CS880: Approximation and Online Algorithms	**Scribe:** Eva, Xiating
Lecture 28: Submodular Functions and Their Applications	**Date:** 12/4/2019

In this lecture, we introduce submodular functions and its relationship with various combinatorial optimization problems. We will introduce and analyze (1) a greedy $(1-1/e)$-approximation algorithms to maximization problems for monotone submodular functions under cardinality constraints, and (2) a greedy $(1-1/e)$-approximation algorithms to maximization problems for monotone submodular functions under matroid constraints. For (2), we introduce the *multilinear extension*, extending the discrete submodular functions to the continuous settings and rounding the continuous solution back to a discrete one.

28.1 Submodular functions

Let S be a set of elements and consider a function $f : 2^S \to \mathbb{R}$.

Definition 28.1.1 *The function $f : 2^S \to \mathbb{R}$ is submodular if for all $A, B \subseteq S$, we have that*

$$f(A \cap B) + f(A \cup B) \le f(A) + f(B).$$

An equivalent definition of submodularity exhibits its diminishing marginal value gain over larger sets. Due to this property, the submodular functions have wide applications in game theory, operation research and machine learning.

Definition 28.1.2 *The function $f : 2^S \to \mathbb{R}$ is submodular if for all $A \subset B \subset S$ and for each $i \in S \setminus B$, we have*

$$f(A \cup \{i\}) - f(A) \ge f(B \cup \{i\}) - f(B).$$

Lemma 28.1.3 *Definition 28.1.1 is equivalent to definition 28.1.2.*

Proof: We first assume that for all $A, B \subset S$, we have

$$f(A \cap B) + f(A \cup B) \le f(A) + f(B).$$

Suppose that $A \subset B$, then for any $i \in S \setminus B$, we have that

$$\begin{aligned} f(A \cup \{i\}) + f(B) &\ge f(A \cup B \cup \{i\}) + f((A \cup \{i\}) \cap B) \\ &= f(B \cup \{i\}) + f(A), \end{aligned}$$

where the equality holds since $A \subset B$.

We now assume that

$$f(A \cup \{i\}) - f(A) \ge f(B \cup \{i\}) - f(B)$$

for each $A \subset B \subset S$ and $i \in S \setminus B$.

Consider any two sets A and B. If $A \setminus B = \emptyset$, then we have $A \subseteq B$, and thus

$$f(A \cap B) + f(A \cup B) = f(A) + f(B) \le f(A) + f(B).$$

Otherwise, let $B \setminus A = \{v_1, v_2, \dots, v_n\}$ and denote $X_i = \{v_1, v_2, \dots, v_i\}$ and $X_0 = \emptyset$. Since $(A \cap B) \cup X_i \subset A \cup X_i$ We thus have

$$f((A \cap B) \cup X_i \cup \{v_{i+1}\}) - f((A \cap B) \cup X_i) \geq f((A \cup X_i) \cup \{v_{i+1}\}) - f((A \cup X_i),$$

that is

$$f((A \cap B) \cup X_{i+1}) - f((A \cap B) \cup X_i) \geq f(A \cup X_{i+1}) - f(A \cup X_i).$$

Summing from $i = 0$ to $n - 1$, and we yield

$$f((A \cap B) \cup X_n) - f(A \cap B) \geq f(A \cup X_n) - f(A).$$

Combined with $X_n = B \setminus A$, we have

$$f(A \cap B) + f(A \cup B) \leq f(A) + f(B).$$

■

We may still define some common properties with continuous functions on submodular functions. A submodular function f is *monotone* if for all sets $A \subset B$, $f(A) \leq f(B)$. Definition 28.1.2 resembles the definition of concave functions in the continuous setting. However, maximizing certain submodular functions such as weighted coverage or mutual information, unlike concave functions, is NP-hard [4, 5], which resembles convexity.

