Reading notes: A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization

Eric Balkanski

April 12, 2015

1 Introduction

Last meeting, we studied the submodular maximization paper of Feige et al. [1] where the submodular function f over a ground set $N = \{u_1, \ldots, u_n\}$ is non-negative, unconstrained, and not necessarily monotone. The main technique used was local search. In these reading reading notes, we explore the paper of Buchbinder et al. [2], which studies submodular maximization in an identical setting and improves on the results of Feige et al. [1] by using an adaptation of the greedy approach. These results are summarized in the table below.

	Feige et al. [1]		Buchbinder et al. [2]	
	Approx.	Runtime	Approx.	Runtime
Deterministic	$\frac{1}{3} - \frac{\epsilon}{n}$	$O(\frac{1}{\epsilon} n^3 \log n)$	$\frac{1}{3}$	$O(n)$
Randomized	$\frac{2}{5} - \frac{9}{5n}$	$O(n^2)$	$\frac{1}{2}$	$O(n)$

The classical greedy algorithm in submodular maximization repeatedly adds the element which maximizes the marginal contribution to the current solution where we start with our solution being the empty set. This approach is effective in the case of monotone submodular maximization by adapting it for different type of constraints. However this greedy algorithm performs poorly with non-monotone submodular functions. In these reading notes, by greedy algorithm, we will refer to a slightly different algorithm. The greedy algorithm that we will be interested about is the algorithm that starts with a solution being the empty set, then goes through all the elements one by one, and adds the current element to the current solution if the marginal contribution of the current element to the current solution is positive. This greedy algorithm can also perform poorly, but we will explore an adaptation of this greedy approach with surprisingly good guarantees. Consider a similar algorithm to this greedy algorithm that starts with a solution consisting of all the elements, and then goes through all the elements one by one, and removes the element from the current solution is its marginal contribution to the current solution is negative. We call this algorithm reverse greedy.

The main idea is to combine greedy and reverse greedy by running both of them concurrently. When going through the elements one by one and when considering an element, information from both greedy and reverse greedy is used to decide whether to add that element to the solution or not. Two algorithms that use this idea are presented: a deterministic and a randomized algorithm.

2 The Deterministic Algorithm

DeterministicUSM (USM standing for Unconstrained Submodular Maximization) is described formally as Algorithm 1. Initially the solutions are $X_0 = \emptyset$ for the greedy algorithm and $Y_0 = N$ for the reverse greedy algorithm. The algorithm goes through each element one by one. At the ith iteration, when considering element u_i, a_i is the marginal contribution of adding u_i to X_{i-1} and b_i is the marginal contribution of removing u_i from Y_{i-1}. The main step of the algorithm is then to compare a_i and b_i and to decide whether u_i should be in our solution based on this comparison. We then update X_i and Y_i based on this decision and therefore get $X_n = Y_n$ as the final solution.

We first make observations about X_i and Y_i:

- $X_i \cup \{i+1, i+2, \ldots, n\} = \emptyset$
- $Y_i \cap \{i+1, i+2, \ldots, n\} = \{i+1, i+2, \ldots, n\}$
- $X_i \subseteq Y_i$
- $X_n = Y_n$

Algorithm 1 DeterministicUSM(f, N)

Start with $X_0 = \emptyset$ and $Y_0 = N$
for $i = 1$ to n **do**
  Let $a_i = f(X_{i-1} \cup \{u_i\}) - f(X_{i-1})$
  Let $b_i = f(Y_{i-1} \setminus \{u_i\}) - f(Y_{i-1})$
  if $a_i \geq b_i$ **then**:
    Let $X_i = X_{i-1} \cup \{u_i\}$, $Y_i = Y_{i-1}$
  else: $X_i = X_{i-1}$, $Y_i = Y_{i-1} \setminus \{u_i\}$
  end if
end for
Return X_n

The first lemma shows that the solution that we are building improves at every step: we show that $a_i + b_i \geq 0$ for all i, which implies that either $a_i \geq 0$ or $b_i \geq 0$.

Lemma 1. *For every i, $a_i + b_i \geq 0$.*

Proof. Observe that

- $(X_{i-1} \cup \{u_i\}) \cup (Y_i \setminus \{u_i\}) = Y_{i-1}$
- $(X_{i-1} \cup \{u_i\}) \cap (Y_i \setminus \{u_i\}) = X_{i-1}$

Therefore,

$$\begin{aligned} a_i + b_i = (f(X_{i-1} \cup \{u_i\}) + f(Y_i \setminus \{u_i\})) \\ - (f(X_{i-1}) + f(Y_{i-1})) \geq 0 \end{aligned}$$

where the inequality holds by submodularity. □

Let OPT be an optimal solution and $OPT_i = (OPT \cup X_i) \cap Y_i$. OPT_i will be the crucial sequence of sets that we will use through the analysis. An alternative way to define OPT_i is that it is the set that coincides with X_i and Y_i on $1, \ldots, i$ and coincides with OPT in $i+1, \ldots, n$. The following are useful properties about OPT_i:

- $OPT_0 = OPT$
- $OPT_n = X_n = Y_n$
- $X_i \subseteq OPT_i \subseteq Y_i$ (useful for submodularity)

Since $OPT_0 = OPT$ and OPT_n is the solution returned by the algorithm, we are going to analyze the sequence $f(OPT_0), \cdot, f(OPT_n)$ and bound the loss in value along that sequence.

