Notes on Local Search and Independence Systems

Justin Ward

February 13, 2012

1 Introduction

In this document, we consider the problem of maximizing a submodular function subject to various constraints. Our main algorithm will be the simple, oblivious local search algorithm that repeatedly attempts to bring a *single* element into the current solution, and then discarding some set of elements to maintain feasibility. In the last section, we briefly discuss the greedy algorithm for k-extendible systems.

Our proofs will rely on the following lemmas In both lemmas, we assume f is a submodular function on ground set $\mathcal{X}$.

Lemma 1.1. *Let $S \subseteq \mathcal{X}$, and $C = \{c_1, \ldots, c_l\} \subseteq \mathcal{X} \setminus S$ Then,*

$$\sum_{i=1}^{l} [f(S + c_i) - f(S)] \geq f(S \cup C) - f(S)$$

Proof. Let $C_i = \{c_x : x < i\}$. Then,

$$\sum_{i=1}^{l} [f(S + c_i) - f(S)] \geq \sum_{i=1}^{l} [f(S \cup C_{i-1} + c_l) - f(S \cup C_{i-1})]$$
$$= \sum_{i=1}^{l} [f(S \cup C_i) - f(S \cup C_{i-1})] = f(S \cup C_l) - f(S \cup C_0) = f(S \cup C) - f(S)$$

where the first inequality follows from submodularity. □

Lemma 1.2. *Let $S \subseteq \mathcal{X}$, and $\{T_i\}_{i=1}^{l}$ be a collection of subsets of S such that each element of S appears in at most k of the subsets. Then,*

$$\sum_{i=1}^{l} [f(S) - f(S \setminus T_i)] \leq kf(S)$$

Proof. Left as an exercise for the next homework. □

2 Matroids

Suppose that we want to maximize a submodular function over a single matroid constraint. We consider the local search routine that repeatedly removes a single element to the current solution and then adds some element not in the current solution to the result. Let S be the locally optimal solution produced by this algorithm on some instance and let O be a globally optimal solution for this instance. We exploit a useful theorem of Brualdi [2] which essentially states that (since we are in a matroid) there is a bijection π between $O \setminus S$ and $S \setminus O$ such that for all elements $x \in O$, we have $(S \setminus \{\pi(x)\}) \cup \{x\}$ independent. For each $x \in O$ our algorithm will consider each of these replacements. Thus, from the local optimality of S, we know that:

$$f(S) \geq f((S \setminus \{\pi(x)\}) \cup \{x\}) \text{ for all } x \in O \setminus S$$

Subtracting $f(S \setminus \{\pi(x)\})$ from each side gives:

$$f(S) - f(S \setminus \{\pi(x)\}) \geq f((S \setminus \{\pi(x)\}) \cup \{x\}) - f(S \setminus \{\pi(x)\}) \text{ for all } x \in O \setminus S$$

From submodularity, we have

$$f((S \setminus \{\pi(x)\}) \cup \{x\}) - f(S \setminus \{\pi(x)\}) \geq f(S \cup \{x\}) - f(S)$$

Thus, we have

$$f(S) - f(S \setminus \{\pi(x)\}) \geq f((S \cup \{x\}) - f(S) \text{ for all } x \in O \setminus S$$

Summing over all x then gives:

$$\sum_{x \in (O \setminus S)} [f(S) - f(S \setminus \{\pi(x)\})] \geq \sum_{x \in (O \setminus S)} [f((S \cup \{x\}) - f(S)]$$

Now, we note that since π is a bijection, each element of $S \setminus O$ appears in at most 1 of the sets $\{\pi(x)\}$. Applying Lemma 1.1 on the right and Lemma 1.2 on the left then gives:

$$f(S) \geq f(S \cup (O \setminus S)) - f(S) = f(O \cup S) - f(S) \geq f(O) - f(S)$$

And thus, $2f(S) \geq f(O)$.

