MIT 6.854/18.415: Advanced Algorithms	Spring 2016
Lecture 13 – March 16, 2016	
Prof. Ankur Moitra	*Scribe: Emilio Pace, Max W. Shen, Sachin Shinde*

1 Introduction

In this section, we consider a specific application of algorithms for continuous convex optimization (e.g. the ellipsoid method presented in Lecture 12) to efficiently minimize submodular functions, a discrete analogue of convex functions.

2 Submodular Functions

Definition 1 (Submodular Function)**.** *Consider a set N of size n. A function $f : 2^N \to \mathbb{R}$ (where 2^N denotes the power set of N) is submodular if, for all subsets $A \subseteq B \subseteq N$ and elements $j \in N, j \notin B$,*

$$f(A \cup \{j\}) - f(A) \geq f(B \cup \{j\}) - f(B). \tag{1}$$

Intuitively, a submodular function exhibits "diminishing returns" as elements are added to a subset of N: adding the element j to the larger set B causes f to grow no more than adding j to the smaller set A. Alternatively, if adding $\{j\}$ to A decreases the value of f, the decrease will only be larger when $\{j\}$ is added to B.

Corollary 2. *f is submodular if and only if for all subsets $A, B \subseteq N$,*

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B). \tag{2}$$

Proof. Good exercise, or see pages 25–29 of Prof. Jeff Bilmes's lecture slides [1]. □

2.1 Example: Coverage Function

Consider a bipartite graph $G = (V, E)$ with parts N and M as shown in Figure 1. For $A \subseteq N$, the function $f(A) = |neighborhood(A)|$ (i.e. the number of nodes in M connected to at least one node in A) is a submodular function.

A practical example arrises when adding sensors to observe an area. Adding a new sensor will either cover an entirely new area, or overlap with area that is already covered. For sensor subsets A and B with $A \subseteq B$, the new sensor cannot cover more area once added to the set B than when added to the set A.

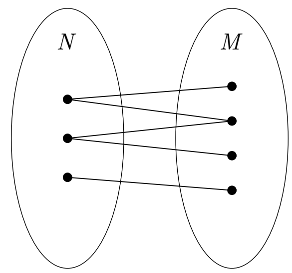

Figure 1: Bipartite graph G.

2.2 Example: Entropy

Consider a collection of n random variables $X_1, \ldots, X_n$. The joint entropy function

$$\begin{aligned} f(A) &= H(\{X_i\}_{i \in A}) \\ &= \sum_{\{x_i\}_{i \in A}} -P(\{x_i\}_{i \in A}) \log_2[P(\{x_i\}_{i \in A})] \end{aligned}$$

for $A \subseteq \{1, \ldots, n\}$ is a submodular function.

2.3 Example: Graph Cut Function

Consider a graph $G = (V, E)$. Then $f(A) = |E(A, V \setminus A)|$, the number of edges in the cut-set, is a submodular function.

3 Optimizing over Convex Sets

It turns out that we can reduce minimizing a submodular function to minimizing a convex function with convex constraints (i.e. over a convex set). We will first explore how we can solve this continuous optimization problem efficiently.

Definition 3 (Convex Set)**.** *A set $S \subseteq \mathbb{R}^n$ is convex if for all $x, y \in S$ and $\lambda \in [0, 1]$, we have*

$$\lambda x + (1 - \lambda) y \in S. \tag{3}$$

Definition 4 (Convex Function)**.** *A function $g : S \to \mathbb{R}$ is convex on a convex set S if, for all $x, y \in S$ and $\lambda \in [0, 1]$,*

$$g(\lambda x + (1 - \lambda) y) \leq \lambda g(x) + (1 - \lambda) g(y). \tag{4}$$

Intuitively, $g(z)$ lies below the line connecting $(x, g(x))$ and $(y, g(y))$ for all z in between x and y.