Lemma 2. *For all i, $f(OPT_{i-1}) - f(OPT_i) \leq [f(X_i) - f(X_{i-1})] + [f(Y_i) - f(Y_{i-1})]$.*

Assuming Lemma 2 holds, we show that the algorithm is a 1/3 approximation.

Theorem 1. *The approximation guarantee from Algorithm 1 is 1/3.*

Proof. Observer that

$$\begin{aligned} & f(OPT_0) - f(OPT_n) && (1) \\ &= \sum_{i=1}^{n} f(OPT_{i-1}) - f(OPT_i) && (2) \\ &\leq \sum_{i-1}^{n} [f(X_i) - f(X_{i-1})] + [f(Y_i) - f(Y_{i-1})] && (3) \\ &= f(X_n) - f(X_0) + f(Y_n) - f(Y_0) && (4) \\ &= f(X_n) + f(Y_n) && (5) \end{aligned}$$

where (2) and (4) follows from telescoping sums, (3) from Lemma 2, and (5) from $f(X_0)$ and $f(Y_0)$ being non negative. We conclude the proof by noting that the algorithm returns $OPT_n = X_n = Y_n$. □

It remains to prove Lemma 2.

Proof of Lemma 2. Assume that $a_i \geq b_i$ (the case $b_i > a_i$ follows similarly), so

- $X_i = X_{i-1} \cup \{u_i\}$
- $Y_i = Y_{i-1}$
- $OPT_i = OPT_{i-1} \cup \{u_i\}$

We therefore need to show that $f(OPT_{i-1}) - f(OPT_i) \leq f(X_i) - f(X_{i-1}) = a_i$. There are two cases.

If $u_i \in OPT_{i-1}$, then $f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\}) = 0 \leq a_i$ since $a_i \geq 0$ by Lemma 1.

If $u_i \notin OPT_{i-1}$. Then by submodularity,

$$\begin{aligned} & f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\}) \\ & \quad \leq f(Y_{i-1} \setminus \{u_i\}) - f(Y_{i-1}) \\ & \quad = b_i \leq a_i \end{aligned}$$

The authors also note that a 1/3-approximation is tight for this algorithm, we skip the proof of that result.

3 The Randomized Algorithm

The 1/3-approximation of DeterministicUSM is improved to 1/2 with a randomized version of the algorithm called RandomizedUSM, described formally as Algorithm 2, and very similar to the deterministic algorithm. The intuition behind why this randomized

algorithm performs better than the deterministic one is the following. In the case where a_i and b_i are both positive and almost equal, then both the decision of picking u_i or of rejecting u_i could be a good decision and we do not want to commit to either. By either picking it or rejecting it with some probability, we keep the doors open to both possibilities, which makes the decision "smoother".

Algorithm 2 RandomizedUSM(f, N)

Start with $X_0 = \emptyset$ and $Y_0 = N$
for $i = 1$ to n **do**
  Let $a_i = f(X_{i-1} \cup \{u_i\}) - f(X_{i-1})$
  Let $b_i = f(Y_{i-1} \setminus \{u_i\}) - f(Y_{i-1})$
  Let $a_i' = \max\{a_i, 0\}, b_i' = \max\{b_i, 0\}$
  Let $r = 1$ with probability $a_i'/(a_i' + b_i')$
  if $r = 1$ **then**:
    Let $X_i = X_{i-1} \cup \{u_i\}$, $Y_i = Y_{i-1}$
  else: $X_i = X_{i-1}$, $Y_i = Y_{i-1} \setminus \{u_i\}$
  end if
end for
Return X_n

We use the same notation for the analysis as for the deterministic algorithm. The equivalent Lemma for this algorithm to Lemma 2 is the following:

Lemma 3. *For all i, $\mathbf{E}[f(OPT_{i-1}) - f(OPT_i)] \leq \frac{1}{2}\mathbf{E}[[f(X_i) - f(X_{i-1})] + [f(Y_i) - f(Y_{i-1})]]$.*

Assuming Lemma 3 holds, this algorithm is a $1/2$ approximation.

Theorem 2. *The approximation guarantee from Algorithm 1 is $1/2$.*

The proof of Theorem 2 follows similarly as the proof of Theorem 1. We now prove Lemma 3.

