ENEE 698C - Controls Seminar: Advanced Convex Optimization

Introduction to Submodularity

Raviteja Vemulapalli

Professor Michael C. Rotkowitz

February 26, 2013

Contents

Notations

- $|A|$ denotes cardinality of set A.
- 2^A denotes the power set of set A.
- $A - B$ denotes the set of all elements in A that are not in B.
- $\emptyset$ denotes the empty set.
- $\vec{1}_X$ is a column vector of appropriate size, where $(\vec{1}_X)_k = 1$ if $k \in X$ and $(\vec{1}_X)_k = 0$ if $k \notin X$.
- $\mathcal{R}^p$ represents p-dimensional real space.
- $\mathcal{R}^p_+$ represents the non-negative orthant of p-dimensional real space.
- $\det(\Sigma)$ denotes the determinant of matrix Σ.

1 Submodularity

Till now we have learned a lot of things related to continuous optimization, where the variables take continuous values. Contrary to this, we will be looking at discrete optimization today.

Note: Through out this document, we use I to represent the set $\{1, 2, \ldots, p\}$. Remember that $|2^I| = 2^p$.

1.1 Set Functions

Let $F : 2^I \rightarrow \mathcal{R}$ be a real-valued function defined on the subsets of I. We refer to such functions as *set functions*. In this talk we are interested in optimization problems of the following type, where $\mathcal{S} \subseteq 2^I$:

$$\begin{aligned} \text{min/max} \quad & F(X) \\ \text{subject to} \quad & X \in \mathcal{S}. \end{aligned} \tag{1}$$

Example 1.1 - Sensor placement - Maximum coverage: Consider the problem of placing temperature sensors in a building. Let I be the index set of the set of all candidate locations (assumed to be finite). Let A_k represent the area of the building that would we covered by a sensor at location k. Then the total area covered by a set of sensors placed according to $X \in 2^I$ is given by $F(X) = \cup_{k \in X} A_k$. Our goal is to maximize $F(X)$ subject to some constraints like $|X| \leq N$.

Example 1.2 - Image segmentation - Minimum cut: In this task we are interested in dividing an image into two partitions having different visual content. Consider a weighted graph $G = (V, E)$, where each node represents a pixel in the image. Let $W = [w_{ij}]$ be the matrix of positive weights, where w_{ij} is the weight for the edge between nodes i and j. The value of w_{ij} represents the visual similarity between pixels i and j. Now, image segmentation can be formulated as a graph partitioning problem. Our aim is to divide the vertex set V into two disjoint sets X and $V - X$ such that the two partitions have different visual content. The visual similarity between X and $V - X$ can be measured by the sum of weights of all the edges from X to $V - X$, i.e., $F(X) = \sum_{i \in X,\ j \in V - X} w_{ij}$. So image segmentation can be formulated as minimizing $F(X)$ subject to the constraint $|X| \geq 1$.

Example 1.3 - Facility/Plant location: Suppose a company wants to establish new facilities to serve the demand from m specific cities. Let I be the index set of the set of all candidate locations, and let $M = \{1, 2, \ldots, m\}$ be the index set of the set of cities. Assume that the facilities would have unlimited capacity once built. Let c_j

be the cost of building a facility at location $j \in I$. Let $b_{ij}, i \in M, j \in I$ be the profit achieved if the city i is handled by the facility at location j. Suppose we decide to establish the facilities at locations given by $X \in 2^I$. Then, the cost of building the facilities is $\sum_{k \in X} c_k$ and the profit is $\sum_{i \in M} (\max_{j \in X} b_{ij})$. Hence, we can decide the locations by maximizing the overall profit $F(X) = \sum_{i \in M} (\max_{j \in X} b_{ij}) - \sum_{k \in X} c_k$.

Note that problems of the above kind are combinatorial in nature and simple brute-force search can have exponential runtime. So, a natural question to ask would be: ***Under what conditions can we solve such problems efficiently?***.