Examples

- Coverage problem: Consider a universe U of elements and n subsets $A_1, A_2, \dots, A_n \subseteq U$. The problem is to pick at most k subsets from each A_i such that the number of elements *covered* by the union of the picked subsets is maximized. If for each index subset $S \subseteq [n]$, we define the coverage function f by $f(S) = |\bigcup_{i \in S} A_i|$, we see that the problem is precisely $\max_{S \subseteq [n]} f(S)$ subject to $|S| \leq k$. One may verify that f is submodular.

- Maximum cut: Recall that the `MAX-CUT` problem is NP-complete. Given a weighted directed graph and a nonnegative weight function $c : E \to \mathbb{R}^+$, the cut function $f(S) = c(\delta(S))$ is submodular. This is because for any vertex v, we have

  $$f(S \cup \{v\}) - f(S) = \sum_{u \in N^+(v) \setminus S} c(v, u) - \sum_{u \in S \cap N^-(v)} c(u, v),$$

  which decreases as S grows. Note that this also implies a reduction from `MAX-CUT` to submodular maximization (even with cardinality constraints), showing that the latter is NP-hard.

28.2 Maximizing submodularity under cardinality constraints

In this section we consider the monotone submodular function maximization problem under the cardinality constraint: Given a ground set S, a monotone submodular function $f : 2^S \to \mathbb{R}$ and a nonnegative integer k, maximize $f(X)$ for all $X \subseteq S$ subject to $|X| \leq k$.

One natural greedy algorithm `GREEDY` is as follows: Initialize $X_0 = \emptyset$, and at each step t, we add the element $i_t \in S \setminus X_{t-1}$ that achieves the largest marginal gain, that is,

$$i_t = \operatorname*{arg\,max}_{i \in S \setminus X_{t-1}} f(X_{t-1} \cup \{i\}) - f(X_{t-1}),$$

and let $X_t = X_{t-1} \cup \{i_t\}$, until $|X_t| = k$ or no such i_t exists.

We show that the above greedy algorithm achieves an approximation ratio of $1 - 1/e$.

Lemma 28.2.1 *`GREEDY` achieves an approximation ratio of* $1 - 1/e$*.*

Proof:

Let $X^* = \{v_1, v_2, \ldots, v_m\}$ be the optimal solution for this problem with $m \leq k$. Denote $X_j^* = \{v_i : i \leq j\}$ and $X_0^* = \emptyset$. For any step t, since f is monotone, we have that

$$\begin{aligned}
f(X^*) - f(X_{t-1}) &\leq f(X_{t-1} \cup X^*) - f(X_{t-1}) \\
&= \sum_{1 \leq j \leq m} f(X_{t-1} \cup X_j^*) - f(X_{t-1} \cup X_{j-1}^*) \\
&= \sum_{1 \leq j \leq m} f(X_{t-1} \cup X_{j-1}^* \cup \{v_j\}) - f(X_{t-1} \cup X_{j-1}^*) \\
&\leq \sum_{1 \leq j \leq m} f(X_{t-1} \cup \{v_j\}) - f(X_{t-1}) \\
&\leq \sum_{1 \leq j \leq m} f(X_{t-1} \cup \{i_t\}) - f(X_{t-1}) \\
&= \sum_{1 \leq j \leq m} f(X_t) - f(X_{t-1}) \\
&\leq k(f(X_t) - f(X_{t-1})),
\end{aligned}$$

where the first inequality holds due to monotonicity; the second inequality holds by submodularity; and the third inequality holds since our algorithm maximizes the marginal gain at step t.

Rearranging the inequality and letting $t = k$ would yield

$$\begin{aligned}
f(X^*) - f(X_k) &\leq (1 - 1/k)(f(X^*) - f(X_{k-1})) \\
&\leq (1 - 1/k)^k (f(X^*) - f(\emptyset)) \leq (1 - 1/k)^k f(X^*) \leq e^{-1} f(X^*),
\end{aligned}$$

which is equivalent to

$$f(X_k) \geq (1 - 1/e) f(X^*).$$

■

28.3 Matroids

The notion of a matroid is introduced in the previous lectures. It essentially extends the notion of linear independence in linear algebra into combinatorial structures.