3 Non-Oblivious Local Search for $(k+1)$-Claw Free Graphs

Now, we make 2 modifications to the local search algorithm for weighted MIS in $(k+1)$-claw free graphs. For simplicity, we'll restrict ourselves to the submodular case. First, we make the neighborhood larger, so that we are allowed to bring up to k independent vertices at once (again, we discard all conflicting vertices in the current solution after bringing these k vertices in). Second, we use the function

$$w^2(S) = \sum_{x \in S} w(x)^2$$

to evaluate the current solution, rather than the sum of weights. This is an example of a *non-oblivious* local search algorithm, which is a local search algorithm that uses a secondary potential function to evaluate solutions. Some vague intuition for squaring the weights is as follows: suppose that 2 independent sets have the same total weight but 1 has fewer elements. We should prefer the latter, since it is "more flexible" in future search iterations. This is exactly the behavior we get after squaring the weights.

In order to analyze the algorithm, we use a 2-step charging procedure, presented by Berman [1] in the original analysis of this algorithm. Again, let S be a locally optimal solution for some instance and let O be the globally optimal solution to this instance. We will show how distribute the total weight of O amongst the vertices of S so that no vertex of S receives more than $(k+1)/2$ times its own weight. We charge all vertices in $O \cap S$ their own weight (and only this weight). For a vertex in $O \setminus S$ we proceed as follows. In step 1, each vertex of $v \in O \setminus S$ sends to each of its neighbors x in S weight $w(x)/2$ (note that these are all in fact in $S \setminus O$ since O is independent, thus no vertex of $S \cap O$ will receive any charge in this step). Vertex v may have some weight left over after this step. In step 2, v sends all of its remaining weight to its "heaviest" neighbor in S (i.e. the neighbor with maximal weight).

Consider now the total charge received by any vertex $x \in S \setminus O$. In step 1, it receives $w(x)/2$ from each of its neighbors in O. Because the graph is $(k+1)$-claw free and O is an independent set, there can be at most k such neighbors, so each vertex in $S \setminus O$ receives weight at most $kw(x)/2$ in step 1. We now claim that each vertex receives total weight at most $w(x)/2$ in round 2. Verifying this claim will constitute the remainder of our proof.

For a vertex $x \in S \setminus O$, let P_x be the set of all vertices $y \in N(x, O)$ that have x as their heaviest neighbor. Then, $|P_x| \leq k$ so the local search algorithm will attempt to add each such set to the solution S before stopping. Because S is locally optimal with respect to w^2, we must have $w^2((S \setminus N(P_x)) \cup P_x) \leq w^2(S)$ or, equivalently (since w^2 is simply a sum of squared weights):

$$w^2(P_x) \leq w^2(N(P_x, S))$$

Now, note that each vertex in P_x is adjacent to x, so:

$$w^2(N(P_x)) \leq w(x)^2 + \sum_{y \in P_x} w(N(y) - x)$$

Combining these two inequalities and rearranging using we obtain:

$$\sum_{y \in P_x} \left[w^2(y) - w(N(y) - x) \right] \leq w^2(x) \tag{1}$$

Our next task is to bound the summand on the left above for each term $y \in P_x$. We have:

$$0 \leq (w(x) - w(y))^2 = w(x)^2 - 2w(x)w(y) + w(y)^2$$

Furthermore, since $y \in P_x$, every element $z \in N(y)$ has weight at most $w(x)$. Thus,

$$w^2(N(y) - x) = \sum_{z \in N(y) - x} w(z)^2 \leq \sum_{z \in N(y) - x} w(x)w(z) = w(x) \sum_{z \in N(y) - x} w(z) = w(x)w(N(y) - x)$$

We add these 2 inequalities, rearrange, and then use $w(x)w(N(y)-x)+w(x)^2 = w(x)N(y)$ to obtain:

$$w(y)^2 - w^2(N(y)-x) \geq 2w(x)w(y) - \left[w(x)w(N(y)-x)+w(x)^2\right] = w(x)\left[2w(y) - w(N(y)\right] \ . \tag{2}$$

Applying (2) to each term on the left of (1) then yields:

$$\sum_{y \in P_x} w(x)\left[2w(y) - w(N(y)\right] \leq w^2(x)$$

and so

$$\sum_{y \in P_x} \left[w(y) - \frac{1}{2}w(N(y))\right] \leq \frac{w(x)}{2} \ .$$

Now, note that the term on the left is precisely the total amount of charge that x gets sent in step 2. Thus, we have indeed shown that every vertex x in $S \setminus O$ is sent total charge at most $w(x)/2$ in step 2, and so at most $\frac{k+1}{2}w(x)$ altogether. Thus, $\frac{k+1}{2}w(S) \geq w(O)$.

