The Continuous Greedy Algorithm

Eric Balkanski

February 28, 2015

1 Introduction

Last meeting, we studied how to maximize a monotone submodular function f subject to a matroid constraint using the pipage rounding technique when f is a sum of weighted rank functions of matroids [1]. The framework is to maximize an extension $\tilde{f}$ of f that is an approximation of the multilinear extension F of f and to then use pipage rounding on F.

In this paper, Vondrák [3] shows that instead of maximizing $\tilde{f}$, we can maximize F directly. The issue with this approach in the framework of [1] is that $\tilde{f}$ can be written as an LP, which can then be maximized easily, however it is not clear how to write an LP formulation for the multilinear relaxation of any submodular function. The solution is to create a continuous greedy process where we slowly move a point inside of the feasible region towards a good solution.

This approach, combined with the pipage rounding technique, gives a $(1-1/e)$-approximation for F and for sudmodular maximization subject to any matroid constraint. This result matches the known lower bound of $(1-1/e+\epsilon)$ for any $\epsilon > 0$ and generalizing the $(1-1/e)$-approximation of [1] for a sum of weighted rank functions of matroids to any submodular function. In particular, this generalization provides a $(1-1/e)$-approximation for the Submodular Welfare Problem in the value oracle model.

2 Preliminaries

The Multilinear Extension $F : [0,1]^N \to \mathbb{R}^+$ of a discrete submodular function $f : 2^N \to \mathbb{R}^+$, also called the extension by expectation, is defined as the expected value of $f(S)$ for some $y \in [0,1]^N$ where each agent i is in S with probability y_i, independently.

Notation: for the rest of these reading notes,

$$S \sim y$$

is a set where each agent i is in S with probability y_i, independently.

Therefore,

$$F(y) = \mathbf{E}_{S\sim y}[f(S)]\, 1 \quad = \sum_{R\subseteq N} f(R) \prod_{i\in R} y_i \prod_{i\notin R}(1-y_i).$$

The following properties of F will be useful (see http://theory.stanford.edu/ jvondrak/CS369P-files/lec17.pdf for proofs):

- If f is monotone, then F is non-decreasing along any line of direction $\mathbf{d} \geq 0$: $\frac{\partial F}{\partial y_j} \geq 0$.
- If f is submodular, then F is concave along any line of direction $\mathbf{d} \geq 0$: $\frac{\partial^2 F}{\partial y_i \partial y_j} \leq 0$.
- If f is submodular, then F is convex along any line of direction $\mathbf{e}_i - \mathbf{e}_j$.

The non-decreasing and concavity properties of F are necessary for the maximization of F and the convexity property of F is necessary for pipage rounding.

3 The Continuous Greedy Process

We start by describing the continuous process. This process will be discretized in the next section, so the purpose of this process is to give an intuition of how a $(1-1/e)$-approximation of F is achieved. This continuous process is for for any down-monotone polytope P and for any smooth monontone submodular functions F, meaning F is non-decreasing, concave, and with second partial derivatives everywhere. The function $y(t)$ defined in this process can be seen as a particle moving in the feasible region.

This particle moves from $t = 0$ to $t = 1$ in a direction constrained by the feasible region P to maximize local gain.

Lemma 1. *$y(1) \in P$ and $F(y(1)) \geq (1-1/e)OPT$*

Algorithm 1 The Continuous Greedy Process

1: Start with $y(0) = 0$
2: Let $v(y) = \text{argmax}_{v \in P}(v \cdot \nabla F(y))$
3: Set $\frac{dy}{dt} = v(y)$
4: Output $y(1)$

Proof. Observe that $y(1)$ is a convex combination of vectors in P, so $y(1) \in P$. Next, we claim the following: for any $y \in P$, $\exists v \in P$ such that $v \cdot \nabla F(y) \geq OPT - F(y)$. Fix $y \in P$, let z be such that $F(z) = OPT$, and defined $v = (z - y)_+$. Then,

$$v \cdot \nabla F(y) \geq F(y + v) - F(y) \geq OPT - F(y)$$

where the first inequality follows from $\frac{\partial F}{\partial y_j} \geq 0$ and $\frac{\partial^2 F}{\partial y_i \partial y_j} \leq 0$. This claim says that the rate of increase in F is at least the deficit $OPT - F(y)$, which is the main observation that will get us an $(1 - 1/e)$-approximation using differential equations.