Looks like submodularity would help us to some extent. Lets check it out…

1.2 Submodular and Supermodular Functions

Definition 1.1 - Submodular function: A set function $F : 2^I \to \mathcal{R}$ is submodular if and only if for any $X, Y \subseteq I$, we have

$$F(X) + F(Y) \geq F(X \cup Y) + F(X \cap Y). \tag{2}$$

Other equivalent definitions:

- F is submodular $\iff$ for any $X \subseteq Y \subset I$, and $Z \subseteq I - Y$, we have
$$F(X \cup Z) - F(X) \geq F(Y \cup Z) - F(Y). \tag{3}$$
- F is submodular $\iff$ for any $X \subseteq Y \subset I$, and $\{j\} \subseteq I - Y$, we have
$$F(X \cup \{j\}) - F(X) \geq F(Y \cup \{j\}) - F(Y). \tag{4}$$
- F is submodular $\iff$ for any $X \subset I$, and $i, j \in I - X$, we have
$$F(X \cup \{j\}) - F(X) \geq F(X \cup \{i, j\}) - F(X \cup \{i\}). \tag{5}$$

Equations (4), (3) and (5) reflect the "diminishing return" property of the submodular functions. If strict inequality holds in the above equations, then the function F is strictly submodular.

Definition 1.2 - Supermodular function: A set function $F : 2^I \to \mathcal{R}$ is (strictly)supermodular if and only if $-F$ is (strictly)submodular.

Definition 1.3 - Modular function: A set function $F : 2^I \to \mathcal{R}$ is modular if and only if it is both submodular and supermodular.

Definition 1.4 - Monotone increasing submodular function: A submodular function $F : 2^I \to \mathcal{R}$ is monotone increasing(non-decreasing) if and only if for every $X \subset Y \subseteq I$, we have $F(X) < (\leq)\ F(Y)$.

Definition 1.5 - Monotone decreasing submodular function: A submodular function $F : 2^I \to \mathcal{R}$ is monotone decreasing(non-increasing) if and only if for every $X \subset Y \subseteq I$, we have $F(X) > (\geq)\ F(Y)$.

Definition 1.6 - Symmetric function: A set function $F : 2^I \to \mathcal{R}$ is said to be symmetric if $F(X) = F(I - X)$.

Definition 1.7 - Complement function: Given a set function $F : 2^I \to \mathcal{R}$, its complement $\bar{F} : 2^I \to \mathcal{R}$ is defined as $\bar{F}(X) = F(I - X)$. Note that $\bar{\bar{F}} = F$.

Theorem 1.1: A set function $F : 2^I \to \mathcal{R}$ is submodular(supermodular) if and only if its complement $\bar{F}$ is submodular(supermodular).

Proof: Let $X, Y \subseteq I$. If F is submodular, we have

$$\begin{aligned} & F(I - X) + F(I - Y) \geq F((I - X) \cap (I - Y)) + F((I - X) \cup (I - Y)) \\ \implies & \bar{F}(X) + \bar{F}(Y) \geq F(I - (X \cup Y)) + F(I - (X \cap Y)) \\ \implies & \bar{F}(X) + \bar{F}(Y) \geq \bar{F}(X \cup Y) + \bar{F}(X \cap Y)) \end{aligned} \tag{6}$$

So, if F is submodular, $\bar{F}$ is also submodular. Similarly if $\bar{F}$ is submodular, $\bar{\bar{F}} = F$ is also submodular.

Example 1.4 - Coverage: Let I be the index set of a set of areas $\{A_1, A_2, \ldots, A_p\}$. Then the cover function defined as $F(X) = \cup_{k \in X} A_k$, for $X \subseteq I$, is submodular.

Example 1.5 - Entropy: Let I be the index set of a set of random variables $\{U_1, U_2, \ldots, U_p\}$. We will use the notation U_X to denote the set of random variables $\{U_k\}_{k \in X}$. Then the function $F(X) = H(U_X)$, defined for $X \subseteq I$, is submodular. Note that $H(\emptyset) = 0$.

