UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga
(Final project paper)

Submodularne funkcije

(Submodular functions)

Ime in priimek: Edin Husić
Študijski program: Matematika
Mentor: izr. prof. dr. Martin Milanič

Koper, september 2015

Ključna dokumentacijska informacija

Ime in PRIIMEK: Edin HUSIĆ

Naslov zaključne naloge: Submodularne funkcije

Kraj: Koper

Leto: 2015

Število listov: 35 Število slik: 1 Število tabel: 2

Število referenc: 34

Mentor: izr. prof. dr. Martin Milanič

Ključne besede: kombinatorična optimizacija, submodularna funkcija, maksimizacija submodularne funkcije, aproksimacijski algoritem.

Math. Subj. Class. (2010): 68W25, 90C27, 05C65

Izvleček:
V zaključni nalogi je opisan pojem submodularne funkcije in z njim povezani optimizacijski problemi, vključno s problemi minimizacije submodularnih funkcij, maksimizacije submodularnih funkcij in submodularnega pokritja. Opisani in analizirani so izbrani aproksimacijski algoritmi za maksimizacijo submodularnih funkcij, in sicer požrešna metoda za maksimizacijo monotonih submodularnih funkcij ter trije aproksimacijski algoritmi za maksimizacijo submodularnih funkcij v splošnem s faktorji aproksimacije 1/4, 1/3 in 1/2. Dva izmed teh treh algoritmov sta verjetnostna, tretji je deterministični. V nalogi je podan tudi nov dokaz izreka, da za vsak hipergraf obstaja eksakten polimatroidni separator.

Key words documentation

Name and SURNAME: Edin HUSIĆ

Title of final project paper: Submodular functions

Place: Koper

Year: 2015

Number of pages: 35 Number of figures: 1 Number of tables: 2

Number of references: 34

Mentor: Assoc. Prof. Martin Milanič, PhD

Keywords: combinatorial optimization, submodular function, submodular function maximization, approximation algorithm.

Math. Subj. Class. (2010): 68W25, 90C27, 05C65

Abstract:
In the final project paper the notion of submodular functions is described. Several examples of optimization problems with submodular objective functions are given, together with the problem of minimization and maximization of submodular functions and the submodular set cover problem. Description and analysis of approximation algorithms for the problem of maximizing a given submodular function under a cardinality constraint and for the unconstrained submodular maximization problem are given. More precisely, we present a greedy algorithm for maximizing a submodular function under a cardinality constraint, one deterministic algorithm and two randomized algorithms with approximation factors 1/3, 1/4 and 1/2, respectively, for solving the uncostrained submodular maximization problem. A new proof of the theorem stating that every hypergraph has an exact polymatroid separator is also given.

Acknowledgement

Most of all I would like to thank my mentor Prof. Martin Milanič who guided me with his advices through the final project paper. I would like to express my sincere gratitude for his patience, motivation, knowledge and time devoted to help me. I am also thankful to him for encouraging the use of correct grammar and consistent notation in my writings. I was very fortunate to have Him as an advisor.

I would like to acknowledge Prof. Vladimir Gurvich for pointing out useful references that gave me insight in polymatroid functions. I am also thankful to Prof. Endre Boros for remarks related to Theorem 2.11 and for commenting on my views.

I express my gratitude to UP FAMNIT for giving me so many opportunities to learn, study, and explore everything and everywhere.

Special thank goes to my family for supporting me in every possible way and always being there for me when I need them. Thank you mom, dad, and brother.

Contents

List of Tables

List of Figures

List of Abbreviations

i.e. that is
e.g. for example
SFM submodular function minimization
USM unconstrained submodular maximization

1 Introduction

Submodularity is a property of set functions, that is, functions $f : 2^V \to \mathbb{R}$ that assign to each subset $S \subseteq V$ a real value $f(S)$. They arise in many different ares like economy and game theory as utility functions on indivisible goods – discrete items that can be traded only as a whole. Submodularity appears in machine learning and generalizes many important problems in optimization such as maximum cut in directed and undirected graphs and hypergraphs, see Section 2.5. Maximization of submodular function also generalizes the problem of maximum facility location and other constraint satisfaction problems, as we will see in Section 2. The biggest part of this final project paper is devoted to the problem of maximizing sumbodular functions [7, 9, 21]. Several different maximization problems and approximation algorithms for solving them are analysed in Section 4.2. Section 3 offers a short overview of operations that preserve modularity.

Two Fulkerson prizes were awarded for results concerning submodular functions minimization, see [22]. In 1982, the Fulkerson prize was awarded to D.B. Judin, Arkadi Nemirovski, Leonid Khachiyan, Martin Grötschel, László Lovász and Alexander Schrijver for their work on the ellipsoid method. As will be stated in Section 4.1, this proved that submodular minimization is possible in polynomial time. In 2003, Satoru Iwata, Lisa Fleischer, Satoru Fujishige, and Alexander Schrijver recived the Fulkerson prize for showing that submodular minimization is strongly polynomial. Besides that, in Section 2.9 special subclass of submodular functions, namely polymatroid functions, will be presented, which have many applications in graph theory [3]. I observed that polymatroid functions can be used to capture the notion of independence in hypergraphs, see Theorem 2.11 and the remarks following it.

Submodularity can be viewed as discrete analogue of concavity considering derivatives, and has some properties similar to convexity such as allowing the possibility for efficiently finding the minimum function value.

Definition 1.1. A set function on a finite set V,[1] $f : 2^V \to \mathbb{R}$, is *submodular* if for every $A, B \subseteq V$ we have

[1] Usually we will denote the power set of a set V with 2^V.

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

Further on we will always assume that V, also called ground set, is finite. A more intuitive definition of submodularity can be stated using discrete derivatives.

Definition 1.2. Let $f : 2^V \to \mathbb{R}$ be a set function. For a subset $S \subseteq V$ and $v \in V$, we define

$$\triangle_f(v \mid S) = f(S \cup \{v\}) - f(S)$$

and we say that $\triangle_f(v \mid S)$ is the *discrete derivative* of f at S with respect to v.

Discrete derivative is sometimes called *marginal gain* and exhibits a natural diminishing return property. A diminishing discrete derivative reminds us of a decreasing first derivative of differentiable functions. It is well known that a differentiable function f is concave on an interval if its derivative function f' is monotonically non-increasing. The following two results make it clear why we call submodularity a discrete analogue of concavity.

Proposition 1.3. *A set function $f : 2^V \to \mathbb{R}$ is submodular if and only if*

$$f(S \cup \{s\}) + f(S \cup \{t\}) \geq f(S) + f(S \cup \{s, t\}) \tag{1.1}$$

for each $S \subseteq V$ and distinct $s, t \in V \setminus S$.

Proof. Necessity is trivial by applying the definition of submodularity with $A = S \cup \{s\}$ and $B = S \cup \{t\}$. We will prove sufficiency by induction on $|A \triangle B|$, where $A \triangle B := (A \setminus B) \cup (B \setminus A)$. Let $A, B \subseteq V$. First, suppose $|A \triangle B| \leq 2$. Then we have either $A \subseteq B$, $B \subseteq A$ or $A \setminus B = \{s\}$ and $B \setminus A = \{t\}$ in which case the statement is true by (1.1). We set the inductive hypothesis: if $n \geq 3$ and we have two sets $X, Y \subseteq V$ such that $|X \triangle Y| < n$, then they satisfy: $f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y)$. Suppose $|A \triangle B| = n$, we may assume by symmetry that $|A \setminus B| \geq 2$. Choose $t \in A \setminus B$. Notice that $|((A \setminus \{t\}) \cup B) \triangle A| < n$ and $|(A \setminus \{t\}) \triangle B| < n$. Then, from the inductive hypothesis applied to $(X, Y) = ((A \setminus \{t\}) \cup B, A)$ and to $(X, Y) = (A \setminus \{t\}, B)$, respectively, we derive

$$\begin{aligned} f(A \cup B) + f(A \setminus \{t\}) &\leq f((A \setminus \{t\}) \cup B) + f(A)\,, \\ f((A \setminus \{t\}) \cup B) + f(A \cap B) &\leq f(B) + f(A \setminus \{t\})\,. \end{aligned}$$

The submodularity condition for (A, B) follows immediately by summing up the above two inequalities. □

Rewriting the same condition with discrete derivatives, we can state that f is submodular if and only if $\triangle_f(s \mid S) \geq \triangle_f(s \mid S \cup \{t\})$. We can easily extend this to following theorem.

Theorem 1.4. *A function $f : 2^V \to \mathbb{R}$ is submodular if and only if for every $A \subseteq B \subseteq V$ and $v \in V \setminus B$ it holds that $\triangle_f(v \mid A) \geq \triangle_f(v \mid B)$.*

Necessity is again trivial, while sufficiency is easily proved with Proposition 1.3.

Similarly we can define the notion of *supermodular functions*, just turning the inequality in (1.1). It is clear that if a function g is supermodular, then function $-g$ is submodular, and vice versa. A corresponding equivalence with discrete derivatives also holds. This way all properties of submodular functions can be translated to supermodular functions, and vice versa.

Definition 1.5. A function $f : 2^V \to \mathbb{R}$ is *monotone increasing* or just *monotone*, if for every $A \subseteq B \subseteq V$, $f(A) \leq f(B)$.

We say that a function f is *monotone decreasing* if $-f$ is monotone increasing. A simple example of a monotone submodular function f can be seen on a two-element set $\{a, b\}$, defined by $f(\{a, b\}) = 3$, $f(\{a\}) = f(\{b\}) = 2$ and $f(\emptyset) = 0$.

Monotonicity as one of very simple and elementary properties of set functions also has its interpretation in terms of discrete derivatives.

Proposition 1.6. *A function f is monotone if and only if all its discrete derivatives are nonnegative.*

Proof. Suppose f is monotone. Then for $A \subseteq V$ and $v \in V \setminus A$ it holds $\triangle_f(v \mid A) = f(A \cup \{v\}) - f(A) \geq 0$ by monotonicity. This proves one implication. For the other implication, suppose all discrete derivatives are nonnegative. Let $A \subseteq B \subseteq V$ and $v \in V \setminus A$, then by assumption we have $\triangle_f(v \mid A) \geq 0 \Rightarrow f(A \cup \{v\}) \geq f(A)$. Let $B \setminus A = \{v_1, v_2, \ldots v_k\}$. Applying the inequality $f(A \cup \{v\}) \geq f(A)$ k times, we get $f(B) = f(A \cup \{v_1, v_2, \ldots, v_k\}) \geq f(A \cup \{v_1, v_2, \ldots, v_{k-1}\}) \geq \cdots \geq f(A)$. (A formal proof would proceed by induction on k.) □

1.1 Probability preliminaries

For analysing randomized algorithms, Algorithm 2 and Algorithm 4 in Section 4 we need a few preliminaries about expected value. A precise definition of a random variable will not be needed. Let us just mention that a random variable is a variable whose value is subject to variations due to chance. A random variable is *discrete* if it can take only a finitely or countably many different values. For our purpose we need just the case where a discrete random variable can take finitely many values. Precise definitions and proofs of propositions can be found in [14].