To minimize convex $g : \mathbb{R}^n \to \mathbb{R}$ over a convex set P, we can use the ellipsoid method to find a point within the subset $S_c = \{x \mid x \in P \wedge g(x) \leq c\}$, and use a technique such as binary search to find the right c. It's easy to see that the convexity of P and g implies that S_c is a convex set.

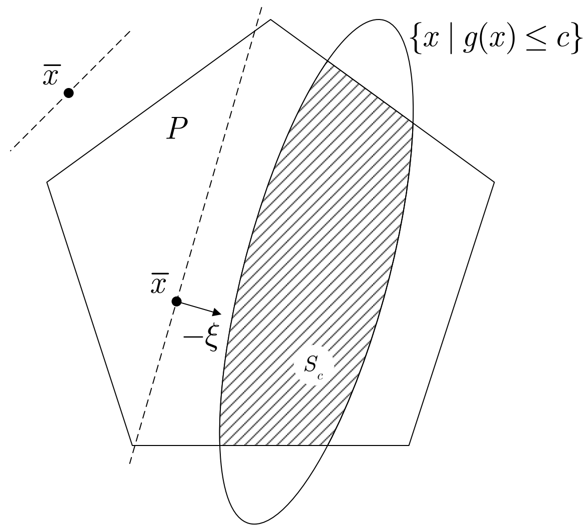

Figure 2: Convex optimization of g over P via the ellipsoid method.

The separation oracle for S_c can be computed efficiently provided we have a separation oracle for P. Given a query point $\overline{x}$, we first check whether $\overline{x} \in P$ using P's oracle, returning the resultant hyperplane if $\overline{x} \notin P$. If $\overline{x} \in P$, then return $\overline{x} \in S_c$ if $g(x) \leq c$, and return a subgradient of g at $\overline{x}$ otherwise. You can usually just think of the subgradient as a gradient, but we use a slightly more general definition to account for non-smooth level sets (i.e. when S_c has sharp corners).

Definition 5 (Subgradient)**.** *A subgradient of $g : \mathbb{R}^n \to \mathbb{R}$ at $\overline{x}$ is a vector $\xi \in \mathbb{R}^n$ such that for all $x \in \mathbb{R}^n$,*

$$g(x) \geq g(\overline{x}) + \xi^T (x - \overline{x}). \tag{5}$$

If we are given any $\overline{x} \notin S_c$ and a subgradient ξ of g at $\overline{x}$, we have for all $x \in S_c$ that

$$\begin{aligned} \xi^T x &\leq \xi^T \overline{x} + g(x) - g(\overline{x}) & (6)\\ &< \xi^T \overline{x} + c - c & (7)\\ &= \xi^T \overline{x}, & (8) \end{aligned}$$

so ξ separates $\overline{x}$ from S_c.

This tells us that the ellipsoid method can be used to efficiently minimize a convex function on a convex set.

4 Lovász Extension

Let's now examine how we can extend any submodular function to a convex functions. Note that any submodular function $f_0 : 2^N \to \mathbb{R}$ can be represented as a binary function $f : \{0,1\}^n \to \mathbb{R}$, by enumerating N as $\{a_1, \ldots, a_n\}$ and letting $f(b_1 b_2 \ldots b_n) = f_0\big(\bigcup_{i|b_i=1} \{a_i\}\big)$. Hereon, we will consider the binary representation, treating N as the set $\{1, 2, \ldots, n\}$.

Definition 6 (Lovász Extension)**.** *Given a binary function $f : \{0,1\}^n \to \mathbb{R}$, its Lovász extension $\hat{f} : [0,1]^n \to \mathbb{R}$ is given by*

$$\hat{f}(z) = \mathbb{E}_{\lambda \sim Unif[0,1]}[f(\{i | z_i \geq \lambda\})], \tag{9}$$

which extends f to a continuous function on $[0,1]^n$.