We now use the chain rule,

$$\frac{dF}{dt} = \sum_j \frac{\partial F(y)}{\partial y_j} \frac{dy_j}{dt} = \nabla F(y) \cdot v(y) \geq OPT - F(y).$$

Solving the differential equation $\frac{dF}{dt} \geq OPT - F(y)$, we get our desired result: $F(y(t)) \geq (1 - e^{-t})OPT$. □

4 Discretization for Matroid Constraints

In this section, Algorithm 1 is discretized to obtain a polynomial time algorithm. The challenge is to discretize the continuous process in steps small enough to get an error from this discretization that is at most $o(1)$. In the continuous greedy algorithm, formally described below, F is the multilinear extension of some monotone submodular function f and the feasibility region is a matroid polytope $P(\mathcal{M})$ with $\mathcal{M} = (N, \mathcal{I})$.

This algorithm is very similar to the continuous process, except that we use f instead of F for computation purposes. Observe that $w(t)$ is the local gain at $y(t)$, which is the analogue of $\nabla F(y)$, and that $I(t)$ is the feasible vector that maximizes the local gain, which is the analogue of $v(y)$.

Note that this algorithm has a polynomial runtime. There are n^2 iterations and in each iteration, n^5 samples are needed for each of the n agents. The maximum weight independent set is found using the greedy algorithm.

Algorithm 2 The Continuous Greedy Algorithm

1: Start with $\delta = 1/n^2$, $t = 0$, and $y(0) = 0$
2: Estimate $w_j(t) := \mathbf{E}\big[f_{R(t)}(j)\big]$ with n^5 samples, where $R(t)$ is the random set such that $R(t) \sim y(t)$
3: Let $I(t)$ be the maximum weight independent set in $\mathcal{M}$ according to weights $w_j(t)$
4: Set $y(t + \delta) = y(t) + \delta \cdot 1_{I(t)}$
5: Output $y(1)$

We first show the following useful lemma, which is a generalization of the alternate definition for submodular functions that we used for the proofs of the greedy algorithm during the first meeting to the multilinear extension of submodular functions.

Lemma 2. *Let f be a monotone submodular function and $y \in [0, 1]^N$, then*

$$OPT \leq F(y) + \max_{I \in \mathcal{I}} \sum_{j \in I} \mathbf{E}_{R \sim y}[f_R(j)]$$

Proof. Let O be such that $f(O) = OPT$, then since f is a submodular function,

$$OPT = f(O) \leq f(R) + \sum_{j \in O} f_R(j)$$

for any R. So taking over the sets R in expectation:

$$\begin{aligned} OPT &\leq \mathbf{E}_{R \sim y}\left[f(R) + \sum_{j \in O} f_R(j)\right] \\ &\leq F(y) + \max_{I \in \mathcal{I}} \sum_{j \in I} \mathbf{E}_{R \sim y}[f_R(j)] \end{aligned}$$

□

We are now ready to prove the main result.

Theorem 1. *$y(1) \in P$ and $F(y(1)) \geq (1 - 1/e - o(1))OPT$ with high probability*

Proof. Again, $y(1)$ is a convex combination of vectors in P, so $y(1) \in P$. Define $D(t)$ to be the random set such that $D(t) \sim \delta \cdot 1_{I(t)}$. So $R(t)$ is the random set with probability that is the current position of y and $D(t)$ is the random set with probability the marginal increase in y at time step t. Observe that

$$F(y(t + \delta)) = \mathbf{E}[f(R(t + \delta))] \geq \mathbf{E}[f(R(t) \cup D(t))]$$

since $\Pr(i \in R(t+\delta)) = y_i(t) + \delta \cdot (1_{I(t)})_i$ and $\Pr(i \in R(t) \cup D(t)) = 1 - (1 - y_i(t))(1 - \delta \cdot (1_{I(t)})_i)$ implies that $\Pr(i \in R(t+\delta)) \geq \Pr(i \in R(t) \cup D(t))$ for any agent i. We now lower bound the increase in F at a step t:

$$
\begin{aligned}
&F(y(t+\delta)) - F(y(t)) && (1)\\
&\geq \mathbf{E}[f(R(t) \cup D(t)) - f(R(t))] && (2)\\
&\geq \sum_j \Pr(D(t) = \{j\})\, \mathbf{E}\big[f_{R(t)}(j)\big] && (3)\\
&= \sum_{j \in I(t)} \delta(1-\delta)^{|I(t)|-1} \mathbf{E}\big[f_{R(t)}(j)\big] && (4)\\
&\geq \delta(1 - n\delta) \sum_{j \in I(t)} \mathbf{E}\big[f_{R(t)}(j)\big] && (5)\\
&\overset{w.h.p.}{\geq} \delta(1 - 1/n)(\max_{I \in \mathcal{I}} \sum_{j \in I} \mathbf{E}\big[f_{R(t)}(j)\big] - OPT/n) && (6)\\
&\geq \delta(1 - 1/n)(OPT - F(y(t)) - OPT/n) && (7)\\
&\geq \delta(O\bar{P}T - F(y(t))). && (8)
\end{aligned}
$$