Definition 1.7. Let X be a discrete random variable which can take values $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}$. We define the *expected value* of X as

$$\mathbb{E}[X] = \sum_{i=1}^{n} x_i P(X = x_i)$$

where $P(X = x_i)$ denotes the probability that random variable X takes value x_i.

Proposition 1.8. *The expected value is linear, i.e., if X, Y are random variables taking values in $\mathbb{R}$ and $\alpha, \beta \in \mathbb{R}$ then*

$$\mathbb{E}[\alpha X + \beta Y] = \alpha \mathbb{E}[X] + \beta \mathbb{E}[Y]\,.$$

Proposition 1.9. *Let X, Y be a random variables such that X takes values in $\mathbb{R}$. Then the tower property holds, i.e.,*

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X|Y]]\,.$$

Proposition 1.10. *The formula for total expected value holds, i.e., given random variables X, Y which take values in a subset of $\mathbb{R}$ and in $\{y_1, \ldots, y_k\}$, respectively, we have*

$$\mathbb{E}[X] = \sum_{i=1}^{k} P(Y = y_i)\mathbb{E}[X|Y = y_i]\,.$$

Suppose we given two random variables A and B. The last proposition allows us to prove $\mathbb{E}[A] * \mathbb{E}[B]$ where $* \in \{=, <, >, \leq, \geq\}$ by proving $\mathbb{E}[A|Y = y_i] * \mathbb{E}[B|Y = y_i]$ for some arbitrary y_i and a discrete random variable Y. This will be used in Lemma 4.9 under the term *unconditioning*.

2 Examples

In this section we will see several examples of submodular functions and optimization problems with submodular objective functions.

2.1 Modular functions and generalizations

If we replace inequality in the definition of submodular functions by equality we get the so-called *modular functions*, i.e., functions $f : 2^V \to \mathbb{R}$ such that for all $A, B \subseteq V$ it holds $f(A) + f(B) = f(A \cup B) + f(A \cap B)$. Equivalently $\triangle_f(v \mid A) = \triangle_f(v \mid B)$ for all A, B and $v \in V \setminus (A \cup B)$. Modular functions are a discrete analogue of linear functions, since their discrete derivatives are constant.

Assume $f(\emptyset) = 0$. Then $\triangle_f(v \mid A) = \triangle_f(v \mid \emptyset) = f(\{v\})$ or $f(A \cup \{v\}) = f(A) + f(\{v\})$ for all $A \subset V$ and $v \in V \setminus A$. This gives us a representation of modular functions in the form $f(S) = \sum_{v \in S} w(v)$ for some weight function $w : V \to \mathbb{R}$.

Another source of examples of submodular functions is given by the composition of any monotone modular function and any concave function. Recall that a function $h : [a, b] \to \mathbb{R}$ is *concave* if for every two points $x, y \in [a, b]$ and any $\lambda \in [0, 1]$, it holds

$$h(\lambda x + (1 - \lambda)y) \geq \lambda h(x) + (1 - \lambda)h(y)\,.$$

If h has a second derivative on the interval $[a, b]$, then the equivalent condition for concavity is that the second derivative is non-positive, i.e., $h''(x) \leq 0\ \forall x \in [a, b]$. This means that the function h' is monotonically non-increasing on the interval $[a, b]$.

Proposition 2.1. *The composition $g = h \circ f$ of a monotone modular function $f : 2^V \to \mathbb{R}$ and a concave function $h : \mathbb{R} \to \mathbb{R}$ is submodular.*

Proof. Let $A \subseteq B \subseteq V$ and $v \in V \setminus B$. Then

$$\begin{aligned}
\triangle_g(v \mid A) &= h(f(A \cup \{v\})) - h(f(A)) \\
&= h(f(A) + f(\{v\})) - h(f(A)) \text{ (by modularity)} \\
&\geq h(f(B) + f(\{v\})) - h(f(B)) \text{ (since } f \text{ is monotone and } h \text{ is concave)} \\
&= h(f(B \cup \{v\})) - h(f(B)) = \triangle_g(v \mid B)\,.
\end{aligned}$$

□

2.2 Diameter of edge-induced subforest

We denote the distance between two vertices u and v in a connected graph G, i.e., the minimum length of a path connecting them, with $d(u, v)$. Furthermore, we define the *diameter* of a graph $G = (V, E)$ to be the length of a longest shortest path. Formally, $diam(G) := max_{u,v \in V} d(u, v)$.

Let $G = (V, E)$ be a forest (a graph without cycles). Then, the function f defined for each subset of edges $X \subseteq E$ as

$$f(X) := \sum_{K} diam(K)\,,$$

where K ranges over all components of the graph (V, X), is submodular [30].

2.3 Facility location

Suppose we are given n locations where we can open a few facilities in order to serve a set of m customers. If we open a facility at some location j, among all locations $V = \{1, 2, \ldots, n\}$, then we say it provides a service of value $M_{i,j}$ to customer i. This way we obtain a matrix $M \in \mathbb{R}^{m \times n}$. If all customers are rational and each of them chooses a facility that provides her/him the highest value, then the total value of services provided by non-empty set $S \subseteq V$ is

$$f(S) = \sum_{i=1}^{m} \max_{j \in S} M_{i,j}\,.$$

We also set $f(\emptyset) = 0$. If $\forall i, j$ it holds that $M_{i,j}$ is non-negative, then $f(S)$ is monotone submodular [11].

We now state one similar problem. Suppose that a function $c : V \to \mathbb{R}^+$ represents the cost of opening a facility at certain location. The function $g : 2^V \to \mathbb{R}$, defined as the sum of total value of services provided by a set $S \subseteq V$ plus the cost of facilities we did not open, i.e.,

$$g(S) = f(S) + c(V \setminus S)$$

is non-negative submodular.[1] The function g represents the gain obtained by opening facilities at locations S plus the remainder of budget that could be spent compared to the case if we would open facilities on every location. The difference between f and g arises when we want to maximize them. Since f is monotone, maximizing it without any constraint is obvious $\max_{S \subseteq V} f(S) = f(V)$. Usually we want to maximize f, demanding some cardinality constraint, and we try to maximize g without any constraint. (See Section 4.) Facility location models are general and can be used for other problems as well.

[1] $c(V \setminus S) := \sum_{i \in V \setminus S} c(i)$.

2.4 k-dimensional volume of parallelepiped

Let $A = \{a_1, a_2, \ldots, a_n\} \subset \mathbb{R}^n$ be a set of linearly independent vectors, and let for $X \subseteq A$, $X = \{a_{i_1}, a_{i_2}, \ldots, a_{i_k}\}$,

$$f(X) = \log\, vol_k(a_{i_1}, a_{i_2}, \ldots, a_{i_k})$$

where vol_k denotes the volume of the parallelepiped spanned by vectors in X. Then

$$\begin{aligned}\triangle_f(a \mid X) &= \log\, vol_{k+1}(a_{i_1}, a_{i_2}, \ldots, a_{i_k}, a) - \log\, vol_k(a_{i_1}, a_{i_2}, \ldots, a_{i_k}) \\ &= \log \frac{vol_{k+1}(a_{i_1}, a_{i_2}, \ldots, a_{i_k}, a)}{vol_k(a_{i_1}, a_{i_2}, \ldots, a_{i_k})}\end{aligned}$$

is the logarithm of the height of a on the plane spanned by X, so the discrete derivative definiton of submodularity holds.

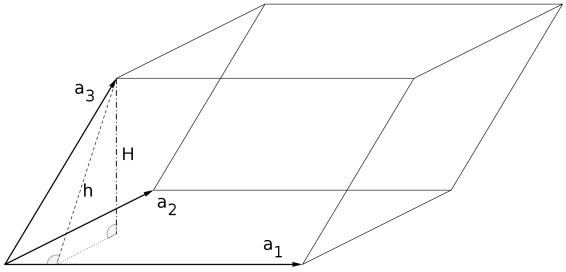

Figure 1: Example for $k = 2$.

Example for $k = 2$ can be seen in Figure 1. Following the definition of f, it is clear that $h = \triangle_f(a_3 \mid \{a_1\})$ and $H = \triangle_f(a_3 \mid \{a_1, a_2\})$. Intuitively we can see how one can verify submodularity using the discrete derivatives ($H \leq h$).

2.5 Cuts

The very important problems of finding a minimum and a maximum cut in a given directed or undirected graph or even hypergraph can be translated to problems of optimizing a derived submodular function [8, 33]. Let $G = (V, E)$ be a directed graph with a non-negative capacity function $c : E \to \mathbb{R}^+$ and with two distinguished vertices, source s and sink t. For a subset $S \subseteq V$ containing the source s but not containing the sink t we define $\delta^+(S)$ to be *the set of edges leaving set* S, $\delta^+(S) := \{(v, x) \in E : v \in S, x \notin S\}$. The *cut capacity* of S, denoted by $f(S)$, is simply the sum of capacities

of all edges that leave S, i.e.,

$$f(S) := \sum_{e \in \delta^+(S)} c(e), \quad \forall S, s \in S \subseteq V \setminus \{t\}\,.$$

Cuts in directed graphs provide very natural upper bounds on the values of flows. The well known Max-Flow Min-Cut Theorem tells us that the minimum value of $f(S)$ is equal to the maximum value of a flow in G going from s to t [10]. The problem of finding a minimum cut can be solved efficiently using the Ford-Fulkerson algorithm, some of its specializations such as the improvement by Edmonds and Karp, or by solving corresponding linear program. We can look at this problem differently, knowing that function f defined above is submodular.