Example 1.6 - Mutual information: Let I be the index set of a set of random variables $\{U_1, U_2, \ldots, U_p\}$. Then the function $F(X) = I(U_X; U_{I-X}) = H(U_X) + H(U_{I-X}) - H(U_I)$, defined for $X \subseteq I$, is submodular.

Example 1.7 - Graph functions: Let $G = (V, E)$ be a weighted graph, where V is the set of nodes and E is the set of edges. Let $W = [w_{ij}]$ be the matrix of positive weights, where w_{ij} is the weight for the edge between nodes i and j. Let $\Gamma(X)$ for $X \subseteq V$, denote the set of all nodes in $V - X$ that are connected to the nodes in X.

Let $Id(X)$ for $X \subseteq V$, denote the set of all edges with atleast one vertex in X. Let $E(X)$ for $X \subseteq V$, denote the set of all edges with both vertices in X. Let $C(X)$ for $X \subseteq E$, denote the number of connected components in the subgraph (V, X). Let $\bar{C}(X)$ for $X \subseteq E$, denote the number of connected components in the subgraph $(V, E - X)$. Then the following functions are submodular

- Cut function: $F(X) = \sum_{i \in X,\ j \in V - X} w_{ij}$ defined for $X \subseteq V$,
- Neighbor function: $F(X) = |\Gamma(X)|$,
- Incidence function: $F(X) = |Id(X)|$,

and the following functions are supermodular:

- Interior edge function: $F(X) = |E(X)|$,
- Connected components function: $F(X) = C(X)$ and $F(X) = \bar{C}(X)$.

Example 1.8 - Rank: Let $A = [a_1, a_2, \ldots, a_p] \in \mathcal{R}^{n \times p}$. Let I be the set of column vector indices. Given $X = \{x_1, x_2, \ldots, x_d\} \subseteq I$ Let $r(X)$ denote the rank of the sub-matrix $[a_{x_1}, a_{x_2}, \ldots, a_{x_d}]$. Then $r(X)$ is submodular.

Example 1.9 - LogDet: Let $\Sigma \in \mathcal{R}^{p \times p}$ be a symmetric positive definite matrix. For $X (\neq \emptyset) \subseteq I$, let Σ_X denote the sub-matrix obtained from Σ by only including the entries in the rows/columns given by X. Then $F(X) = \log(\det(\Sigma_X))$ with $F(\emptyset) = 0$, is submodular.

Example 1.9 - Cardinality: The function $F(X) = |X|$, defined for $X \subseteq I$, is modular.

Example 1.10: For any $s \in \mathcal{R}^p$, the function $F(X) = \sum_{k \in X} s_k$ defined for $X \subseteq I$, is modular.

Example 1.11: For any $s \in \mathcal{R}_+^p$, the function defined as $F(X) = \max_{k \in X} s_k$ for $X \subseteq I$, with $F(\emptyset) = 0$, is submodular.

Example 1.12: For any $s \in \mathcal{R}_+^p$, the function $F(X) = g\left(\sum_{k \in X} s_k\right)$, defined for $X \subseteq I$, is submodular if g is a concave function.

Properties:

- If $F : 2^I \to \mathcal{R}$ and $G : 2^I \to \mathcal{R}$ are submodular, then $\alpha_1 F + \alpha_2 G$ is submodular $\forall \alpha_1, \alpha_2 \in \mathcal{R}_+$.