Definition 28.3.1 *A matroid $\mathcal{M}$ is a pair $\mathcal{M} = (N, \mathcal{I})$ where N is a ground set of element and $\mathcal{I} \subseteq 2^N$ satisfying that*

1. *$\emptyset \in \mathcal{I}$;*
2. *If $A \in \mathcal{I}$ and $B \subseteq A$, then $B \subseteq \mathcal{I}$; and*
3. *If $A, B \in \mathcal{I}$ and $|B| > |A|$, then there exists an element $x \in B \setminus A$ such that $A \cup \{x\} \in \mathcal{I}$.*

Definition 28.3.2 *Any maximal independent set is called the* basis *of the matroid.*

A useful property that will be used later in this lecture is the **bases exchange** property, which states the following

Let $A, B \in \mathcal{B}$ where $A \neq B$, then for any element $a \in A \setminus B$ there exists an element $b \in B \setminus A$ such that $(A \setminus \{a\}) \cup \{b\} \in \mathcal{B}$. This essentially means that if in a basis we exchange an element with one from another basis, the new set is still a basis, and therefore no basis can be a subset of another.

We denote by $\mathcal{B}$ the set of bases of the matroid. The size of the maximum independent set for any set N is called the *rank* of the set N. In the following sections we will see an algorithm that maximizes a submodular function over a matroid constraint. To do this, we also define the *matroid polytope*, which is just the convex hull of the indicator vectors if the independent sets of $\mathcal{M}$. An equivalent, maybe more intuitive characterization of this polytope is the following (this is due to Edmonds [3]), which uses the rank function.

$$\mathcal{P}(\mathcal{M}) = \{x \in \mathbb{R}^n : \sum_{i \in N} x_i \leq r(N),\ \forall N \subseteq S\}$$

We denote $\mathcal{P}(\mathcal{M})$ by just $\mathcal{P}$ to simplify the notation. In what follows we consider the problem of maximizing a monotone submodular over matroid constraints, which means our feasible vectors are inside the matroid polytope. The solution to this, surprisingly however, utilizes tools in the continuous setting.

28.4 Extensions of Submodular functions

Until now we tried to directly maximize the discrete function. However, similarly to Linear Programming relaxations, it may make sense to relax the problem, solve the continuous one, and then round back to an integral solution for the initial problem. Contrary to linear programming relaxations, it is not enough to just allow for the input to take fractional values, but we need to extend the function f to these new values. There are many different relaxations or *extensions* of this function to continuous values; the convex/concave extensions, the Lovasz extension, each with different applications to different types of problems. The one we are going to define here is called the *multilinear extension.*

28.4.1 The Multilinear Extension

This extension which will be used in the algorithm discussed in the next section, where we maximize a monotone submodular function over a matroid constraint. Initially note that a function F :

$[0,1]^S \to \mathbb{R}$ is called multilinear if it is separately linear in every variable ; i.e. if we keep x_{-i} constant, then F is linear on x_i, for every i. The extension we define will "agree" with f on the integral points, but will extend f to the continuous $[0,1]^S$.

Definition 28.4.1 (Multilinear Basis function) *Let $A \subseteq S$ and $x \in [0,1]^S$. The* Multilinear Basis function *is defined as*

$$M_A(x) = \prod_{i \in A} x_i \prod_{i \notin A} (1 - x_i)$$

Observe that if we think of x_i as the probability that $x \in A$, the multilinear basis function is exactly the probability we find A.

Definition 28.4.2 (Multilinear Extension) *Let $f : 2^S \to \mathbb{R}$. The* multilinear extension *of f denoted by $F : [0,1]^S \to \mathbb{R}$ is defined as*

$$F(x) = \sum_{A \subseteq S} f(A) M_A(x)$$

Using the previous observation, $F(x)$ can be thought of as the expected value of f when each value i is drawn with probability x_i. This leads to the following equivalent definition of the multilinear extension

Definition 28.4.3 (Multilinear Extension v2) *Let $f : 2^S \to \mathbb{R}$ and for $x \in [0,1]^S$ denote by $D^i(x)$ the distribution on 2^S that picks each $v \in X$ independently with probability $x(v)$ The* multilinear extension *of f denoted by $F : [0,1]^S \to \mathbb{R}$ is defined as the expected value of f over draws from $D^i(x)$*

$$F(x) = \sum_{A \subseteq S} \mathbf{E}_{A \sim D^i(x)} \left[f(A) \right]$$

It is worth noting that the distribution we use to define F is independent in every way of the function f.