Proposition 2.2. *The cut capacity function is submodular.*

Proof. Let $A \subseteq B \subseteq V \setminus \{t\}$ and $v \in V \setminus (B \cup \{t\})$. Then

$$\begin{aligned} f(A \cup \{v\}) &= \sum_{a \in A} \sum_{x \in V \setminus (A \cup \{v\})} c(a,x) + \sum_{x \in V \setminus (A \cup \{v\})} c(v,x) \\ f(A) &= \sum_{a \in A} \sum_{x \in V \setminus (A \cup \{v\})} c(a,x) + \sum_{a \in A} c(a,v) \\ \triangle_f(v \mid A) &= f(A \cup \{v\}) - f(A) = \sum_{x \in V \setminus (A \cup \{v\})} c(v,x) - \sum_{a \in A} c(a,v)\,. \end{aligned}$$

Similarly we get

$$\triangle_f(v \mid B) = \sum_{x \in V \setminus (B \cup \{v\})} c(v,x) - \sum_{b \in B} c(b,v).$$

Inequality $\triangle_f(v \mid A) \geq \triangle_f(v \mid B)$ holds since $c(u,v) \geq 0$ for all$(u,v) \in E$, $V \setminus (B \cup \{v\}) \subseteq V \setminus (A \cup \{v\})$ and $A \subseteq B$. □

So we can also find a minimum cut if we minimize the function f, since minizmization of submodular function can be done in polynomial time, see Section 4.1. The problem of maximizing function f as defined above is know as Max DiCut problem. Maximizing a non-negative submodular function is an NP-hard problem. For an arbitrary submodular function given by an oracle, even checking whether there exists a set S such that $f(S) > 0$ is NP-hard. Usually, when we want to maximize such a function, we consider the cut capacity of an undirected graph $G = (V, E)$ without sink and source. We refer to it as the Max Cut problem. Then what we want to maximize is :

$$f(S) := \sum_{e \in \delta(S)} c(e), \quad \forall S, S \subseteq V$$

where $c : E \to \mathbb{R}^+$ and $\delta(S) := \{\{u,v\} \in E : u \in S, v \notin S\}$ denotes set of all edges with one end in S and the other one in $V \setminus S$. Using the same proof as for the directed case, we can prove that f is again submodular. Such a function is an example of a *symmetric* submodular function (that is, $f(S) = f(V \setminus S)$ for all $S \subseteq V$).

2.6 Coverage functions

One more important example of a submodular function is a *coverage function*. Let U be a finite universal set and let V be a collection of subsets of U. Then the function $f : 2^V \to \mathbb{R}^+$ defined, for all subcollections $S \subseteq V$, as

$$f(S) = \left| \bigcup_{A \in S} A \right|$$

is monotone submodular [30]. A typical goal in aplications of maximizing such a function is to maximize the number of elements covered, subject to a cardinality constraint, $|S| \leq k$ for some $k \in \mathbb{N}$. The problem of maximizing $f(S)$ in that case is well-known *max-cover* problem [34].

2.7 Maximizing the spread of influence through a social network

A social network can be represented by a graph of relationships and interactions between members of a group. A social network is a fundamental tool for describing the spread of information and influence among members of certain group. We can represent a social network as a weighted directed graph whose vertices represent people, and weights of edges represent probability that the starting vertex of the edge will influence its end vertex. In such a setting we want to explore the influence of a subset of people on the entire social netwok, in order to find a group of people that influence the network the most. This way we could supply the most influential people with some information, and the probability that the information would spread across the complete network would be the highest possible. For various models of spreading the information, the resulting influence function is submodular. The greedy algorithm proposed by Nemhauser et al. [25] can be used for finding a most influential group in a social network. More details about precise formulation of the problem can be found in [18, 20].

2.8 The rank function of a matroid

Definition 2.3. A pair $(V, \mathcal{I})$ is matroid if V is a finite set and $\mathcal{I}$ is a nonempty collection of subsets of V satisfying

- $Y \subseteq X \in \mathcal{I} \Rightarrow Y \in \mathcal{I}$
- $X, Y \in \mathcal{I}$, $|X| < |Y| \Rightarrow \exists v \in Y \setminus X$ such that $X \cup \{v\} \in \mathcal{I}$.

We call V *ground set* and we say that sets in $\mathcal{I}$ are *independent* and set not in $\mathcal{I}$ are *dependent*.

Given a matroid $M = (V, \mathcal{I})$, we define the *rank function* r of M for all $S \subseteq V$ as

$$r(S) := \max\{|X| \,:\, X \subseteq S, X \in \mathcal{I}\}\,.$$

Theorem 2.4. *Let $M = (V, \mathcal{I})$ be a matroid and r the rank function of M. Then r is submodular.*

A proof of this theorem can be found in, e.g., [8, 30].

Canonical examples of matroids are the so-called *linear matroids*. Let $A \in \mathbb{R}^{m \times n}$ and $V = \{1, \ldots, n\}$. Let $\mathcal{I}$ be the collection of all subsets $X \subseteq V$ for which the submatrix of A consisting of the columns indexed by X has rank $|X|$. That is, the collection of all subsets $X \subseteq V$ for which the columns of A indexed by X are linearly independent. In this case the value of rank function of subset $X \subseteq V$ is equal to the rank of the matrix formed by the columns indexed by X.

2.9 Polymatroid functions

A special subclass of submodular functions is represented by the so-called polymatroid functions. They are a very useful tool for bounding the number of maximal independent sets in hypergraphs and lattices [4] as well as enumerating all maximal independent sets in hypergraph [3] and data mining [19]. We call a function $f : 2^V \to \mathbb{R}$ *polymatroid* if the following is satisfied:

- $f(\emptyset) = 0$,
- f is monotone submodular,
- f is integer-valued.

It is obvious that the rank function of a matroid and coverage function are polymatroid (Sections 2.6 and 2.8). For better understanding we need to recall the definitions of a hypergraph and of an independent set in a hypergraph.

Definition 2.5. A *hypergraph* is a pair $\mathcal{H} = (V, \mathcal{E})$ where V is a finite set of *vertices* and $\mathcal{E}$ is a set of non-empty subsets of V called *hyperedges* (or simply just *edges*).

Definition 2.6. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, we say that a set $I \subseteq V$ is *independent* in $\mathcal{H}$ if it contains no hyperedge of $\mathcal{H}$.

If it is clear from context to which hypergraph we refer to, we will say just a set I is independent. For the definition of an exact polymatroid separator we first need the definition of a polymatroid separator.

Definition 2.7. [3] Given a non-empty hypergraph $\mathcal{H} = (V, \mathcal{E})$, a polymatroid function $f : 2^V \to \mathbb{Z}^+$, and a number $t \in \mathbb{Z}^+$, the pair (f, t) is called *polymatroid separator* for $\mathcal{H}$ if $f(E) \geq t$ holds for every hyperedge $E \in \mathcal{E}$.

It is obvious from the above definition that if (f, t) is a polymatroid separator of $\mathcal{H}$, then $f(S) \geq t$ for every set $S \subseteq V$ that is not independent in $\mathcal{H}$.

Definition 2.8. A polymatroid separator for $\mathcal{H} = (V, \mathcal{E})$ is called *exact* if $f(I) < t$ for every $I \subseteq V$ independent set in $\mathcal{H}$.

Notice that using truncation (see Section 3), if we have an exact polymatroid separator (f, t) for $\mathcal{H} = (V, \mathcal{E})$, then also (g, t) is an exact polymatroid separator for $\mathcal{H} = (V, \mathcal{E})$, where $g(S) := \min\{f(S), t\}$ $\forall S \subseteq V$. Then, for every set A that is not independent it holds $g(A) = t$.

We say a hypergraph is *Sperner* if no hyperedge contains another hyperedge.

Definition 2.9. The *dual* $\mathcal{H}^d$ of a hypergraph $\mathcal{H} = (V, \mathcal{E})$ is the hypergraph with the same vertex set V and hyperedge set that contains all inclusion-wise minimal sets that intersect all hyperedges in $\mathcal{H}$, i.e., $\mathcal{H}^d = (V, \mathcal{E}^d)$ where

$$\mathcal{E}^d = \text{Inclusion-wise minimal elements in } \{X \subseteq V \mid \forall E \in \mathcal{E},\, X \cap E \neq \emptyset\}$$

It is known that the dual of every hypergraph is Sperner. The following propostion was proved by Boros et. al [3].

Proposition 2.10. *Every non-empty Sperner hypergraph $\mathcal{H} = (V, \mathcal{E})$ has an exact polymatroid separator defined $\forall S \subseteq V$ as*

$$f(S) := |\{X \in \mathcal{E}^d \,:\, S \cap X \neq \emptyset\}|$$

and $t = |\mathcal{E}^d|$.

Note that Proposition 2.10 implies that every non-empty hypergraph has an exact polymatroid separator. Indeed, every non-empty hypergraph $\mathcal{H} = (V, \mathcal{E})$ can be reduced to a non-empty Sperner hypergraph $\mathcal{H}'$ on the same vertex set simply by keeping only those hyperedges of $\mathcal{H}$ that do not contain any other hyperedge (that is, the minimal elements of $\mathcal{E}$ with respect to inclusion relation). It follows directly from the definitions that every exact polymatroid separator of $\mathcal{H}'$ is also an exact polymatroid separator of $\mathcal{H}$.

We will now give an alternative proof of the fact that every hypergraph has an exact polymatroid separator. For convenience, let us first show the following theorem.