The first thing to note is that $\hat{f}(x) = f(x)$ for any $x \in \{0,1\}^n$. Accordingly, the minimum of $\hat{f}(x)$ is immediately a lower bound on the minimum of $f(x)$. Additionally, since $\hat{f}(z)$ is defined as the expectation of $f(X_z)$ for a random variable $X_z \in \{0,1\}^n$, it must be that for all z there is some realization of X_z (i.e. some $x \in \{0,1\}^n$) such that $f(x) \leq \hat{f}(z)$.

It follows that the minimums of $\hat{f}(x)$ and $f(x)$ must be exactly equal. So, if we can minimize the Lovász extension, we can minimize our convex function[1].

Now we just need to know whether or not we can minimize the Lovász extension efficiently. The following theorem, due to Lovász, implies that we can:

Theorem 7. *$\hat{f}$ is convex if and only if f is submodular.*

For our purpose of optimizing f, we only need to prove the backwards direction; that if f is submodular, then $\hat{f}$ is convex.

Proof. To make this easier to follow, we will use two assumptions. First, let's assume that $\hat{f}(\emptyset) = 0$ (note that subtracting off $f(\emptyset)$ preserves submodularity). Secondly, let's assume that the components of $z \in [0,1]^n$ are sorted in descending order as $z_1 \geq z_2 \geq \ldots \geq z_n$. The fully general case can be recovered by permuting the coordinates and keeping track of the permutation.

Now define $S_i = \{1, 2, \ldots, i\}$. Expanding the expectation in Definition 6 gives

$$\hat{f}(z) = \sum_{i=1}^{n-1} (z_i - z_{i+1}) f(S_i) + z_n f(S_n) \tag{10}$$

because $z_i - z_{i+1}$ is the probability that $z_{i+1} \leq \lambda \leq z_i$.

The key idea is to show that $\hat{f}(z)$ is the solution to the following maximization problem:

$$\begin{aligned} (P): \quad & \max_x \ z^T x \\ & \text{s.t.} \quad x(S) \leq f(S) \quad \forall\, S \subsetneq N && \text{(11a)} \\ & \qquad\ \ x(N) = f(N) && \text{(11b)} \end{aligned}$$

where we have defined $x(S) = \sum_{i \in S} x_i$.

To see why this works, let F denote the feasible region (which is independent of z), and let $f^*(z)$ denote the optimal solution to (P) for a given $z \in [0,1]^n$. For any $z, z' \in [0,1]^n$ and $\lambda \in [0,1]$, we

[1] There are still a few details to work through – e.g. we need to ensure that when we return a minimize for $\hat{f}(x)$, it is actually an extreme point of $[0,1]^n$. We have shown that there *always is* a minimizer that is an extreme point, but there could be valid minimizers that are not.

have

$$
\begin{aligned}
f^*(\lambda z + (1-\lambda)z') &= \max_{x\in F}(\lambda z + (1-\lambda)z')^T x && (12)\\
&\le \lambda \max_{x\in F} z^T x + (1-\lambda)\max_{x\in F} z'^T x && (13)\\
&= \lambda f^*(z) + (1-\lambda) f^*(z'). && (14)
\end{aligned}
$$

and so $f^*(z)$ is convex. So proving $\hat{f}(z) = f^*(z)$ suffices to prove Theorem 7.

To do this, we will use weak duality. (P)'s dual is given by

$$
\begin{aligned}
(D): \quad & \min_y \sum_{S\subseteq N} y_S f(S) \\
& \text{s.t.} \sum_{S\subseteq N} y_S e_S = z && (15a)\\
& \qquad y_S \ge 0 \quad \forall\, S \subsetneq N && (15b)
\end{aligned}
$$

where e_S is the indicator function on S, i.e.