- If $F : 2^I \to \mathcal{R}$ is submodular and $G : 2^I \to \mathcal{R}$ is modular, then $\alpha_1 F + \alpha_2 G$ is submodular $\forall \alpha_1 \in \mathcal{R}_+$ and $\forall \alpha_2 \in \mathcal{R}$.
- If $F : 2^I \to \mathcal{R}$ is submodular, then the functions $F^{'} : 2^I \to \mathcal{R}$ and $F^{''} : 2^I \to \mathcal{R}$ defined as $F^{'}(X) = F(X \cap A)$ and $F^{''}(X) = F(X \cup A)$ for some $A \subseteq I$, are submodular.
- If $F : 2^I \to \mathcal{R}$ is a monotone submodular function, then $\min(F(X), K)$ is submodular, where K is any constant.
- If $F : 2^I \to \mathcal{R}$ and $G : 2^I \to \mathcal{R}$ are submodular, and $F - G$ is either monotone nonincreasing or nondecreasing, then $F(X) = \min(F(X), G(X))$ is submodular.
- If $F : 2^I \to \mathcal{R}$ is submodular, then $G(X) = \min_{Y \subseteq X} F(Y)$ is monotone nonincreasing submodular function.
- If $F : 2^I \to \mathcal{R}$ is submodular and $M : 2^I \to \mathcal{R}$ is modular, then the convolution of F and M defined as $G(X) = \min_{Y \subseteq X}(F(Y) + M(X - Y))$ is submodular.

Note: If $F : 2^I \to \mathcal{R}$ and $G : 2^I \to \mathcal{R}$ are submodular, $\min(F(X), G(X))$ and $\max(F(X), G(X))$ are not necessarily submodular.

2 Submodular Minimization

In this section we will cover unconstrained submodular minimization.

Note: Given a $s \in \mathcal{R}^p$, we use F_s to denote a set function defined over 2^I, whose values are given by $F_s(X) = \sum_{k \in X} s_k$. Note that $F_s(\emptyset) = 0$. In the rest of this document all the set functions F are assumed to be defined on 2^I and $F(\emptyset) = 0$.

2.1 Submodular and Base Polyhedra

Definition 2.1 - Submodular polyhedron: The submodular polyhedron P_F of a submodular function F is defined as

$$P_F = \{s \in \mathcal{R}^p : \forall X \subseteq I,\ F_s(X) \le F(X)\}. \tag{7}$$

Definition 2.2 - Base polyhedron: The base polyhedron B_F of a submodular function F is defined as

$$B_F = \{s \in \mathcal{R}^p : F_s(I) = F(I),\ \forall X \subset I,\ F_s(X) \le F(X)\}. \tag{8}$$

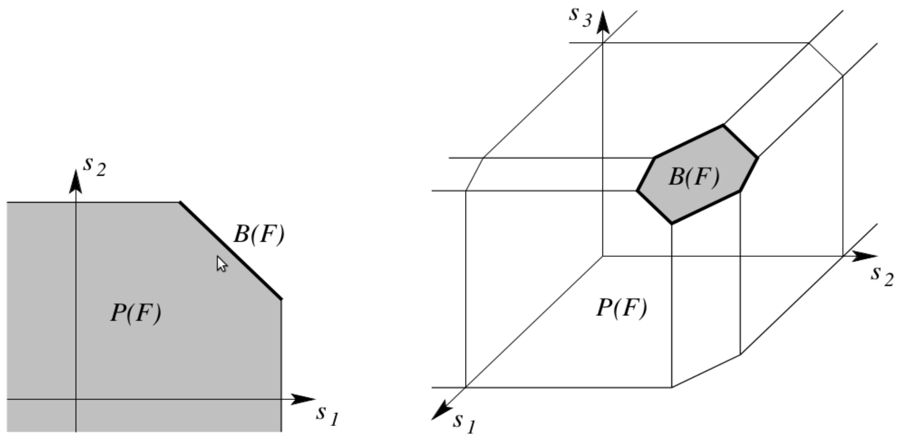

Figure 1: Submodular polyhedron P_F and base polyhedron B_F for p = 2 (left) and p = 3 (right), for a non-decreasing submodular function. Figure credit: [2]

Note that $B_F = P_F \cap \{s \in \mathcal{R}^p : F_s(I) = F(I)\}$. The submodular polyhedron is unbounded where as the base polyhedron is bounded.