Theorem 2.11. *Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph. Then there exists a nonnegative monotone submodular function $f : 2^V \to \mathbb{R}^+$ such that for all $S \subseteq V$:*

$$f(S) < 1 \iff S \text{ is an independent set in } \mathcal{H}\,.$$

Proof. For all $S \subseteq V$, let us define $f : 2^V \to \mathbb{R}^+$ as follows:

$$f(S) = \begin{cases} \sum_{i=1}^{|S|} \frac{1}{2^i} & \text{if } S \text{ is independent in } \mathcal{H} \\ 1 & \text{otherwise} \end{cases}$$

It is clear from the definition that $f(S) < 1$ if and only if S is independent, since $\sum_{i=1}^{|S|} \frac{1}{2^i} = 1 - \frac{1}{2^{|S|}} < 1$. What we need to prove is that f is monotone submodular. Monotonicity is obvious. Let $A \subseteq B \subset V$ and let $v \in V \setminus B$. We will show that $\triangle_f(v \mid A) \geq \triangle_f(v \mid B)$. If B is not independent then $\triangle_f(v \mid B) = 0$ and that case is done. Suppose B is independent, then A is also independent. If $A = B$, then clearly $\triangle_f(v \mid A) = \triangle_f(v \mid b)$. So let $A \subset B$ and denote $|A| = n$ and $|B| = m$. Then $m > n$. We consider two possibilities for $A \cup \{v\}$, and $B \cup \{v\}$.
If $A \cup \{v\}$ is independent then

$$\triangle_f(v \mid A) = \sum_{i=1}^{n+1} \frac{1}{2^i} - \sum_{i=1}^{n} \frac{1}{2^i} = \frac{1}{2^{n+1}}\,.$$

If $A \cup \{v\}$ is not independent then

$$\triangle_f(v \mid A) = 1 - \sum_{i=1}^{n} \frac{1}{2^i} = \frac{1}{2^n}$$

In the same way we get :

$$\triangle_f(v \mid B) = f(B \cup \{v\}) - f(B) = \begin{cases} \frac{1}{2^{m+1}} & \text{if } B \cup \{v\} \text{ is independent} \\ \frac{1}{2^m} & \text{if } B \cup \{v\} \text{ is not independent} \end{cases}$$

Hence, in all cases we have:

$$\triangle_f(v \mid B) \leq \max\left\{\frac{1}{2^{m+1}}, \frac{1}{2^m}\right\} = \frac{1}{2^m} \leq \frac{1}{2^{n+1}} = \min\left\{\frac{1}{2^{n+1}}, \frac{1}{2^n}\right\} \leq \triangle_f(v \mid A)\,.$$

So if we combine those two parts, we have $\triangle_f(v \mid B) \leq \triangle_f(v \mid A)$. □

The function f given by Theorem 2.11 is easily transformed to an exact polymatroid separator for $\mathcal{H} = (V, \mathcal{E})$. The function obtained by multiplying f with 2^α where α is the maximum cardinality of an independent set in $\mathcal{H}$, is polymatroid. Such a function together with $t = 2^\alpha$ is an exact polymatroid separator for $\mathcal{H}$. Note also that the advantage of this approach over the one given by Proposition 2.10 is that we do not need to calculate the dual of $\mathcal{H}$ (whose size can be exponential in terms of the size of $\mathcal{H}$). Instead, the corresponding function values $f(S)$ can be computed directly using a single call to the membership oracle for $\mathcal{H}$. The membership oracle is a procedure that, given a set $S \subseteq V$, correctly asserts whether S is independent or not.

3 Operations preserving submodularity

László Lovász was the first to show a similarity between submodularity and convexity in [24]. He also proved many nice properties of submodular functions. Proposition 2.1 gives a way of obtaining submodular functions using modular ones. We will now discuss some other operations which can be used to build new submodular functions from existing ones [21]. For example we can check that submodularity is preserved under taking nonnegative liner combinations.

Proposition 3.1. *If $g_1, \ldots, g_n : 2^V \to \mathbb{R}$ are submodular and $\alpha_1, \ldots, \alpha_n \geq 0$, then the function $f : s^V \to \mathbb{R}$, given by $f(S) := \sum_{i=1}^{n} \alpha_i g_i(S)$, for all $S \subseteq V$, is submodular.*

The proposition is readily proved using the definition based on discrete derivatives (Theorem 1.4).

Proposition 3.2. *Let a function $g : 2^V \to \mathbb{R}$ be submodular and $B \subseteq V$. Then, the function*

$$f(S) := g(S \cup B)$$

is submodular as well.

Proof. Let $X, Y \subseteq V$, then

$$\begin{aligned} f(X \cup Y) + f(X \cap Y) &= g(X \cup Y \cup B) + g((X \cap Y) \cup B) \\ &= g((X \cup B) \cup (Y \cup B)) + g((X \cup B) \cap (Y \cup B)) \\ &\leq g(X \cup B) + g(Y \cup B) = f(X) + f(Y)\,. \end{aligned}$$

□

Submodularity is also preserved under taking the residual.

Proposition 3.3. *Given a submodular function $g : 2^V \to \mathbb{R}$ and two disjoint sets $A, B \subset V$, the residual $r : 2^A \to \mathbb{R}$ defined $\forall S \subseteq A$, as*

$$r(S) := g(S \cup B) - g(B)$$

is submodular.

Proof. The function r is the sum of the function f defined in Proposition 3.2 and function $-g(B)$, which can be seen as a constant function over 2^V, and thus modular. By Proposition 3.1, the function r is submodular. □

One more interesting property is that monotone submodular functions retain this property under *truncation*, i.e., if $g : 2^V \to \mathbb{R}$ is monotone submodular, then so is $f(S) := \min\{g(S), c\}$ defined for all $S \subseteq V$ and for a constant $c \in \mathbb{R}$. This fact can be proved using Definition 1.1 and checking all possible cases for values of $f(A \cap B)$, $f(A)$ and $f(B)$.

If we consider a similar aproach for the minimum or maximum of two submodular functions, we see that it does not work. In other words, given two submodular functions $f_1, f_2 : 2^V \to \mathbb{R}$, the functions $f_{\min}(S) := \min\{f_1(S), f_2(S)\}$ and $f_{\max}(S) := \max\{f_1(S), f_2(S)\}$, defined for all $S \subseteq V$, are not necessarily submodular.

Example 3.4. Consider two submodular functions f_1, f_2 defined on set $V = \{a, b\}$ as in Table 1. Notice that then $f_{\min}(\{a, b\}) - f_{\min}(\{b\}) = 1 > 0 = f_{\min}(\{a\}) - f_{\min}(\emptyset)$,

	$\emptyset$	$\{a\}$	$\{b\}$	$\{a, b\}$
$f_1(\cdot)$	0	1	0	1
$f_2(\cdot)$	0	0	1	1
$f_{\min}(\cdot)$	0	0	0	1

Table 1: Functions f_1, f_2, and $f_{\min}$.

and $f_{\min}$ is not submodular.

Example 3.5. Suppose we are given two submodular functions g_1, g_2 defined on set $V = \{x, y\}$ as in Table 2. Notice that g_1 and g_2 are modular. Define $g_{\max}$ analogously as $f_{\max}$. Then

$$\triangle_{g_{\max}}(y \mid \{x\}) = g_{\max}(\{x, y\}) - g_{\max}(\{x\}) = 1 > 0 = g_{\max}(\{y\}) - g_{\max}(\emptyset) = \triangle_{g_{\max}}(y \mid \emptyset)\,.$$

Hence $g_{\max}$ is not submodular.

	$\emptyset$	$\{x\}$	$\{y\}$	$\{x, y\}$
$g_1(\cdot)$	0	1	1	2
$g_2(\cdot)$	1	1	1	1
$g_{\max}(\cdot)$	1	1	1	2

Table 2: Functions g_1, g_2, and $g_{\max}$.

Note that all four functions f_1, f_2, g_1, and g_2 are modular. This shows that the minimum (resp., maximum) of two modular functions is not necessarily modular (in fact, not even submodular).

Although neither the minimum nor the maximum of two submodular functions is submodular in general, it is proved in [24] that with additional assumptions the minimum of two submodular functions will be again submodular.

Proposition 3.6. *If f_1 and f_2 are submodular set functions such that $f_1 - f_2$ is either monotone increasing or monotone decreasing, then $f_{\min}$ is also submodular.*

Using Definition 1.1, it is easy to see that given a submodular function $f : 2^V \to \mathbb{R}$ we can define a new submodular function g, defined as $g(S) := f(V \setminus S)$ for all $S \subseteq V$. Given two sets $X, Y \subseteq V$, we have

$$\begin{aligned} g(X \cup Y) + g(X \cap Y) &= f(V \setminus (X \cup Y)) + f(V \setminus (X \cap Y)) \\ &= f((V \setminus X) \cap (V \setminus Y)) + f((V \setminus X) \cup (V \setminus Y)) \\ &\leq f(V \setminus X) + f(V \setminus Y) = g(X) + g(Y) \end{aligned}$$

Building new submodular functions from old ones allows us to extend the effective range of certain theorems. Using the properties of submodular functions, we can prove submodularity easily in many other cases.

4 Submodular functions optimization

Lovász showed that algorithmically, submodular functions behave similarly to convex functions [24]. Minimization of submodular set function can be done in polynomial time, just like a convex function can be minimized efficiently [1, 28]. This way submodularity represents a discrete analogue of convex functions. While convex functions play essential role in continuous optimization, submodular functions are very important in the development of techniques for discrete optimization. Direct applications of submodular functions in many different practical areas like electrical networks, airline crew scheduling, biodiversity conservation, sensors placing, statics, machine learning, enviromental monitoring as well as their application to mathematical areas like game theory, graph theory and combinatorics make them very interesting to explore and optimize [2, 5, 12, 20–22, 27]. Submodular functions also generalize many different optimization problems as we have seen in Section 2. Besides maximum facility location, Max Cut, Max DiCut there are many more submodular problems, see, e.g., [8, 24].

4.1 Minimization

Suppose we are given a submodular function $f : 2^V \to \mathbb{R}$. The problem of submodular function minimization (SFM) is to find a set $S \subseteq V$ minimizing $f(S)$. The first polynomial and strongly polynomial algorithm for solving the SFM problem was given by Grötschel, Lovász, and Schrijver [15, 16]. The algorithms that they constructed rely on the ellipsoid method. Suppose $f : 2^V \to \mathbb{R}$ is a set function. We can enumarate elements of V as $V = \{1, 2, \ldots, n\}$. Then we can assing to each subset $S \subseteq V$ a binary vector e_S where the i–th component of e_S is 1 if $i \in S$, and 0 otherwise. We can look at our function f as a function defined over the corners of the unit cube: $\tilde{f} : \{0, 1\}^n \to \mathbb{R}$ defined as $\tilde{f}(e_S) := f(S)$ for all $S \subseteq V$. Now it is natural that we extend $\tilde{f}$ to the entire unit cube $[0, 1]^n$. The *Lovaśz extension* extends $\tilde{f}$ to a convex function. This way we can minimize the function $\tilde{f}$ using the ellipsod method, which is an iterative method for minimizing convex functions. The idea behind this approach is used to prove that SFM is possible in polynomial time. A different

extension, namely *multilinear extension* proposed by Vondrák, has proven to useful for constrained submodular function maximization [31].

In 2000, Schrijver developed a combinatorial algorithm that solves SFM.[1] He proved that his algorithm runs in $O(n^8EO+n^9)$ time [29].[2] Later, in 2003, Vygen proved that the algorithm proposed by Schrijver runs in fact in $O(n^7EO+n^8)$ [32].

In 2009 James B. Orlin improved at that time the best strongly polynomial time combinatorial algorithm for SFM proposed by Iwata, which runs in $O((n^6EO+n^7)\log n)$ time, by more than a factor of n [17, 26]. Orlin proved that SFM can be done in $O(n^5EO+n^6)$.

