A note on maximizing a submodular set function subject to a knapsack constraint

Maxim Sviridenko

IBM, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA

Received 6 March 2002; received in revised form 21 April 2003; accepted 21 April 2003

Abstract

In this paper, we obtain an \((1 - e^{-1})\)-approximation algorithm for maximizing a nondecreasing submodular set function subject to a knapsack constraint. This algorithm requires \(O(n^5)\) function value computations.

\(\text{c}^\text{©} 2003\) Published by Elsevier B.V.

Keywords: Submodular function; Approximation algorithm

1. Introduction

Let \(I = \{1, \ldots, n\}\). Let \(B\) and \(c_i, i \in I\), be nonnegative integers. In this note, we consider the following optimization problem:

\[
\max_{S \subseteq I} \left\{ f(S) : \sum_{i \in S} c_i \leq B \right\},
\]

where \(f(S)\) is a nonnegative, nondecreasing, submodular, polynomially computable set function (a set function is (i) submodular if \(f(S) + f(T) \geq f(S \cup T) + f(S \cap T)\) for all \(S, T \subseteq I\) and (ii) nondecreasing if \(f(S) \leq f(T)\) for all \(S \subseteq T\).

Nemhauser et al. [6] consider the special case of problem (1) with \(c_i = 1\), for all \(i \in I\). They prove that the simple greedy algorithm has performance guarantee \(1 - e^{-1} \approx 0.35\), where \(e\) is a unique root of equation \(e^x = 2 - x\).

The max \(k\)-cover (or maximum coverage) problem with a knapsack constraint is one of the most interesting special cases of problem (1). Khuller et al. [3] prove that the greedy algorithm, combined with the partial enumeration procedure due to Sahni [7], has performance guarantee \(1 - e^{-\beta} \approx 0.632\). This is the best possible performance guarantee achievable in polynomial time, unless \(P = NP\) [1], even in the case when \(c_i = 1\), for all \(i \in I\). Another well-studied example of a problem of maximizing a nondecreasing submodular set function is the entropy of a positive semidefinite matrix (see [2,5,4]).

\(\text{E-mail address: sviri@us.ibm.com}\) (M. Sviridenko).
In this note, we show that the algorithm by Khuller et al. has performance guarantee $1 - e^{-1}$, even for general problem (1). This algorithm requires $O(n^5)$ function value computations.

2. Algorithm and its analysis

Next, we describe a modification of the greedy algorithm for solving problem (1).

In the first phase, the algorithm enumerates all feasible solutions (sets) of cardinality one or two. Let $S_1$ be a feasible set of cardinality one or two that has the largest value of the objective function $f(S)$. In the second phase, the algorithm considers all feasible sets of cardinality three. The algorithm completes each such set greedily and keeps the current solution feasible with respect to the knapsack constraint (see the formal description of the second phase in the next paragraph).

Let $S_2$ be the solution obtained in the second phase that has the largest value of objective function, over all choices of the starting set for the greedy algorithm. Finally, the algorithm outputs $S_1$ if $f(S_1) > f(S_2)$ and $S_2$ otherwise. Below, we formally describe the second phase of the algorithm.

For all $U \subseteq I$ such that $|U| = 3$, carry out the following procedure: Let $S^0 = U, t = 1, t^0 = I$. At step $t$, we have a partial solution $S^{t-1}$. Find

$$\theta_t = \max_{i \in t^{t-1} \setminus S^{t-1}} \frac{f(S^{t-1} \cup \{i\}) - f(S^{t-1})}{c_i}.$$  (3)

Let the maximum in (3) be attained on the index $i_t$. Let $S^t = S^{t-1} \cup \{i_t\}$ and $t = t^t - 1$ if $\sum_{i \in S^{t-1} \cup \{i_t\}} c_i \leq B$.

Otherwise, let $S^t = S^{t-1}$ and $t = t^t - 1 \setminus \{i_t\}$. Let $t = t + 1$, and go to the next step. Stop when $I \setminus S^t = \emptyset$.