28.4.2 Useful Properties

In this section we present some important properties of the multilinear relaxation presented above, that will be useful in the proofs of the next section. The first property, formally presented in Lemma 28.4.4 is *monotonicity.* The second one, presented in Lemma 28.4.5 is *up-concavity*, meaning that the function is concave in every direction.

Lemma 28.4.4 *Let $f : 2^S \to \mathbb{R}$ be a monotone submodular function, then its multilinear extension $F : [0,1]^S \to \mathbb{R}$ is also monotone in every direction $\boldsymbol{d} \geq 0$*

Proof: We will first show that $\frac{\partial F}{\partial x_j} \geq 0$ for every coordinate j. Denote by $A_j = \{N \subseteq S : j \in N\}$, and $A'_j = \{N \subseteq S : j \notin N\}$ be the sets that contain the element j and not contain it respectively, out of all the possible subsets of the ground set. Then the partial derivative is

$$\begin{aligned}\frac{\partial F}{\partial x_j} &= \frac{\partial}{\partial x_j}\left(\sum_{N\in A_j} f(N)x_j \prod_{i\in N\setminus\{j\}} x_i \prod_{i\notin N}(1-x_j) + \sum_{N\in A'_j} f(N)(1-x_j)\prod_{i\in N} x_i \prod_{i\notin N\setminus\{j\}}(1-x_j)\right)\\ &= \left(\sum_{N\in A_j} f(N) \prod_{i\in N\setminus\{j\}} x_i \prod_{i\notin N}(1-x_j) - \sum_{N\in A'_j} f(N)\prod_{i\in N} x_i \prod_{i\notin N\setminus\{j\}}(1-x_j)\right) \geq 0\end{aligned}$$

where the last inequality holds since for every $N \in A_j$ there is a $N' \in A'_j$ such that $N = N' \cup \{j\}$, the sets differ at exactly one element, namely the j'th element, and since $N' \subseteq N$ from the monotonicity of f we get that $f(N') \leq f(N)$.

Let $\phi(\lambda) = F(\boldsymbol{x} + \lambda \boldsymbol{d})$ for some positive direction $\boldsymbol{d} \geq 0$, be the function F in this line of direction. Using the chain rule, and denoting by u_i the i'th argument of the function F we have

$$\phi'(\lambda) = \sum_{i\in S} \frac{\partial F}{\partial u_i}\frac{\partial u_i}{\partial \lambda} = \sum_{i\in S} d_i \frac{\partial F}{\partial u_i} \geq 0$$

proving that ϕ is monotone.

■

Lemma 28.4.5 *Let $f : 2^S \to \mathbb{R}$ be a monotone submodular function, then its multilinear extension $F : [0,1]^S \to \mathbb{R}$ also concave in every direction $\boldsymbol{d} \geq 0$.*

Proof: First we show that $\partial^2 F/\partial x_i \partial x_j \leq 0$ for all $i, j \in S$. To simplify the notation, denote by $M_N^{j,u}(\boldsymbol{x}) = \prod_{i\in N\setminus\{j,u\}} x_i \prod_{i\in N\setminus\{j,u\}}(1-x_i)$. Following the proof for $\partial F/\partial x_j$, we end up with the following expression

$$\begin{aligned}\frac{\partial F}{\partial x_i x_j} = \sum_{N\in A_i\cap A_j} f(N)M_N^{j,i}(\boldsymbol{x}) - \sum_{N\in A_i\cap A'_j} f(N)M_N^{j,i}(\boldsymbol{x}) - \sum_{N\in A'_i\cap A_j} f(N)M_N^{j,i}(\boldsymbol{x})\\ + \sum_{N\in A'_i\cap A'_j} f(N)M_N^{j,i}(\boldsymbol{x}) \leq 0\end{aligned}$$