Recently, a further improvement was announced by Lee et al. [23].

4.2 Submodular maximization under constraints

Unlike minimization, maximization of a submodular function is an NP-hard problem for many classes of submodular functions, such as weighted coverage or maximization of a monotone submodular function under a cardinality constraint [13,21]. This means we are interested in finding a subset of V which solves a problem of the form

$$\max_{S\subseteq V} f(S) \text{ subject to some constraints on } S. \tag{4.1}$$

Most often we require a *cardinality constraint*, that is, $|S| \leq k$ for some $k \in \mathbb{N}$. Since these problems are NP-hard, we can approach them with different heuristics, approximations algorithms or practical exact methods for solving the problem to optimality on small instances.

We will start with analysing a very simple algorithm for finding approximate solutions to the problem of maximizing a monotone submodular function under a cardinality constraint, (4.1). As we mentioned in Section 2.3, maximizing a monotone function without any constraints is solved by taking the entire set V. Given a monotone submodular function $f : 2^V \to \mathbb{R}$ and an integer k, we want to solve

$$\max_{|S|\leq k} f(S)\,.$$

Nemhauser et al. proved that the *greedy algorithm* provides a very good constant approximation to the optimal solution of our problem [25].

[1] The term "combinatorial algorithm" is not precisely defined, and it is used to emphasize that the algorithm does not rely on the ellipsoid method.

[2] EO represents the maximum amount of time it takes to evaluate $f(S)$ for a subset $S \subseteq V$. EO stands for evaluation oracle.

The algorithm starts with the empty set S_0, and at each iteration $i \geq 1$ adds an element maximizing the discrete derivative $\triangle(v \mid S_{i-1})\}$, i.e., adds the element from $\arg\max_v \triangle(v \mid S_{i-1})$. In problems of maximizing a submodular function we assume that we are given a function by a *value oracle*, i.e., we can query a black box which calculates $f(S)$ in certain unit of time.

Algorithm 1: The greedy algorithm for maximizing a submodular function subject to a cardinality constraint $|S| \leq \ell$.

Input: A monotone submodular function f given by a value oracle and a number ℓ.

Output: A subset $S_\ell \subseteq V$.

```
S_0 := ∅;
for i = 1, ..., ℓ do
    v* ∈ arg max_v △(v | S_{i-1});
    S_i = S_{i-1} ∪ {v*};
return S_ℓ;
```

It makes no sense to look at cases where $k > |V|$ or $\ell > |V|$. Nemhauser et al. proved that $f(S_k) \geq (1 - 1/e) \max_{|S| \leq k} f(S)$, but this can be slightly generalised, as in the following theorem from [21].

Theorem 4.1. *For a non-negative monotone submodular function $f : 2^V \to \mathbb{R}^+$ and numbers $k, \ell \in \{1, \ldots, |V|\}$, let $\{S_i\}_{i=0}^{|V|}$ be the sequence of sets selected by Algorithm 1. Then the following holds*

$$f(S_\ell) \geq \left(1 - e^{-\ell/k}\right) \max_{S:|S| \leq k} f(S)\,.$$

Proof. Let $k, \ell \in \mathbb{N}$, and let S^* be an optimal set of size k. We can assume that S^* is of size k, since f is monotone. Order the elements of S^* as $\{v_1, \ldots, v_k\}$. Then for all $i < \ell$, the following inequalities hold.

$$f(S^*) \leq f(S^* \cup S_i) \tag{4.2}$$

$$= f(S_i) - f(S_i) + f(S_i \cup \{v_1\}) \mp \cdots - f(S_i \cup \{v_1, \ldots, v_{k-1}\}) + f(S_i \cup S^*) \tag{4.3}$$

$$= f(S_i) + \sum_{j=1}^{k} \triangle_f(v_j \mid S_i \cup \{v_1, \ldots, v_{j-1}\}) \tag{4.4}$$

$$\leq f(S_i) + \sum_{j=1}^{k} \triangle_f(v \mid S_i) \tag{4.5}$$

$$
\begin{aligned}
&= f(S_i) + \sum_{v \in S^*} \triangle_f(v \mid S_i) \\
&\leq f(S_i) + \sum_{v \in S^*} (f(S_{i+1}) - f(S_i)) \qquad (4.6) \\
&\leq f(S_i) + k(f(S_{i+1}) - f(S_i)) \qquad (4.7)
\end{aligned}
$$

Inequality (4.2) is due to monotonicity of f. Equalities (4.3) and (4.4) are used to tranform a telescopic sum. Next inequality (4.5) follows from the diminishing return proprety of submodularity of f. Inequality (4.6) follows from fact that S_{i+1} is a set chosen greedily at the $(i+1)$st iteration, and it maximizes $\triangle(v \mid S_i)$. The last inequality, (4.7), holds because $|S^*| \leq k$.

Define $d_i := f(S^*) - f(S_i)$, which represents how far we are from the optimal value $f(S^*)$ at the ith iteration. Then we have $d_i = f(S^*) - f(S_i) \leq k(f(S_{i+1}) - f(S_i))$. We can rewrite it as $d_i \leq k(d_i - d_{i+1})$, which gives us

$$d_{i+1} \leq \left(1 - \frac{1}{k}\right) d_i \qquad (4.8)$$

Inequality (4.8) holds for all $i < \ell$, hence $d_\ell \leq (1 - \frac{1}{k})^\ell \, d_0$. Since f is non-negative we have $d_0 = f(S^*) - f(\emptyset) \leq f(S^*)$. Together with the known inequality $1 - x \leq e^{-x}$, which holds for all $x \in \mathbb{R}$, we have

$$d_\ell \leq \left(1 - \frac{1}{k}\right)^\ell d_0 \leq e^{-\ell/k} f(S^*).$$

Substituing back d_ℓ, we get

$$
\begin{aligned}
f(S^*) - f(S_\ell) &\leq e^{-\ell/k} f(S^*) \\
f(S_\ell) &\geq \left(1 - e^{-\ell/k}\right) f(S^*) \,.
\end{aligned}
$$

□

Generalization with $\ell \neq k$ can be used to check how close to the optimum is the solution produced by greedy algorithm. We can run the greedy algorithm up to $5k$ iterations. We will then produce an approximation of ≈ 0.99 to the optimal solution. We can compare $5k$ approximation to our usual solution, allowing k elements under the cardinality constraint. This way we can see how far the solution is from the approximation of ≈ 0.99 to the optimal solution. When $k = \ell$, Theorem 4.1 guarantees us ≈ 0.63 approximation to the optimal solution. Nemhauser also proved that we cannot obtain a better approximation guarante than $(1 - 1/e)$ for many classes of submodular functions.

The Max-k-cover problem, where we want to choose k sets whose union is as large as possible, is one of the problems where the greedy algorithm is optimal, unless $P = NP$ [25]. That is, there exist no polynomial time algorithm that guarantees a better worst-case approximation bound than the greedy algorithm, unless $P = NP$.

In contrast to maximizing a monotone submodular function subject to constraints, we can also ask for a minimum cardinality set that would achieve a given amount q under a monotone submodular function $f : 2^V \to \mathbb{R}$. That is, we want to find a minimum size set $S \subseteq V$ subject to $f(S) \geq q$. We demand that f is monotone, so it is natural to bound q to the interval $[0, f(V)]$. The greedy algorithm provides an approximation to the solution of this problem, although it is not a constant factor approximation.

Theorem 4.2. *Suppose $f : 2^V \to \mathbb{N}$ is a monotone submodular function, and let $q \in [0, f(V)]$. Denote by $\{S_i\}_{i \geq 0}$ the sequence of sets picked by the greedy algorithm, Algorithm 1. If ℓ is the smallest index such that $f(S_\ell) \geq q$, then*

$$\ell \leq \left(1 + \ln \max_{v \in V} f(\{v\})\right) OPT$$

where OPT is the cardinality of a smallest set S for which $f(S) \geq q$.

Theorem 4.2 was proved by Wolsey in [34].

4.3 Unconstrained submodular maximization

The problem of *unconstrained submodular maximization* (USM) is perhaps the most basic submodular maximization problem. We consider the problem of maximizing a non-negative submodular function $f : 2^V \to \mathbb{R}$, i.e., we want to find a subset $S \subseteq V$ maximizing $f(S)$. We are interested in the general case where f is not necessarily monotone. As mentioned before, the cut function in a directed or an undirected graph represents a canonical example. The cut function in an undirected graph is a symmetic submodular function, and we will see that symmetry allows us to prove better and simpler approximation bounds. In general, given a submodular function without any restrictions, even verifying whether there exists a set S such that $f(S) > 0$ in an NP-hard problem [9]. Here we only assume that the function is non-negative.

We will see that a random set gives an approximation factor of $1/4$ for USM, and an approximation factor of $1/2$ for USM in expected value, if we maximize a symmetric function.

4.3.1 Random set

Given a finite non-empty set S and real number $p \in [0, 1]$, we define $S(p)$ to be a random subset of S, where each element is chosen independently with probability p. Define $\emptyset(p) = \emptyset$, $\forall p \in [0, 1]$.

Algorithm 2: Random set algorithm

Input: A non-negative submodular function $f : 2^V \to \mathbb{R}^+$, given by a value oracle.

Output: A subset $S \subseteq V$.

1 $S = V(1/2)$;

2 **return** S;

To prove approximation factors for Algorithm 2, we will need some preliminaries concerning submodularity and probability.