2.2 Lovász Extension

Definition 2.3 - Lovász extension: Given a set function F, its Lovász extension $f : \mathcal{R}^p \to \mathcal{R}$ is defined as

$$\begin{aligned} f(d) &= \sum_{k=1}^{p} d_{j_k} \left[F(\{j_1, j_2, \ldots, j_k\}) - F(\{j_1, j_2, \ldots, j_{k-1}\})\right] \\ &= \sum_{k=1}^{p-1} F(\{j_1, j_2, \ldots, j_k\})(d_{j_k} - d_{j_{k+1}}) + F(I) d_{j_p} \\ &= \int_{\min(d_1,\ldots,d_p)}^{+\infty} F(\{d \geq z\}) dz + F(I) \min\{d_1, \ldots, d_p\} \\ &= \int_0^{+\infty} F(\{d \geq z\}) dz + \int_{-\infty}^{0} \left[F(\{d \geq z\}) - F(I)\right] dz, \end{aligned} \tag{9}$$

where $j_1, j_2, \ldots, j_p$ are indices such that $d_{j_1} \geq d_{j_2} \geq \ldots \geq d_{j_p}$.

Properties:

- If f is the Lovász extension of set function F, then $F(X) = f(\vec{1}_X)$. So, f is indeed an extension of F.

- If F and G are set functions with Lovász extensions f and g, then $\lambda_1 f + \lambda_2 g$ is the Lovász extension of $\lambda_1 F + \lambda_2 G$, $\forall \lambda_1, \lambda_2 \in \mathcal{R}$.

- If F is symmetric, then f is even.

- If $d \in \mathcal{R}_+^p$, then $f(d) = \int_0^{+\infty} F(\{d \geq z\})dz$

- f is positively homogeneous, i.e., $f(\lambda d) = \lambda f(d)$, $\forall \lambda \geq 0$.

- $f(d + \lambda \vec{1}_V) = f(d) + \lambda F(I)$.

Example 2.1: For any $s \in \mathcal{R}^p$, the Lovász extension of F_s is given by $f_s(d) = d^\top s$.

Example 2.2: Lovász extension of $F(X) = |X|$ is given by $f(d) = \sum_{k=1}^p d_k$.

Example 2.3: Given a weighted graph $G = (V, E)$ with weight matrix $W = [w_{ij}]$, the Lovász extension of cut function $F(X) = \sum_{i \in X, j \in V-X} w_{ij}$ is given by $f(d) = \sum_{i,j \in V} w_{ij} \max\{d_i - d_j, 0\}$.

Theorem 2.1: Let F be a submodular function. Let P_F be its submodular polyhedron, B_F be its base polyhedron and f be its Lovász extension. Then, we have

(i). if $d \in \mathcal{R}_+^p$, then $\max_{s \in P_F} d^\top s = f(d)$

(ii). if $d \notin \mathcal{R}_+^p$, then $\max_{s \in P_F} d^\top s = +\infty$

(iii). for any $d \in \mathcal{R}^p$, $\max_{s \in B_F} d^\top s = f(d)$

Moreover a maximizer in the case of (i) and (iii) can be obtained by the following greedy algorithm: order the components of d, as $d_{j_1} \geq d_{j_2} \geq \ldots \geq d_{j_p} \geq 0$ and define $s_{j_k} = F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\})$. Note that though $\max_{s \in P_F} d^\top s$ and $\max_{s \in B_F} d^\top s$ are linear programs with exponential number of constraints, they can be solved very efficiently.

2.3 Unconstrained submodular minimization:

Theorem 2.2 - Submodularity and convexity: A set function F is submodular if and only if its Lovśz extension f is convex.