where for the last inequality we use the same argument as before, but instead of using the monotonicity of f we use the submodularity from Definition 28.1.1 where we set $A = A_i \cap A_j$ and $B = A'_i \cap A'_j$. Now, using the expression for $\phi'(\lambda)$ we calculated in the previous lemma we now find

$$\phi''(\lambda) = \sum_{i\in S} d_i \sum_{j\in S} d_j \frac{\partial^2 F}{\partial u_i \partial u_j} \leq 0$$

which proves the lemma. ■

Lemma 28.4.6 *Let $f : 2^S \to \mathbb{R}$ be a monotone submodular function, then its multilinear extension $F : [0,1]^S \to \mathbb{R}$ also concave in every direction $\boldsymbol{e}_i - \boldsymbol{e}_j$ for any $i, j \in S$.*

Proof: We define $\phi(\lambda) = F(\boldsymbol{x} + \lambda(\boldsymbol{e}_i - \boldsymbol{e}_j))$, and using the chain rule we get that

$$\phi''(\lambda) = \frac{\partial^2 F}{\partial x_i^2} - 2\frac{\partial^2 F}{\partial x_i \partial x_j} + \frac{\partial^2 F}{\partial x_j^2} \leq 0$$

which follows similarly to the previous lemma. ■

28.4.3 Evaluation

It is important to note at this point, that evaluating the multilinear extension exactly requires that we ask the value of f exponential number of times. In order to avoid this, we evaluate $F(x)$ approximately using a number or randomly sampled sets, to get as close as we need to the real value of $F(x)$ with high probability.

Specifically, let $R_1, R_2, \dots R_t$ be random sets $R_i \subseteq S$ where every $i \in S$ appears in R_j independently with probability x_i. Then we can estimate the value of $F(\boldsymbol{x})$ using the estimator $1/t \sum_{i=1}^{t} f(R_i)$, and be close to the real $F(\boldsymbol{x})$ with high probability.

Lemma 28.4.7

$$\left| \frac{1}{t} \sum_{i=1}^{t} f(R_i) - F(\boldsymbol{x}) \right| \leq \epsilon \max_S |f(S)|$$

with probability at least $1 - \exp(t\epsilon^2/2)$.

Proof: Denote by $M = \max_S |f(S)|$ and by $X_i = f(R_i) \in [-M, M]$ the random variables that correspond to the value of f for the randomly sampled sets R_i. Denote by $\overline{X} = 1/t \sum_{i=1}^{t} X_i$ the estimated mean of the variables. Using the Hoeffding inequality we have that

$$\Pr[|\overline{X} - F(\boldsymbol{x})| \geq \epsilon M] \leq 2\exp\left(-\frac{2t^2\epsilon^2 M^2}{4tM^2}\right) = 2\exp\left(-\frac{t\epsilon^2}{2}\right)$$

■

28.5 Continuous Submodular Maximization

Now that we have defined the extension and describes its properties, we describe the algorithm to optimize it. Denote by $\mathcal{I}$ the set of feasible subsets of 2^S ; this set is extended to a polytope $\mathcal{P}$ in $[0,1]^S$, that contains the characteristic vectors of the sets in $\mathcal{I}$. The initial problem and the multilinear relaxation are shown in Table 1.

maximize	$f(A)$	maximize	$F(x)$
subject to	$A \in \mathcal{I} \subseteq 2^S$	subject to	$x \in \mathcal{P} \subseteq [0,1]^S$

Table 1: Initial problem (left) and relaxation (right)

The continuous version of submodular maximization is actually very similar to the greedy discrete algorithm that is described before, and it was first described by Vondrak in [6]. In this continuous case, what the algorithm does is to find the direction $\boldsymbol{v}$, in which the derivative is larger. Since the function F is multilinear, it derivative in every direction will just be the difference between any two values of the function (in this direction) divided by their difference. This algorithm, formally described in 1, also gives a $1-1/e$-approximation guarantee as shown in Theorem 28.5.1.