In the proof of the performance guarantee, we will use the following inequality due to Wolsey [8]: If $P$ and $D$ are arbitrary positive integers, $\rho_i, i = 1, \ldots, P$, are arbitrary nonnegative reals, and $\rho_1 > 0$ (note that Wolsey uses slightly more general conditions), then

$$\frac{\sum_{i=1}^P \rho_i}{\sum_{i=1}^P (\rho_i + D \rho_i)} \geq 1$$

$$- \left(1 - \frac{1}{D}\right)^P > 1 - e^{-P/D}. \quad (4)$$

Theorem 1. The worst-case performance guarantee of the above greedy algorithm for solving problem (1) is equal to $1 - e^{-1}$.

Proof. If there is an optimal solution to problem (1) with cardinality one, two or three, such a solution will be found by the algorithm by the enumeration of all sets of cardinality three or less. So we assume that the cardinality of any optimal solution is larger than three. Let $S^*$ be an optimal solution to problem (1). We order the set $S^*$ so that

$$f(\{i_1, \ldots, i_3\}) = \max_{i \in S^* \setminus \{i_1, \ldots, i_{3-1}\}} f(\{i_1, \ldots, i_{3-1} \cup \{i\}),$$

i.e. $i_1$ is an element of the optimal set $S^*$ having the greatest value of the objective function, $i_2$ is an element that gives the greatest increase in objective value if we add it to the set $\{i_1\}$, and so on. Let $Y = \{i_1, i_2, i_3\}$ be the set that consists of the first three elements of the set $S^*$. Now, we prove an inequality that is a generalization of inequality (3) from [3]. For any element $i_k \in S^*, k \geq 4$, and set $Z \subseteq I \setminus \{i_1, i_2, i_3, i_k\}$, the following series of inequalities follows from submodularity, the ordering of the set $S^*$, and the fact that $f(\emptyset) \geq 0$:

$$f(Y \cup Z \cup \{i_k\}) - f(Y \cup Z) \leq f(\{i_k\})$$

$$- f(\emptyset) \leq f(\{i_1\}),$$

$$f(Y \cup Z \cup \{i_k\}) - f(Y \cup Z) \leq f(\{i_1\} \cup \{i_k\})$$

$$- f(\{i_1\}) \leq f(\{i_1, i_2\} - f(\{i_1\}),$$

$$f(Y \cup Z \cup \{i_k\}) - f(Y \cup Z) \leq f(\{i_1, i_2\} \cup \{i_k\})$$

$$- f(\{i_1, i_2\}) \leq f(\{i_1, i_2, i_3\} - f(\{i_1, i_2\}).$$

Summing up all these inequalities, we obtain

$$3(f(Y \cup Z \cup \{i_k\}) - f(Y \cup Z))$$

$$\leq f(\{i_1, i_2, i_3\} - f(\{i_1, i_2\}) + f(\{i_1, i_2\})$$

$$- f(\{i_1\}) + f(\{i_1\}) = f(Y). \quad (5)$$

From now on, we consider an iteration of the algorithm in which the set $Y$ was chosen at the beginning of the greedy procedure, i.e. $S^0 = Y$. We will prove that the value of the objective function of the solution obtained in this iteration is at least $1 - e^{-1}$ times the value of the optimal solution.

Define the function $g(S) = f(S) - f(Y)$. It is easy to see that the function $g(S)$ is nondecreasing and submodular if the function $f(S)$ is nondecreasing and submodular. Therefore, $g(S)$ satisfies inequality (2).
It is also clear that the function \( g(S) \) is nonnegative, for all sets \( S \) such that \( Y \subseteq S \subseteq I \), since \( f(S) \) is a nondecreasing function.