Proof: Let $X, Y \subseteq I$. The vector $\vec{1}_X + \vec{1}_Y = \vec{1}_{X \cup Y} + \vec{1}_{X \cap Y}$ has components equal to 0 on $I - (X \cup Y)$, 1 on $(A - B) \cup (B - A)$ and 2 on $A \cap B$. Therefore, $f(\vec{1}_X + \vec{1}_Y) = f(\vec{1}_{X \cup Y} + \vec{1}_{X \cap Y}) = \int_0^2 F(\{d \geq z\})dz = \int_0^1 F(X \cup Y)dz + \int_1^2 F(X \cap Y)dz = F(X \cup Y) + F(X \cap Y)$.

Note that due to homogeneity, we have $f(\vec{1}_X + \vec{1}_Y) = f(2(\frac{1}{2}\vec{1}_X + \frac{1}{2}\vec{1}_Y)) = 2f(\frac{1}{2}\vec{1}_X + \frac{1}{2}\vec{1}_Y)$. If f is convex $f(\frac{1}{2}\vec{1}_X + \frac{1}{2}\vec{1}_Y) \leq \frac{1}{2}(f(\vec{1}_X) + f(\vec{1}_Y))$. So, we have $F(X \cup Y) + F(X \cap Y) = f(\vec{1}_X + \vec{1}_Y) \leq f(\vec{1}_X) + f(\vec{1}_Y) = F(X) + F(Y)$. Hence, if f is convex, F is submodular.

If F is submodular, then by theorem 2.1, we have $\max_{s \in B_F} d^\top s = f(d)$. Note for every s, $d^\top s$ is convex in d. Since point-wise maximum of convex functions is convex, $f(d)$ is convex.

Corollary 2.2.1: A set function F is supermodular if and only if its Lovśz extension f is concave.

Corollary 2.2.2: A set function F is modular if and only if its Lovśz extension f is linear.

Theorem 2.3 - Minimization of submodular functions: Let F be a submodular function with Lovász extension f. Then, we have

$$\min_{X \subseteq I} F(X) = \min_{d \in \{0,1\}^p} f(d) = \min_{d \in [0,1]^d} f(d). \tag{10}$$

Proof: Since $f(\vec{1}_X) = F(X)$, we have $\min_{X \subseteq I} F(X) = \min_{d \in \{0,1\}^p} f(d)$. Consider $d \in [0,1]^p$. Let $A_i \subseteq I$ be the set of indices of the i largest values of d. Then there exist $\{\lambda_i\}_{i=1}^p$ such that $\lambda_i \geq 0$, $\sum_{i=1}^p \lambda_i \leq 1$ and $d = \sum_{i=1}^p \lambda_i \vec{1}_{A_i}$. So, $f(d) = \sum_{k=1}^{p-1} F(\{j_1, j_2, \ldots, j_k\})(d_{j_k} - d_{j_{k+1}}) + F(I) d_{j_p} = \sum_{i=1}^p \lambda_i F(A_i)$ (since $d_{j_k} - d_{j_{k+1}} = \lambda_k$). Since $\sum_{i=1}^p \lambda_i F(A_i) \geq \sum_{i=1}^p \lambda_i \min_{X \subseteq I} F(X) \geq \min_{X \subseteq I} F(X)$ (since $\min_{X \subseteq I} F(X) \leq 0$ and $\sum_{i=1}^p \lambda_i \leq 1$), $\forall d \in [0,1]^p$, we have $f(d) \geq \min_{X \subseteq I} F(X) = \min_{d \in \{0,1\}^p} f(d)$. Hence, $f(d)$ attains its minimum at one of the corners of the hypercube $[0,1]^p$.

With theorems 2.1 and 2.3, unconstrained minimization of a submodular function F can be formulated as

$$\begin{aligned} \underset{c,d}{\text{minimize}} \ & c \\ \text{subject to} \ \ & c \geq d^\top s \quad \forall s \in P_F \\ & d \in [0,1]^p \end{aligned} \tag{11}$$

Though the minimization problem (11) is an LP with infinitely many constraints, it can be solved in polynomial time using ellipsoid algorithm[1]. Hence, unconstrained submodular minimization can be solved in polynomial time.