Algorithm 1: Continuous Greedy Algorithm

Data: Polytope $\mathcal{P}$, multilinear extension F

$\boldsymbol{x}(0) = \mathbf{0}$
foreach $t \in [0,1]$ **do**
  $\boldsymbol{v}_{max}(\boldsymbol{x}(t)) = argmax_{v\in\mathcal{P}}\langle \boldsymbol{v}, \nabla F(\boldsymbol{x})\rangle$
  Increase $x(t)$ at rate $\boldsymbol{v}_{max}(\boldsymbol{x}(t))$
end
return $x(1)$

The main theorem shows that we loose the $1-1/e$ factor when we try to optimize the continuous $F(\boldsymbol{x})$ (Theorem 28.5.1). After this, we can round this continuous solution to a discrete one without loosing anything in terms of approximation, as shown in Theorem 28.5.3.

Theorem 28.5.1 *Let $f : 2^S \to \mathbb{R}^+$ be any mononone submodular function, let $F : [0,1]^S \to \mathbb{R}$ be its multilinear extension, let $\mathcal{P} \subseteq [0,1]^S$ be a polytope, algorithm 1 finds a point $\boldsymbol{x}(1) \in \mathcal{P}$ such that*

$$F(\boldsymbol{x}(1)) \geq (1-1/e)OPT$$

Before proving Theorem 28.5.1 we show that there exists a direction of "movement" that makes substantial progress. This is formalized in the following lemma. Note also that the solution the algorithm returns is a feasible point inside the polytope. To see this, think that the value $x(1)$ is essentially the sum of all the small "steps" we took in the direction of the derivative, so $x(1) = \int_0^1 v_{\max}(x(t))dt$. Since the algorithm stops at time 1, and each times moves in the direction of an independent set, this is a convex combination of points inside $\mathcal{P}$. Having said this, we proceed to prove the lemma and the theorem.

Lemma 28.5.2 *Let $f : 2^S \to \mathbb{R}^+$ be any monotone submodular function, let $F : [0,1]^S \to \mathbb{R}$ be its multilinear extension, let $\mathcal{P} \subseteq [0,1]^S$ be a polytope, $\boldsymbol{x} \in [0,1]^S$ be an arbitrary point and $OPT = \max_{\boldsymbol{y}\in\mathcal{P}}\{F(\boldsymbol{y})\}$. Then, there exists a feasible point $\boldsymbol{v} \in \mathcal{P}$ such that*

$$\langle \boldsymbol{v}, \nabla F(\boldsymbol{x})\rangle \geq OPT - F(\boldsymbol{x})$$

Proof:

Denote by $\boldsymbol{d}$ the direction $\boldsymbol{d} = (\boldsymbol{v}-\boldsymbol{x}) \vee \mathbf{0}$, where $\vee$ is the pointwise maximum of the two vectors i.e. $(\boldsymbol{a}\vee\boldsymbol{b})_i = \max\{a_i, b_i\}$. Note that the direction vector $\boldsymbol{d}$ satisfies

$$\boldsymbol{v} \geq \boldsymbol{d} \geq \mathbf{0} \tag{28.5.1}$$

Let $\phi(\lambda)$ be the objective function along the direction $\boldsymbol{d}$: $\phi(\lambda) = F(\boldsymbol{x} + \lambda \boldsymbol{d})$. From the properties of F we know that it is concave in every direction, which means that $\phi(\lambda)$ is also concave. From this fact we get that

$$\phi(1) - \phi(0) \leq \phi'(0) \tag{28.5.2}$$

using the equation above, and substituting the values of ϕ we get

$$F(\boldsymbol{x} + \boldsymbol{d}) - F(\boldsymbol{x}) \leq \langle \boldsymbol{d}, \nabla F(\boldsymbol{x}) \rangle \leq \langle \boldsymbol{v}, \nabla F(\boldsymbol{x}) \rangle$$