Lemma 4.3. *Let $g : 2^V \to \mathbb{R}^+$ be a submodular function and $S \subseteq V$. Then*

$$\mathbb{E}[g(S(p))] \geq (1 - p)g(\emptyset) + pg(S). \tag{4.9}$$

Proof. The proof goes by induction on $|S|$. For $S = \emptyset$, we have

$$\mathbb{E}[g(\emptyset)] = g(\emptyset) = (1 - p)g(\emptyset) + pg(\emptyset).$$

So suppose that $S = S' \cup \{v\}$, $v \notin S'$. The inductive hypothesis yields $\mathbb{E}[g(S'(p))] \geq (1 - p)g(\emptyset) + pg(S')$. Then $S(p) \cap S'$ is a subset of S', where each element is sampled

with probability p, so $S'(p) = S(p) \cap S'$. Using submodularity, we get

$$g(S(p)) + g(S') \geq g(S(p) \cup S') + g(S(p) \cap S)\,,$$

which implies

$$g(S(p)) \geq g(S(p) \cup S') + g(S'(p)) - g(S')\,.$$

$$\begin{aligned}
\text{Therefore } \mathbb{E}[g(S(p))] &\geq \mathbb{E}[g(S(p) \cup S') + g(S'(p)) - g(S')] \\
&= \mathbb{E}[g(S'(p))] + \mathbb{E}[g(S(p) \cup S') - g(S')] && (4.10)\\
&= \mathbb{E}[g(S'(p))] + p(g(S) - g(S')) + (1-p)(g(S') - g(S')) && (4.11)\\
&\geq (1-p)g(\emptyset) + pg(S') + pg(S) - pg(S') \\
&= (1-p)g(\emptyset) + pg(S)\,.
\end{aligned}$$

Equation (4.10) is due to linearity of expected value (Proposition 1.8). Equality (4.11) holds since $S = S' \cup \{v\}$ and $v \notin S'$, it follows that $S(p) \cup S'$ equals either S with probability p or S' with probability $1 - p$. □

Lemma 4.4. *Let f be a submodular function on a finite set V, and let $A, B \subseteq V$. Let $A(p)$, $B(q)$ be their independently sampled subsets, where $p, q \in [0, 1]$. Then*

$$\mathbb{E}[f(A(p) \cup B(q))] \geq (1-p)(1-q)f(\emptyset) + p(1-q)f(A) + (1-p)qf(B) + pqf(A \cup B)\,.$$

Proof. Fix $R \subseteq A$, and define $g(T) = f(R \cup T)$ for all $T \subseteq B$. The function g is submodular, see Proposition 3.2. Lemma 4.3 implies $\mathbb{E}[g(B(q))] \geq (1-q)f(R) + qf(R \cup B)$. We know that $\mathbb{E}[g(B(q))] = \mathbb{E}[f(A(p) \cup B(q))|A(p) = R]$. Combining the last two facts with Proposition 1.9 we get

$$\begin{aligned}
\mathbb{E}[f(A(p) \cup B(q))] &= \mathbb{E}[\mathbb{E}[f(A(p) \cup B(q))|A(p)]] \\
&\geq \mathbb{E}[(1-q)f(A(p)) + qf(A(p) \cup B)]\,.
\end{aligned}$$

Again by applying Lemma 4.3 to functions f and $h(S) := f(S \cup B)$ for $S \subseteq A$, we get

$$\begin{aligned}
\mathbb{E}[f(A(p))] &\geq (1-p)g(\emptyset) + pg(A) \\
\mathbb{E}[f(A(p) \cup B)] &\geq (1-p)f(B) + pf(A \cup B).
\end{aligned}$$

Together with the linearity of expected value this implies the claim. □

Lemma 4.3 and Lemma 4.4 will immediately show the performance of Algorithm 2.

Theorem 4.5. *Let $f : 2^V \to \mathbb{R}$ be a non-negative submodular function. Let OPT be a set maximizing the function f, i.e., $OPT \in \arg\max_{S \subseteq V} f(S)$. If R denotes a set produced by the random set algorithm, then*

$$\mathbb{E}[f(R)] \geq \frac{1}{4} f(OPT)\,.$$

In addition, if f is symmetric, then $\mathbb{E}[f(R)] \geq \frac{1}{2} f(OPT)$.

Proof. Suppose that S is a set that achieves $f(OPT)$. Then we can write $R = S(1/2) \cup (V \setminus S)(1/2)$. Then by Lemma 4.4

$$\mathbb{E}[f(R)] \geq \frac{1}{4}f(\emptyset) + \frac{1}{4}f(S) + \frac{1}{4}f(V \setminus S) + \frac{1}{4}f(V) \geq \frac{1}{4}f(S) = \frac{1}{4}fOPT)\,.$$

Inequality follows from the non-negativity of f.
If f is symmetric, then also $f(V \setminus S) = f(OPT)$ (since $f(S) = f(V \setminus S)$). Therefore if f is symmetric then $\mathbb{E}[f(R)] \geq \frac{1}{2}f(OPT)$. □

4.3.2 A deterministic (1/3)-approximation algorithm

We will see one more simple (1/3)-approximation algorithm for USM proposed by Buchbinder et. al [7]. The algorithm is deterministic and runs in linear time. First, we order our ground set V arbitrarily, $V = \{v_1, \ldots, v_n\}$. We start with two solutions X_0 and Y_0. For each element v_i, the algorithm makes one corresponding iteration modifying current solutions X_{i-1} and Y_{i-1}. The idea is that, in the ith iteration, based on discrete derivative of the two options we either add v_i to X_{i-1} or we remove v_i from Y_{i-1}. This means that at each iteration $X_i \subset Y_i$ for all $i < n$, and after nth iteration we have $X_n = Y_n$.

Algorithm 3: A deterministic algorithm for USM

Input: A non-negative submodular function $f : 2^V \to \mathbb{R}^+$, given by a value oracle.

Output: A subset $X_n \subseteq V$ which approximates $\max_{S \subseteq V} f(S)$.

```
X_0 = ∅, Y_0 = V;
for i = 1, ..., n do
    a_i = f(X_{i-1} ∪ {v_i}) − f(X_{i-1});
    b_i = f(Y_{i-1} \ {v_i}) − f(Y_{i-1});
    if a_i ≥ b_i then
        X_i = X_{i-1} ∪ {v_i};
        Y_i = Y_{i-1};
    else
        X_i = X_{i-1};
        Y_i = Y_{i-1} \ {v_i};
return X_n, or equivalently Y_n;
```

We need two lemmas to prove that the approximation guarantee of Algorithm 3 is

1/3. Like in the algorithm, for all i denote

$$a_i := f(X_{i-1} \cup \{v_i\}) - f(X_{i-1}) \,,$$
$$b_i := f(Y_{i-1} \setminus \{v_i\}) - f(Y_{i-1}) \,.$$

Lemma 4.6. *For every $1 \leq i \leq n$, it holds $a_i + b_i \geq 0$.*

Proof. Since $X_{i-1} \subseteq Y_{i-1}$ and we know that $v_i \notin X_{i-1}$ and $v_i \in Y_{i-1}$, we have that

$$(X_{i-1} \cup \{v_i\}) \cup (Y_{i-1} \setminus \{v_i\}) = X_{i-1} \cup Y_{i-1} \cup \{v_i\} = Y_{i-1} \,,$$
$$(X_{i-1} \cup \{v_i\}) \cap (Y_{i-1} \setminus \{v_i\}) = (X_{i-1} \cap Y_{i-1}) \setminus \{v_i\} = X_{i-1} \,.$$

Then by submodularity we have

$$\begin{aligned} a_i + b_i &= f(X_{i-1} \cup \{v_i\}) - f(X_{i-1}) + f(Y_{i-1} \setminus \{v_i\}) - f(Y_{i-1}) \\ &= [f(X_{i-1} \cup \{v_i\}) + f(Y_{i-1} \setminus v_i)] - [f(X_{i-1}) + f(Y_{i-1})] \geq 0 \,. \end{aligned}$$ □

Let OPT denotes a set that achieves the maximum of f. Define $OPT_i := (OPT \cup X_i) \cap Y_i = X_i \cup (OPT \cap Y_i)$. Notice that $OPT_0 = OPT$ and $OPT_n = X_n = Y_n$. The idea of the proof is to bound the loss in each set along the sequence $f(OPT_0), f(OPT_1), \ldots, f(OPT_n)$. The next lemma will bound the loss in value in each cosecutive step, i.e., we will bound $f(OPT_{i-1}) - f(OPT_i)$.

Lemma 4.7. *For every $1 \leq i \leq n$, following holds*

$$f(OPT_{i-1}) - f(OPT_i) \leq f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})$$

Proof. Let us first assume that $a_i \geq b_i$. Then $X_i = X_{i-1} \cup \{v_i\}$ and $Y_i = Y_{i-1}$. Notice that also $OPT_i = (OPT \cup X_i) \cap Y_i = (OPT \cup X_{i-1} \cup \{v_i\}) \cap Y_{i-1} = ((OPT \cup X_{i-1}) \cap Y_{i-1}) \cup (\{v_i\} \cap Y_{i-1}) = OPT_{i-1} \cup \{v_i\}$. Now the inequality that we need to prove becomes:

$$f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) \leq f(X_i) - f(X_{i-1}) = a_i \,.$$

By Lemma 4.6 $a_i + b_i \geq 0$, and we assumed that $a_i \geq b_i$, so $a_i \geq 0$. Now we consider two cases. If $v_i \in OPT$ then $f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) = 0$ and the inequality holds. Suppose $v_i \notin OPT$. Then since $v_i \notin X_{i-1}$, we have $v_i \notin OPT_{i-1}$. Also since $OPT_{i-1} \subset Y_i$ and $v_i \in Y_i$, we can use submodularity to derive

$$f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) \leq f(Y_i \setminus v_i) - f(Y_{i-1}) = b_i \leq a_i \,.$$

The case where $b_i \geq a_i$ is analogous. □

Theorem 4.8. *Let $f : 2^V \to \mathbb{R}$ be a non-negative submodular function. Let OPT be a set maximizing the function f, i.e., $OPT \in \arg\max_{S\subseteq V} f(S)$. Then the solution produced by Algorithm 3 is a $(1/3)$-approximation,*

$$f(X_n) = f(Y_n) \geq \frac{1}{3} f(OPT)\,.$$

Proof. Consider the following sum:

$$\sum_{i=1}^{n} [f(OPT_{i-1}) - f(OPT_{i-1})] \leq \sum_{i=1}^{n} [f(X_i) - f(X_{i-1})] + \sum_{i=1}^{n} [f(Y_i) - f(Y_{i-1})]\,.$$

Summing up the above telescopic sum and using fact that f is non-negative we get

$$f(OPT_0) - f(OPT_n) \leq [f(X_n) - f(X_0)] + [f(Y_n) - f(Y_0)] \leq f(X_n) + f(Y_n)\,.$$

Recall that $f(OPT_0) = f(OPT)$, $f(OPT_n) = f(X_n)$, we obtain,

$$f(X_n) = f(Y_n) \geq f(OPT)/3\,.$$ □

4.3.3 A randomized $(1/2)$-approximation algorithm

The algorithm from the previous section can be improved to a $(1/2)$-approximation by allowing randomness in it. In Algorithm 3 we had a deterministic way of choosing whether we include or exclude some element v_i in our solution. Making this decision random based on values a_i and b_i, we will improve our approximation to $1/2$ under expected value. The algorithm follows an arbitrary order of the ground set, $V = \{v_1, v_2, \ldots, v_n\}$.