$$
(e_S)_i = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}.
$$

Weak duality tells us that for any feasible x and y, we have

$$
z^T x \le \sum_{S\subseteq N} y_S f(S). \tag{16}
$$

Therefore, to find the optimum $f^*(z)$, it suffices to find a feasible x^* and y^* with

$$
z^T x^* = \sum_{S\subseteq N} y_S^* f(S). \tag{17}
$$

It turns out that (17) is satisfied if we define x^* and y^* as:

$$
\begin{aligned}
x_i^* &= f(S_i) - f(S_{i-1}) && (18)\\
y_S^* &= \begin{cases} z_i - z_{i-1} & \text{if } S = S_i \text{ for } i < n \\ z_n & \text{if } S = N \\ 0 & \text{otherwise} \end{cases} && (19)
\end{aligned}
$$

Notice that by rearranging, we have

$$
\begin{aligned}
z^T x^* &= \sum_{i=1}^{n} z_i (f(S_i) - f(S_{i-1})) && (20)\\
&= \sum_{i=1}^{n-1} (z_i - z_{i+1}) f(S_i) + z_n f(S_n) && (21)\\
&= \sum_{S\subseteq N} y_S^* f(S). && (22)
\end{aligned}
$$

So, as long as x^* and y^* are feasible, x^* is optimal and as hoped,

$$f^*(z) = z^T x^* \tag{23}$$
$$= \sum_{i=1}^{n-1}(z_i - z_{i+1}) f(S_i) + z_n f(S_n) \tag{24}$$
$$= \hat{f}(z). \tag{25}$$

Let's first show that x^* is feasible. Recalling the assumption that $f(S_0) = f(\emptyset) = 0$, we have

$$x^*(N) = \sum_{i=1}^{n} f(S_i) - f(S_{i-1}) \tag{26}$$
$$= f(S_n) - f(S_0) \tag{27}$$
$$= f(N) \tag{28}$$

as desired. To show constraint (11a) is satisfied, let's induct on $|S|$. The base case of $|S| = 0$ trivially holds as $x^*(\emptyset) = 0 \leq f(\emptyset)$. For the inductive step, let i be the largest element of S. By Corollary 2, we have

$$f(S) + f(S_{i-1}) \geq f(S \cup S_{i-1}) + f(S \cap S_{i-1}) \tag{29}$$
$$= f(S_i) + f(S \setminus \{i\}) \tag{30}$$

which rearranges to

$$f(S) \geq f(S_i) - f(S_{i-1}) + f(S \setminus \{i\}) \tag{31}$$
$$= x_i^* + f(S \setminus \{i\}). \tag{32}$$

Since $|S \setminus \{i\}| = |S| - 1$, the induction hypothesis gives $x^*(S \setminus \{i\}) \leq f(S \setminus \{i\})$, so

$$f(S) \geq x_i^* + x^*(S \setminus \{i\}) \tag{33}$$
$$= x^*(S) \tag{34}$$

and the inductive step is complete.

The case for y^* is more straightforward. First note that for any $i \in N$,

$$\left(\sum_{S \subseteq N} y_S^* e_S\right)_i = \left(\sum_{j=1}^{n-1}(z_j - z_{j+1}) e_{S_j} + z_n e_{S_n}\right)_i \tag{35}$$
$$= \sum_{j=i}^{n-1}(z_j - z_{j+1}) + z_n \tag{36}$$
$$= z_i \tag{37}$$

and thus constraint (15a) holds.

Furthermore, our assumption that $z_1 \geq z_2 \geq \ldots \geq z_n$ implies that $z_i - z_{i+1} \geq 0$ for all $i < n$. As y_S^* takes on one of these values or zero for all $S \subsetneq N$, we have $y_S^* \geq 0$ for all $S \subsetneq N$. That is, y^* satisfies constraint (15b), and is therefore feasible. $\square$

References

[1] J. Bilmes. EE595. Class Lecture, Topic: "Submodular functions, their optimization and applications." Dept. of Elect. Eng., Univ. Washington, Seattle, Apr. 1, 2011 [Online]. Available: http://melodi.ee.washington.edu/~bilmes/ee595a_spring_2011/lecture2.pdf