where for the last inequality we used inequality (28.5.1). Observe now that $\boldsymbol{x}+\boldsymbol{d} \geq \boldsymbol{x}+((\boldsymbol{v}-\boldsymbol{x}) \vee \boldsymbol{0}) \geq \boldsymbol{v}$ (this is immediate from the definition of $\vee$). Since F is non-decreasing in every direction, we get that $F(\boldsymbol{x} + \boldsymbol{d}) \geq F(\boldsymbol{v})$. Now we choose $\boldsymbol{v}$ such that $F(\boldsymbol{v}) = \text{OPT}$, to get the desired result

$$\langle \boldsymbol{v}, \nabla F(\boldsymbol{x}) \rangle \geq \text{OPT} - F(\boldsymbol{x})$$

■

Proof of Theorem 28.5.1:

Initially, we apply the chain rule on $F(\boldsymbol{x}(t))$ to obtain a relation with the directional derivative $\langle \boldsymbol{v}, \nabla F(\boldsymbol{x}) \rangle$ we had in the previous lemma, to reach a differential equation.

$$\frac{dF(\boldsymbol{x}(t))}{dt} = \sum_{i \in S} \frac{d\boldsymbol{x}(t)}{dt} \cdot D_i F = \left\langle \frac{d\boldsymbol{x}(t)}{dt}, \nabla F(\boldsymbol{x}) \right\rangle = \langle \boldsymbol{v}_{max}, \nabla F(\boldsymbol{x}) \rangle \geq \langle \boldsymbol{v}, \nabla F(\boldsymbol{x}) \rangle \tag{28.5.3}$$

where in the last equality we used the definition of $d\boldsymbol{x}(t)/dt$ and in the last inequality the fact that we choose v_{max} as the maximum over every $\boldsymbol{v} \in \mathcal{P}$. By using Lemma 28.5.2 and inequality (28.5.3) we get

$$\frac{dF(\boldsymbol{x}(t))}{dt} + F(\boldsymbol{x}(t)) \geq \text{OPT}$$

in order to solve this differential equation we multiply everywhere by e^t, and using the monotonicity of the integral we get

$$e^t F(\boldsymbol{x}(t)) \geq \int_0^t e^y \text{OPT} dy = (e^t - 1)\text{OPT}$$

This gives us $F(\boldsymbol{x}(t)) \geq (1 - e^{-t})\text{OPT}$, which for the time $t = 1$ yields the desired result

$$F(\boldsymbol{x}(t)) \geq (1 - 1/e)\text{OPT}$$

■

Note that in order to implement this continuous algorithm we need to discretize the process in the following way: we fix an increment δ and run the algorithm for N steps each time incrementing by $x(t + \delta) = x(t) + \delta \boldsymbol{v}_{\text{max}}$.

Now that we have found a solution $x_0 \in [0,1]^S$ we need to use this to find a discrete solution $x_d \in 2^S$ to our initial submodular maximization problem; this is formalized in Theorem 28.5.3. The rounding presented here is *swap rounding* and was first introduced in [2].

Theorem 28.5.3 *Let $f : 2^S \to \mathbb{R}^+$ be a submodular function, let $F : [0,1]^S \to \mathbb{R}$ be its multilinear extension, a matroid $\mathcal{M} = (N, \mathcal{I})$ and a point $x_0 \in \mathcal{P}(\mathcal{M})$ there is an algorithm that outputs an independent set $S \in \mathcal{I}$ such that $f(S) \geq F(x_0)$.*

Proof: Since the starting point $x(1)$ that is returned by the algorithm is inside the matroid polytope, it can be written as a convex combination of the bases vectors χ_{B_i} where B_i is the basis i. Therefore $x(1) = \sum_i \beta_i \chi_{B_i}$ such that $\sum_i \beta_i = 1$ and $b_i \geq 0$ for every i.

We round the solution in $n-1$ phases, every time merging two bases, using Algorithm 2. This algorithm essentially takes two bases and using the bases exchange property, defined in section 28.3, many times iteratively, manages to make the bases the same.