Let \( t^* + 1 \) be the first step of the greedy algorithm for which the algorithm does not add element \( i_{t^*+1} \in S^* \) to the set \( S'^* \), i.e. \( S'^* = S^* \) and \( I'^* = I^* \setminus \{i_{t^*+1}\} \). Without loss of generality, we assume that \( t^* + 1 \) is the first step \( t \) for which \( S' = S'^{-1} \) and \( I' = I'^{-1} \setminus \{i\} \). We can do this since if it happens earlier for some \( t' < t^* + 1 \), then \( i_{t'} \notin S^* \), and \( i_{t'} \) does not belong to the approximate solution we are interested in; therefore, excluding \( i_{t'} \) from the ground set \( I \) does not change the analysis, the optimal solution \( S^* \), and the approximate solution obtained in the iteration with \( S'^0 = Y \).

If the element \( i_{t^*+1} \) is not included in the set \( S'^* \), then \( c_{i_{t^*+1}} + \sum_{i \in S'^*} c_i > B \). Let \( S' = 0, \ldots, t^* \), be the sets defined in the description of the algorithm. Applying inequality (2) and the definition of \( g(S) \), we obtain

\[
g(S^*) \leq g(S') + \sum_{i \in S'^* \setminus S} (g(S' \cup \{i\}) - g(S'))
\]

\[
= g(S') + \sum_{i \in S'^* \setminus S} (f(S' \cup \{i\}) - f(S'))
\]

\[
\leq g(S') + \left( B - \sum_{i \in Y} c_i \right) \theta_{t+1}
\]

(6)

for all \( t = 0, \ldots, t^* \). The last inequality follows from the facts that (i) \( f(S' \cup \{i\}) - f(S') \leq c_i \theta_{t+1} \), and (ii) \( \sum_{i \in S'^* \setminus S} c_i \leq B - \sum_{i \in Y} c_i \).

Let \( B_t = \sum_{i=1}^t c_i \) and \( B_0 = 0 \). Note that, by the definition of the element \( i_{t^*+1} \), we have \( B' = B_{t^*+1} > B - \sum_{i \in Y} c_i = B'' \). For \( j = 1, \ldots, B' \), we define \( \rho_j = \theta_j \) if \( j = B_{t-1} + 1, \ldots, B_t \). Using this definition, we obtain

\[
g(S'^* \cup \{i_{t^*+1}\}) = \sum_{t=1}^{t^*+1} c_i \theta_i + \sum_{j=1}^{B'} \rho_j
\]

and inequalities (4) and (6), we obtain

\[
g(S'^* \cup \{i_{t^*+1}\})
\]

\[
\geq \min_{t=1, \ldots, t^*} \left\{ \sum_{j=1}^{B_t} \rho_j + B'' \rho_{B_t+1} \right\}
\]

\[
= \min_{t=1, \ldots, t^*} \left\{ g(S') + B'' \theta_{t+1} \right\}
\]

and inequalities (4) and (6), we obtain

\[
g(S'^* \cup \{i_{t^*+1}\})
\]

\[
\geq \frac{\sum_{j=1}^{B'} \rho_j}{\min_{t=1, \ldots, B'} \left\{ \sum_{j=1}^{B'-1} \rho_j + B'' \rho_{B'} \right\}}
\]

\[
\geq 1 - e^{-B'/B''} > 1 - e^{-1}.
\]

(7)

Combining (5) and (7), we obtain

\[
f(S'^*) = f(Y) + g(S'^*)
\]

\[
= f(Y) + g(S'^* \cup \{i_{t^*+1}\})
\]

\[
- (g(S'^* \cup \{i_{t^*+1}\}) - g(S^{''*})))
\]

\[
= f(Y) + g(S'^* \cup \{i_{t^*+1}\})
\]

\[
- (f(S'^* \cup \{i_{t^*+1}\}) - f(S'^*))
\]

\[
\geq f(Y) + (1 - e^{-1}) g(S^*) - f(Y)/3
\]

\[
\geq (1 - e^{-1}) f(S^*).
\]

Since the output of the algorithm is at least as good as \( S'^* \), this proves the performance guarantee of \( 1 - e^{-1} \).

References