2.3.1 Minimum-norm point algorithm

Though the ellipsoid method is polynomial in time, it is very slow and is rarely used in practice. In this section we will study the minimum-norm point algorithm which is the most efficient in practice, but has no known complexity bound.

Theorem 2.4: Let F be a submodular function and B_F be its base polyhedron. Let $s^* = \text{argmin}_{s\in B_F}\|s\|_2^2$ and $\{k : s_k^* < 0\} \subseteq X^* \subseteq \{k : s_k^* \leq 0\}$. Then, $F(X^*) = \min_{X\subseteq I}F(X)$.

The minimum-norm point algorithm[4] finds $\text{argmin}_{s\in B_F}\|s\|_2^2$ using an algorithm proposed by Wolfe[5] and then computes an optimal solution to $\min_{X\subseteq I}F(X)$ by using theorem 2.4. Note that $\text{argmin}_{s\in B_F}\|s\|_2^2$ is an optimization problem with exponential number of constraints. Though the Wolfe's algorithm finds s^* in finite time, there are no known bounds on the runtime of this algorithm.

2.3.2 Combinatorial algorithms

Theorem 2.5- Dual problem: Let F be a submodular function and B_F be its base polyhedron. Then,

$$\min_{X\subseteq I} F(X) = \max_{s\in B_F} F_{s_-}(I), \tag{12}$$

where $s_- = \min\{s, 0\} \in \mathcal{R}^p$. Moreover, for any $X \subseteq I$ and $s \in B_F$, we always have $F(X) \geq F_{s_-}(I)$ with equality occurring if and only if $\{s < 0\} \subseteq X \subseteq \{s \leq 0\}$ and $F_s(X) = F(X)$. We also have,

$$\min_{X\subseteq I} F(X) = \max_{s\in P_F, s\leq 0} F_s(I). \tag{13}$$

Moreover, for any $X \subseteq I$ and $s \in P_F$ such that $s \leq 0$, we always have $F(X) \geq F_s(I)$ with equality occurring if and only if $\{s < 0\} \subseteq X$ and $F_s(X) = F(X)$.

Theorem 2.5 suggests that in order to solve $\min_{X\subseteq I} F(X)$, it is enough to find $X \subseteq I$ and $s \in B_F$ or $s \in P_F, s \leq 0$ such that the conditions in theorem 2.5 are satisfied. There are some combinatorial algorithms that output such an X and s in polynomial time[6,7].

3 References

[1] M. Grötschel, L. Lovász, A. Schrijver, "The Ellipsoid Method and its Consequences in Combinatorial Optimization", *Combinatorica* 1(2):pp. 169-197, 1981.

[2] Francis Bach, "Convex Analysis and Optimization with Submodular Functions: a Tutorial", 2010.

[3] L. Lovaśz, "Submodular Functions and Convexity", *Mathematical programming: the state of the art*, Bonn, pp.235-257, 1982.

[4] Satoru Fujishige, Takumi Hayashi and Shigueo Isotani, "The Minimum-Norm-Point Algorithm Applied to Submodular Function Minimization and Linear Programming", 2006.

[5] P. Wolfe, "Finding the Nearest Point in a Polytope", *Mathematical Programming,* 11(1):128-149, 1976.

[6] S. Iwata, L.Fleischer and S.Fujishige, "A Combinatorial Strongly Polynomial Algorithm for Minimizing Submodular Functions", *Journal of the ACM*, 48(4):761-777, 2001.

[7] J.B. Orlin, "A Faster Strongly Polynomial Time Algorithm for Submodular Function Minimization", *Mathematical Programming*, 118(2):237-251, 2009.

[8] http://submodularity.org/submodularity-slides.pdf

[9] http://j.ee.washington.edu/∼bilmes/classes/ee596a_fall_2012/.

[10] http://submodularity.org/