1] Laurence A. Wolsey, *An analysis of the greedy algorithm for the submodular set covering problem*, Combinatorica, 2(4), 1982

[2] Gruia Calinescu and Alexander Zelikovsky, *The Polymatroid Steiner Problems*, J. Comb. Optim., 9(3), 2005.

[3] Samir Khuller, Anna Moss and Joseph Naor, *The Budgeted Maximum Coverage Problem*, Inf. Process. Lett., 70(1), 1999