It can be readily checked that Algorithm 2, Algorithm 3, Algorithm 4 are very efficient since they run in linear time.

We will keep the notation as above, $OPT_i := (OPT \cup X_i) \cap Y_i$, and also as above it holds that $OPT_0 = OPT$ and $OPT_n = X_n = Y_n$. Again we will use a lemma in which we will consider the sequence $\mathbb{E}[f(OPT_0)], \mathbb{E}[f(OPT_1)], \ldots, \mathbb{E}[f(OPT_n)]$ and the loss in averege value in two consecutive elements, i.e., $\mathbb{E}[f(OPT_i) - f(OPT_{i-1})]$. Notice that $\mathbb{E}[f(OPT_0)] = \mathbb{E}[f(OPT)] = f(OPT)$.

Lemma 4.9. *For every $1 \leq i \leq n$, it holds*

$$\mathbb{E}[f(OPT_{i-1}) - f(OPT_i)] \leq \frac{1}{2}\mathbb{E}[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})]\,.$$

Proof. It is enough to prove the inequality conditioned on an event of the form $X_{i-1} = S_{i-1}$, where $S_{i-1} \subseteq \{v_1, \ldots, v_{i-1}\}$ and the probability that $X_{i-1} = S_{i-1}$ is bigger than

Algorithm 4: A ranodomized algorithm for USM

Input: A non-negative submodular function $f : 2^V \to \mathbb{R}^+$, given by a value oracle.

Output: A subset $X_n \subseteq V$ which approximates $\max_{S \subseteq V} f(S)$.

$X_0 = \emptyset$, $Y_0 = V$;
for $i = 1, \ldots, n$ **do**
  $a_i = f(X_{i-1} \cup \{v_i\}) - f(X_{i-1})$;
  $b_i = f(Y_{i-1} \setminus \{v_i\}) - f(Y_{i-1})$;
  $a'_i = \max\{a_i, 0\}$;
  $b'_i = \max\{b_i, 0\}$;
  with probability $a'_i/(a'_i + b'_i)$ (if $a'_i + b'_i = 0$, then with probability 1) **do:**
  $X_i = X_{i-1} \cup \{v_i\}$;
  $Y_i = Y_{i-1}$;
  otherwise (with probability $b'_i/(a'_i + b'_i)$) **do:**
  $X_i = X_{i-1}$;
  $Y_i = Y_{i-1} \setminus \{v_i\}$;
return X_n, or equivalently Y_n;

zero. So let $X_{i-1} = S_{i-1}$ be such an event. The conditional inequality that we need to prove is:

$$E[f(OPT_{i-1}) - f(OPT_i)|X_{i-1} = S_{i-1}] \leq \frac{1}{2}\mathbb{E}[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})|X_{i-1} = S_{i-1}] \tag{4.12}$$

By the algorithm we know that $Y_{i-1} = X_{i-1} \cup \{v_i, \ldots, v_n\} = S_{i-1} \cup \{v_i, \ldots, v_n\}$ hence if we fix $X_{i-1} = S_{i-1}$, then Y_{i-1} is fixed as well. Notice that same holds for a_i, b_i and $OPT_{i-1} = (OPT \cup X_{i-1}) \cap Y_{i-1} = (OPT \cup S_{i-1}) \cap (S_{i-1} \cup \{v_i, \ldots, v_n\}) = S_{i-1} \cup (OPT \cap \{v_i, \ldots, v_n\})$. By Lemma 4.6, $a_i + b_i \geq 0$, so it cannot be that both a_i and b_i are less than zero. Because of that we consider following three cases:
Case 1: $a_i \geq 0$ *and* $b_i \leq 0$. Here $a'_i/(a'_i + b'_i) = 1$ and the following happens: $X_i = S_{i-1} \cup \{v_i\}$ and $Y_i = Y_{i-1} = S_{i-1} \cup \{v_i, \ldots, v_n\}$. Also $OPT_i = (OPT \cup X_i) \cap Y_i = (OPT \cup S_{i-1} \cup \{v_i\}) \cap (S_{i-1} \cup \{v_i, \ldots, v_n\}) = OPT_{i-1} \cup \{v_i\}$. This way all random events conditioned on $X_{i-1} = S_{i-1}$ in Inequality (4.12) become fixed. Notice that $f(Y_i) - f(Y_{i-1}) = 0$. Thus, we need to prove:

$$f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) \leq \frac{1}{2}[f(X_i) - f(X_{i-1})] = \frac{a_i}{2} \tag{4.13}$$

If $v_i \in OPT$, then by the definition of OPT_i we get that the left hand side of (4.13) is 0, and we know that $a_i \geq 0$.

If $v_i \notin OPT$, then since $v_i \notin X_{i-1}$ it holds $v_i \notin OPT_{i-1}$. Hence $OPT_{i-1} = (OPT \cup X_{i-1}) \cap Y_{i-1} \subseteq Y_{i-1} \setminus \{v_i\}$. This allows us to use the discrete derivative definition of submodularity:

$$f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) \leq f(Y_i \setminus \{v_i\}) - f(Y_{i-1}) = b_i \leq 0 \leq \frac{a_i}{2}\,.$$

Case 2: $a_i < 0$ *and* $b_i \geq 0$. This case is analogous to Case 1.
Case 3: $a_i \geq 0$ *and* $b_i > 0$. Assumption implies $a_i' = a_i$ and $b_i' = b_i$. Then with probability $a_i/(a_i + b_i)$ it happens that $X_i = X_{i-1} \cup \{v_i\}$ and $Y_i = Y_{i-1}$ or, complementarily, with probability $b_i/(a_i + b_i)$ it happens that $X_i = X_{i-1}$ and $Y_i = Y_{i-1} \setminus \{v_i\}$. Since we fixed $X_{i-1} = S_{i-1}$, we have:

$$\begin{aligned}
&\mathbb{E}[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})|X_{i-1} = S_{i-1}] \;= \\
&= \frac{a_i}{a_i + b_i}[f(X_{i-1} \cup \{v_i\}) - f(X_{i-1})] + \frac{b_i}{a_i + b_i}[f(Y_{i-1} \setminus \{v_i\}) - f(Y_{i-1})] \\
&= \frac{a_i^2 + b_i^2}{a_i + b_i}\,. \qquad (4.14)
\end{aligned}$$

Since $X_{i-1} = S_{i-1}$ we get $OPT_i = OPT_{i-1} \cup \{v_i\}$ if $X_i = X_{i-1} \cup \{v_i\}$ and $Y_i = Y_{i-1}$, i.e., with probability $\frac{a_i}{a_i + b_i}$. Otherwise, it happens that $X_i = X_{i-1}$, $Y_i = Y_{i-1} \setminus \{v_i\}$ which gives $OPT_i = OPT_{i-1} \setminus \{v_i\}$. Then by definition of the expected value we derive:

$$\begin{aligned}
&\mathbb{E}[f(OPT_{i-1}) - f(OPT_i)|X_{i-1} = S_{i-1}] = \\
&= \frac{a_i}{a_i + b_i}[f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\})] + \frac{b_i}{a_i + b_i}[f(OPT_{i-1}) - f(OPT_{i-1} \setminus \{v_i\})] \\
&\leq \frac{a_i b_i}{a_i + b_i}\,. \qquad (4.15)
\end{aligned}$$

Note that, by the algorithm, $v_i \in Y_{i-1}$, and $v_i \notin X_{i-1}$. To prove the last inequality in (4.15) we consider two cases depending on whether v_i belongs to OPT.
If $v_i \in OPT$, then $v_i \in OPT_{i-1}$ and $f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) = 0$. Also $X_{i-1} \subseteq ((OPT \cup X_{i-1}) \cap Y_{i-1}) \setminus \{v_i\} = OPT_{i-1} \setminus \{v_i\}$. So, by submodularity,

$$f(OPT_{i-1}) - f(OPT_{i-1} \setminus \{v_i\}) \leq f(X_{i-1} \cup \{v_i\}) - f(X_{i-1}) = a_i\,.$$

If $v_i \notin OPT$, then also $v_i \notin OPT_{i-1}$ and $f(OPT_{i-1}) - f(OPT_{i-1} \setminus \{v_i\}) = 0$. Again we will use submodularity to prove that $f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) \leq b_i$. Since $OPT_{i-1} = (OPT \cup X_{i-1}) \cap Y_{i-1} \subseteq Y_{i-1} \setminus \{v_i\}$,

$$f(OPT_{i-1}) - f(OPT_{i-1} \cup \{v_i\}) \leq f(Y_{i-1} \setminus \{v_i\}) - f(Y_{i-1}) = b_i\,.$$

Recall that applying the inequality of arithmetic and geometric means to x^2 and y^2 for $x, y \in \mathbb{R}^+$ we get that $xy \leq \frac{1}{2}(x^2 + y^2)$. This way we derive $\frac{a_i b_i}{a_i + b_i} \leq \frac{1}{2}\frac{a_i^2 + b_i^2}{a_i + b_i}$. Next, by

(4.14), (4.15) we get:

$$\mathbb{E}[f(OPT_{i-1}) - f(OPT_i)|X_{i-1} = S_{i-1}] \leq \frac{a_i b_i}{a_i + b_i}$$
$$\leq \frac{a_i^2 + b_i^2}{a_i + b_i} = \frac{1}{2}\mathbb{E}[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})|X_{i-1} = S_{i-1}]$$

Unconditioning the last inequality, lemma follows. □

Theorem 4.10. *Let $f : 2^V \to \mathbb{R}$ be a non-negative submodular function. Let OPT be a set maximizing the function f, i.e., $OPT \in \arg\max_{S \subseteq V} f(S)$. Then, the solution produced by Algorithm 4 is a $(1/2)$-approximation in expectation, that is,*

$$\mathbb{E}[f(X_n)] = \mathbb{E}[f(Y_n)] \geq \frac{1}{2} f(OPT)\,.$$

Proof. Similarly as in proof of Theorem 4.8, let us sum the inequality from Lemma 4.9 for all $1 \leq i \leq n$.

$$\sum_{i=1}^{n} \mathbb{E}[f(OPT_{i-1}) - f(OPT_i)] \leq \sum_{i=1}^{n} \frac{1}{2}\mathbb{E}[f(X_i) - f(X_{i-1}) + f(Y_i) - f(Y_{i-1})]$$

The above sum is telescopic, and summing it we get:

$$\mathbb{E}[f(OPT_0) - f(OPT_n)] \leq \frac{1}{2}\mathbb{E}[f(X_n) - f(X_0) + f(Y_n) - f(Y_0)]$$
$$\leq \frac{1}{2}\mathbb{E}[f(X_n) + f(Y_n)]$$

Since $\mathbb{E}[f(OPT_0)] = f(OPT)$ and $\mathbb{E}[f(X_n)] = \mathbb{E}[f(Y_n)] = \mathbb{E}[f(OPT_n)]$, the claim follows:

$$\mathbb{E}[f(X_n)] = \mathbb{E}[f(Y_n)] \geq \frac{1}{2} f(OPT)\,.$$ □

The quality of these results becomes apparent when we look at **inapproximability results** for USM. Suppose we are given a submodular function by a value oracle. For any $\epsilon > 0$ an algorithm achieving a $(1/2 + \epsilon)$-approximation would require an exponential number of queries to the value oracle in terms of cardinality of ground set [9]. The same holds even if the given function is symmetric. Better approximation bounds are known for special cases of submodular functions.