Proof of Lemma 3. We condition on $X_{i-1} = S_{i-1} \subseteq \{u_1, \ldots, u_{i-1}\}$ and show that the result holds for any such S_{i-1}. From this conditioning, the following variables become constants:

- $Y_{i-1} = S_{i-1} \cup \{u_i, \ldots, u_n\}$
- $OPT_{i-1} = S_{i-1} \cup (OPT \cap \{u_i, \cdots, u_n\})$
- a_i and b_i

By Lemma 1, it cannot be the case that $a_i < 0$ and $b_i < 0$. There are three other cases. If $a_i \geq 0$ and $b_i \leq 0$: then,

- $r = 1$ with probability 1
- $Y_i = Y_{i-1} = S_{i-1} \cup \{u_i, \ldots, u_n\}$
- $X_i = S_{i-1} \cup \{u_i\}$
- $OPT_i = OPT_{i-1} \cup \{u_i\}$

We need to show that $f(OPT_{i-1}) - f(OPT_i) \leq \frac{1}{2}(f(X_i) - f(X_{i-1})) = a_i/2$. There are two subcases, if $u_i \in OPT_{i-1}$ then $f(OPT_{i-1}) - f(OPT_i) = 0$ while $a_i \geq 0$. If $u_i \notin OPT_{i-1}$, then by submodularity,

$$\begin{aligned} &f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\}) \\ &\leq f(Y_{i-1} \setminus \{u_i\}) - f(Y_{i-1}) \\ &= b_i \leq 0 \leq a_i/2 \end{aligned}$$

If $a_i < 0$ and $b_i \geq 0$, then the proof follows similarly as the previous case. The only case that remains is therefore $a_i \geq 0$ and $b_i \geq 0$. So $a_i' = a_i$ and $b_i' = b_i$ and,

$$\begin{aligned} &\mathbf{E}[[f(X_i) - f(X_{i-1})] + [f(Y_i) - f(Y_{i-1})]] \\ =&\frac{a_i}{a_i + b_i}(f(X_{i-1} \cup \{u_i\}) - f(X_{i-1})) \\ &+ \frac{b_i}{a_i + b_i}(f(Y_{i-1} \setminus \{u_i\}) - f(Y_{i-1})) \\ =&\frac{a_i^2 + b_i^2}{a_i + b_i} \end{aligned}$$

On the other hand,

$$\begin{aligned} &\mathbf{E}[f(OPT_{i-1}) - f(OPT_i)] \\ =&\frac{a_i}{a_i + b_i}(f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\})) \\ &+ \frac{b_i}{a_i + b_i}(f(OPT_{i-1}) - f(OPT_{i-1} \setminus \{u_i\})) \end{aligned}$$

We claim that $\frac{a_i}{a_i+b_i}(f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\})) + \frac{b_i}{a_i+b_i}(f(OPT_{i-1}) - f(OPT_{i-1} \setminus \{u_i\})) \leq \frac{a_i b_i}{a_i+b_i}$. As previously, we have two cases. If $u_i \notin OPT_{i-1}$, then $\frac{b_i}{a_i+b_i}(f(OPT_{i-1}) - f(OPT_{i-1} \setminus \{u_i\})) = 0$ and by submodularity,

$$\begin{aligned} &f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\}) \\ &\leq f(Y_{i-1} \setminus \{u_i\}) - f(Y_{i-1}) \\ &= b_i \end{aligned}$$

Otherwise, $u_i \in OPT_{i-1}$. Then, $\frac{a_i}{a_i+b_i}(f(OPT_{i-1}) - f(OPT_{i-1} \cup \{u_i\})) = 0$ and by submodularity,

$$\begin{aligned} &f(OPT_{i-1}) - f(OPT_{i-1} \setminus \{u_i\}) \\ &\leq f(X_{i-1} \cup \{u_i\}) - f(X_{i-1}) \\ &= a_i \end{aligned}$$

Finally, since $\frac{1}{2}\frac{a_i^2+b_i^2}{a_i+b_i} \geq \frac{a_i b_i}{a_i+b_i}$, this concludes our proof.

4 Conclusion

This paper improves on previous bounds for maximization of non-negative, unconstrained, and not necessarily monotone submodular functions. The main idea is to run two versions of the algorithms, one which start with the empty set and one which starts with the ground set and to combine information from both of the greedys when going through elements. The authors also show that this same idea can be applied to obtain a 3/4 approximation for the problems of Submodular Max-SAT and Submodular Welfare with 2 players. These approximation match the best known approximation ratios for these problems, but the runtime of these new algorithms is linear time.

References

[1] Uriel Feige, Vahab S Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4): 1133–1153, 2011.

[2] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 649–658. IEEE, 2012.