In (3), we only consider the case where $D(t)$ is a singleton. In (5), note that the runtime could by slightly improved by taking $\delta = 1/nr$ and upper bounding $|I(t)| - 1$ by r where r is the rank of the matroid $\mathcal{M}$. In (6), by Chernoff bounds, the error in the estimate of $w_j(t)$ is OPT/n^2 for any j with high probability since $OPT \geq f_R(j)$. So by union bound, the total error from all the agents in $I(t) \cup \max_{I \in \mathcal{I}}$ is at most OPT/n with high probability. (7) is by lemma 2 and in (8), we take $O\bar{P}T = (1 - 2/n)OPT$. We make the same observation as for the the continuous greedy process: the rate of increase in F is at least the deficit $OPT - F(y)$. Using a similar argument by induction as in the proofs for the greedy algorithms that we saw during our first meeting, we get that $F(y(1)) \geq (1 - 1/e - o(1))OPT$. □

Remark: it is possible to remove the $o(1)$ term by being smarter about how to pick $D(t)$.

5 Submodular Welfare

Consider the submodular welfare problem which consists of n agents with submodular utility functions $w_1, \ldots, w_n$ and m items. The goal is to allocate the items to agents to maximize total welfare.

In this section, we show that the continuous greedy algorithm can be used to obtain a $(1-1/e)$-approximation algorithm for the submodular welfare problem without using pipage rounding. Note that this result is an improvement on the $(1 - 1/e)$-approximation by [2], which is only in the demand oracle model and not for the more general value oracle model.

Define the variable y_{ij} to be the extent to which item j is allocated to agent i. The algorithm described below is the same as the continuous greedy algorithm but with notation adapted to the submodular welfare problem.

Algorithm 3 The Continuous Greedy Algorithm for Submodular Welfare

1: Start with $\delta = 1/(mn)^2$, $t = 0$, and $y(0) = 0$
2: Estimate $w_{ij}(t) := \mathbf{E}\big[f_{R_i(t)}(j)\big]$ with $(mn)^5$ samples, where $R_i(t)$ is the random set of items such that $R_i(t) \sim y_i(t)$
3: For each j, let $i_j(t) = \text{argmax}_i\, w_{ij}(t)$ be the preferred agent for item j.
4: Set $y_{ij}(t+\delta) = y_{ij}(t) + \delta$ if $i = i_j(t)$ and $y_{ij}(t+\delta) = y_{ij}(t)$ otherwise.
5: Output $y(1)$

By theorem 1, using the continuous Greedy algorithm and then pipage rounding, we get a $(1 - 1/e - o(1))$-approximation with high probability. However, an interesting remark is that it is possible to avoid doing pipage rounding.

For each item j independently, allocate item j to at most one agent where each agent has probability y_{ij} of getting the item. First, note that this is possible since for any j, $\sum_i y_{ij} \leq 1$ since our matroid constraint is that we can allocate each item to at most one agent. Second, note that we treating each item j independently but that we are not allocating j to different agents independently, which is fine since this does not affect the objective function $\sum w_i$, that is additive over agents, in expectation.

6 Conclusion

This paper introduces the continuous greedy algorithm, which maximizes the multilinear extension F of a submodular function f subject to a matroid constraint with a $(1 - 1/e)$-appproximation. By combining this algorithm with pipage rounding, we get a $(1-1/e)$-appproximation for submodular maximization subject to any matroid constraint. In particular, this result gives a $(1 - 1/e)$-appproximation for the submodular welfare problem in the value oracle model, improving on previous results. Note that the continuous greedy algorithm is non-deterministic since it requires sampling, an open question is to find a deterministic algorithm that achieves the same approximation.

References

[1] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Integer programming and combinatorial optimization*, pages 182–196. Springer, 2007.

[2] Shahar Dobzinski and Michael Schapira. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1064–1073. Society for Industrial and Applied Mathematics, 2006.

[3] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.