4 Greedy Algorithms for k-Extendible Set Systems

Here we review some results pertaining to the greedy algorithm on set systems. Because k-systems can be unwieldy to work with, Mestre [4] defined the class of k-extendible systems. These systems are a subset of k-systems, and so all the positive results for k-systems apply here as well. But, it is possible to find a k-system that is not k-extendible, and so we lose an exact characterization of when greedy is a k-approximation.

Definition 4.1 (k-extendible system)**.** We say that independence system $(\mathcal{X}, I)$ is k-extendible if for every $A \subseteq B \subseteq \mathcal{X}$ and $x \notin A$ such that $A + x \in I$, there is a subset $Y \subseteq B \setminus A$ of size at most k such that $(B \setminus Y) + x \in I$.

We now show that the greedy algorithm that repeatedly adds the element with the largest marginal gain with respect to the elements chosen so far gives a $k+1$ approximation for submodular maximization in a k-system. Our proof comes from Calinescu et al [3].

Let O be the optimal solution and S be the solution returned by the greedy algorithm, and suppose that $m = |S|$. We partition O into O_i where $1 \leq i \leq m$ and $|O_i| \leq k$ for all i. Let S_i be the greedy solution before the ith element, e_i has been added (so, $S_1 = \emptyset$). Now, we define a sequence of sets $\emptyset = T_m \subseteq T_{m-1} \subseteq \ldots T_1 = O$ such that $S_i \cup T_i$ is independent and $S_i \cap T_i = \emptyset$ for all i. We set $O_i = T_{i-1} \setminus T_i$ and so we additionally want $|T_{i-1} \setminus T_i| \leq k$. Note that T_1 already satisfies the required properties, so it suffices to show how to construct T_i from T_{i-1}, maintaining these properties. If the greedily chosen element e_i is in T_{i-1}, then set $T_i = T_{i-1} - e_i$ (and so $O_i = \{e_i\}$). Otherwise, since $S_{i-1} + e_i$ and $S_{i-1} \cup T_{i-1}$ are independent, there must be some subset Y of $(S_{i-1} \cup T_{i-1}) \setminus S_{i-1} = T_{i-1}$ of size at most k such that $(S_{i-1} \cup T_{i-1}) \setminus Y + e_i$ is independent. Choose the smallest such subset Y and let $T_i = T_{i-1} \setminus Y$ (and hence $O_i = Y$).

Now, we must make use of the fact that our greedy algorithm always chooses the element with the maximal marginal gain (with respect to the current solution). We denote by $f_A(x)$ be the marginal gain $f(A+x)-f(A)$ obtained when element x is added to set A and similarly for a set B we denote by $f_A(B)$ the gain $f(A \cup B) \setminus f(A)$. Now, we note that since $T_i \cup S_i$ is independent and $T_i \cap S_i = \emptyset$, each element $x \in T_i$ is considered at step i of the greedy algorithm. Hence, we must have $f_{S_i}(x) \geq f_{S_i}(e_i)$ for all $x \in T_i$ and thus for all $x \in O_i$. Hence we have:

$$kf(S) = \sum_{i=1}^{m} kf_{S_i}(e_i) \geq \sum_{i=1}^{m} \sum_{x \in O_i} f_{S_i}(x) \geq \sum_{i=1}^{m} f_{S_i}(O_i) \geq \sum_{i=1}^{m} f_{S_m}(O_m) \geq f(S \cup O) - f(S)$$

where we have used submodularity (in particular, Lemma 1.1) in the second and third inequalities and $|O_i| \leq k$ and $f_{S_i}(e_i) \geq f(x)$ for all $x \in O_i$ in the first.

References

[1] Piotr Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. *Nordic Journal of Computing*, 7:178–184, September 2000.

[2] Richard A. Brualdi. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society*, 1(02):161–167, 1969.

[3] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. *SIAM Journal on Computing*, special issue for STOC 2008, (to appear).

[4] Julián Mestre. Greedy in approximation algorithms. In *Proceedings of the 14th conference on Annual European Symposium on Algorithms - Volume 14*, pages 528–539, London, UK, 2006. Springer-Verlag. ACM ID: 1276239.