We show that in every step of this algorithm, the expected value for the new vector $\boldsymbol{x'}$ is at least the old value by exploiting the fact that F is convex in the direction of $e_i - e_j$. Denote by $Y = \sum_{i \notin \{1,2\}} \beta_i \chi_i$ be the sum of the bases that do not participate in the algorithm. From the definition of the algorithm we have that

$$\begin{aligned}
\mathbf{E}_{\boldsymbol{x'} \sim Alg}\left[F(x')\right] &= \frac{\beta_1}{\beta_1+\beta_2} F(\beta_1\chi_1 + \beta_2(\chi_2 + e_i - e_j) + Y) + \frac{\beta_2}{\beta_1+\beta_2} F(\beta_2\chi_2 + \beta_1(\chi_1 + e_j - e_i) + Y) \\
&= \frac{\beta_1}{\beta_1+\beta_2} F(\boldsymbol{x} + \beta_2(e_i - e_j)) + \frac{\beta_2}{\beta_1+\beta_2} F(\boldsymbol{x} + \beta_1(e_j - e_i) + Y) \\
&\geq F\left(\frac{\beta_1}{\beta_1+\beta_2}(\boldsymbol{x} + \beta_2(e_i - e_j)) + \frac{\beta_2}{\beta_1+\beta_2}(\boldsymbol{x} - \beta_1(e_i - e_j))\right) \\
&= F(\boldsymbol{x})
\end{aligned}$$

where the inequality is from the convexity of F in the direction $e_i - e_j$.

Algorithm 2: Merge Bases

Input: β_1, β_2, Bases B_1, B_2
Output: Base B

```
while B_1 ≠ B_2 do
    Find i ∈ B_1 and then find j ∈ B_2 such that (B_1 \ {i}) ∪ {j} ∈ I and
      (B_2 \ {j}) ∪ {i}) ∈ I
    With probability β_1/(β_1+β_2) do B_2 = B_2 \ {j} ∪ {i}
    Else B_1 = B_1 \ {i} ∪ {j}
end
return B_1
```

■

One alternative rounding technique, similar to the one presented before is *pipage rounding*, first introduced in [1].

Pipage Rounding This technique exploits the fact that $F(\boldsymbol{x})$ is convex in every direction $\boldsymbol{e}_i - \boldsymbol{e}_j$, to move from a fractional to an integral point while achieving two things: not decreasing the value of F and moving closer to an integer vector $\boldsymbol{x}$. More specifically, since we know that the function $F(\boldsymbol{x})$ is convex in every direction, the idea is to find a direction $\boldsymbol{d}$ such that if we move either towards $\alpha\boldsymbol{d}$ or towards $-\beta\boldsymbol{d}$ (for some $\alpha, \beta > 0$) the new vector $\boldsymbol{x}'$ will have strictly more integer coordinates, and the value will not be decreased. Since F is convex in this direction, in one of the two directions F should not decrease. Figure 28.5.1 demonstrates this idea pictorially. This is essentially the deterministic version of the rounding presented before.

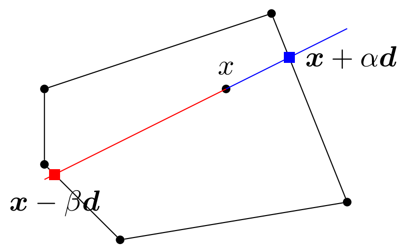

Figure 28.5.1: Rounding in direction $\boldsymbol{d}$

Another alternative technique for rounding, that is also worth mentioning is *contention resolution schemes*, which applies to any downward-closed body P, and not necessarily a matroid polytope, and was first introduced by Chekuri et al. in [7].

References

[1] Alexander A. Ageev and Maxim Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Comb. Optim.*, 8(3):307–328, 2004.

[2] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 575–584, 2010.

[3] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization - Eureka, You Shrink!, Papers Dedicated to Jack Edmonds, 5th International Workshop, Aussois, France, March 5-9, 2001, Revised Papers*, pages 11–26, 2001.

[4] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[5] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pages 324–331, Arlington, Virginia, United States, 2005. AUAI Press.

[6] Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 67–74, 2008.

[7] Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 783–792, 2011.