5 Conclusion

We defined the notion of submodular functions and presented their characterization in the terms of discrete derivatives. Several examples of submodular functions were presented in order to show wide range of their applications. Submodular functions were presented as generalizations of several concepts in graph theory and as a tool for more specific areas of discrete mathematics. We presented a new construction of exact polymatroid separators of hypergraphs.

Motivated by examples, we turned to the problems of minimization and maximization of submodular functions. We stated (without proof) the most important results about minimization of submodular functions. The interested reader is recommended to find more details about mentioned algorithms in cited papers. Two problems of maximization of submodular functions were presented, namely an algorithm for maximization of submodular functions under cardinality constraint and an algorithm for unconstrained submodular function maximization. We proved that the greedy algorithm yields ≈ 0.63 approximation factor to the problem of maximization under cardinality constraint.

The problem of unconstrained submodular maximization was solved by three different algorithms. The random set algorithm has an approximation factor of $1/4$ in expected value. In the other two algorithms, we saw how a slight change from a deterministic algorithm, which has an approximation factor of $1/3$ can improve to an approximation factor of $1/2$ under expected value. Recently, a paper was submitted by Buchbinder and Feldman, in which a deterministic algorithm is presented with approximation factor $1/2$. This is optimal, unless $P = NP$ [6]. Further work on optimization of submodular functions includes finding faster algorithms, since both minimization and maximization problems are solved optimally in terms of approximation.

6 Povzetek naloge v slovenskem jeziku

Namen naloge je bil definirati pojem submodularne funkcije, predstaviti nekatere optimizacijske probleme, povezane z submodularnimi funkcijami, in obravnavati nekatere algoritme za maksimizacijo submodularnih funkcij.

V prvem poglavju je predstavljen pojem submodularne funkcije. Podana je karakterizacija s pomočjo diskretnih odvodov – marginalnega dobička. V tem smislu je funkcija f na končni možici V submodularna, če ima nepadajoči diskretni odvod. Diskretni odvod nam tudi pove, ali je funkcija monotona: funkcija je monotona natanko tedaj, ko so vsi njeni diskretni odvodi nenegativni.

Preprost primer monotone submodularne funkcije f si lahko ogledamo na množici $V = \{a, b\}$, če definiramo funkcijo kot $f(\emptyset) = 0$, $f(\{a\}) = f(\{b\}) = 2$ in $f(\{a, b\}) = 3$. Preprost razred submodularnih funkcij so tudi t.i. modularne funkcije, diskretni analog linearnih funkcij. Če za modularno funkcijo $f : 2^V \to \mathbb{R}$ predpostavimo, da je $f(\emptyset) = 0$, potem vrednost $f(S)$ lahko predstavimo kot vsoto uteži posameznih elementov množice S.

Dobro znani problem optimalne postavitve objektov (ang. *facility location problem*) smo modelirali kot problem maksimizacije prirejene submodularne funkcije. V zaključni nalogi je predstavljeno tudi, zakaj so submodularne funkcije posplošitev problema najmanjšega in največjega prereza v grafih. Definiran je pojem matroida, ki je posplošitev pojma linearne neodvisnosti v linearni algebri. Rang vsakega matroida pripada posebnemu razredu submodularnih funkcij, imenovanih polimatroidalne funkcije. V nalogi so omenjene številne aplikacije polimatroidalnih funkcij. Predstavljen je tudi nov primer konstrukcije eksaktnih polimatroidalnih separatorjev hipergrafov.

Iz obstoječih submodularnih funkcij lahko konstruiramo nove z uporabo operacij, omenjenih v Poglavju 3. Pokazali smo, da je vsaka nenegativna linearna kombinacija submodularnih funkcij tudi submodularna funkcija. Opisane konstrukcije nam pomagajo pri dokazovanju submodularnosti bolj zapletenih funkcij.

Motivirani s primeri smo si ogledali optimizacijo submodularnih funkcij. Najprej smo našteli rezultate in algoritme za minimizacijo. Videli smo, da je minimizacija submodularnih funkcij možna v polinomskem času.

Problem maksimizacije submodularnih funkcij je obravnavan v dveh različnih delih. Najprej smo definirali problem maksimizacije monotone submodularne funkcije pod pogojem, da velikost množice ne presega določenega števila k. Za rešitev tega problema smo predlagali požrešni algoritem, ki za začetno rešitev vzame prazno množico in ki v vsaki ponovitvi doda element, ki vrednost trenutne rešitve najbolj poveča. Če označimo z OPT vrednost optimalne rešitve tako definiranega problema, omenjeni algoritem (Algoritem 1) vrne rešitev, ki ima vrednost vsaj $\approx 0.63OPT$.

Drugi problem, ki smo ga obravnavali v zvezi z maksimizacijo submodularnih funkcij, je problem maksimizacije poljubne (ne nujno monotone) submodularne funkcije. V tem primeru želimo za dano submodularno funkcijo $f : 2^V \to \mathbb{R}$ najti množico $S \subseteq V$, ki maksimizira vrednost $f(S)$. Temu problemu smo na kratko rekli USM (ang. *Unconstrained Submodular Maximization*). Ogledali smo si tri algoritme za rešitev tega problema: Algoritem 2 – Naključna množica, Algoritem 3 – 1/3-aproksimacijski deterministični algoritem za USM – in Algoritem 4 – 1/2-aproksimacijski verjetnostni algoritem za USM.

Algoritem 2 deluje tako, da vsak element množice V doda v rešitev z verjetnostjo 1/2 neodvisno od ostalih elementov. Matematično upanje aproksimacijskega faktorja tako dobljene rešitve je 1/4.

Algoritem 3 je preprost algoritem za problem USM in teče v linearnem času. Množico V poljubno uredimo, $V = \{v_1, \ldots, v_n\}$. Začnemo z dvema množicama $X_0 = \emptyset$ in $Y_0 = V$. Za vsak element v_i algoritem naredi eno iteracijo in izračuna pripadajoči množici X_{i-1} in Y_{i-1}. Ideja algoritma je, da v i-ti iteraciji na podlagi diskretnega odvoda bodisi doda v_i v X_{i-1} ali pa odstrani v_i iz Y_{i-1}. To pomeni, da imamo $X_i \subset Y_i$ za vsak $i < n$ in $X_n = Y_n$, kar algoritem tudi vrne kot rešitev. Algoritem 3 zagotavlja rešitve s faktorjem aproksimacije 1/3.

Algoritem 4 je podoben Algoritmu 3, s to razliko da pri i-tem koraku, kjer se odločamo, ali bomo v_i dodali v X_{i-1} ali odstranili iz Y_{i-1}, dopuščamo naključnost na naslednji način. Definirajmo

$$a'_i := \max\{0, f(X_{i-1} \cup \{v_i\}) - f(X_{i-1})\},$$
$$b'_i := \max\{0, f(Y_{i-1} \setminus \{v_i\}) - f(Y_{i-1})\}.$$

Potem v i-tem koraku z verjetnostjo $a'_i/(a'_i + b'_i)$ dodamo element v_i v množico X_{i-1} ali pa (z verjetnostjo $b'_i/(a'_i + b'_i)$) odstranimo element v_i iz množice Y_{i-1}. Matematično upanje aproksimacijskega faktorja Algoritma 4 je 1/2.

Uporaba naključnosti torej izboljša aproksimacijski faktor iz vrednosti 1/3 na 1/2. Kvaliteta teh rešitev postane razvidna v luči dejstva, da je za vsak $\epsilon > 0$ aproksimacija problema USM z aproksimacijskim faktorjem $1/2 + \epsilon$ NP-težek problem.

7 Bibliography

[1] D.P. Bertsekas, A. Nedic, and A.E. Ozdaglar, Convex analysis and optimization, MA, 2003. *Athena Scientific, Belmont* MR2184037 (2006) . *(Cited on page 16.)*

[2] M. Bordewich and C. Semple, Budgeted nature reserve selection with diversity feature loss and arbitrary split systems. *Journal of mathematical biology* 64 (2012) 69 – 85. *(Cited on page 16.)*

[3] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan, An inequality for polymatroid functions and its applications. *Discrete Applied Mathematics* 131(2) (2003) 255–281. *(Cited on pages 1, 10, and 11.)*

[4] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan, Extending the Balas-Yu bounds on the number of maximal independent sets in graphs to hypergraphs and lattices. *Mathematical programming* 98 (2003) 355–368. *(Cited on page 10.)*

[5] E. Boros, P. Heggernes, P. van 't Hof, and M. Milanič, Vector connectivity in garphs. *Networks* 63.4 (2014) 277–285. *(Cited on page 16.)*

[6] N. Buchbinder and M. Feldman, Deterministic Algorithms for Submodular Maximization Problems. 2015, submitted. arXiv preprint arXiv:1508.02157 *(Cited on page 29.)*

[7] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz, A tight linear time (1/2)–approximation for uncostrained submodular maximization. *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science – FOCS 2012* (2012) 649–658. *(Cited on pages 1 and 23.)*

[8] J. Edmonds, *Submodular functions, matroids, and certain polyhedra*, Combinatorial Structures and their Applications, Gordon and Breach, London, 1970, pp. 69 – 87. *(Cited on pages 7, 10, and 16.)*

[9] U. Feige, V.S. Mirrokni, and J. Vondrák, Maximizing non–monotone submodular functions. *SIAM J. Comput.* 40(4) (2011) 1133–1153. *(Cited on pages 1, 21, and 28.)*

[10] L.R. Ford and D.R. Fulkerson, Maximal flow through a network. *Canadian journal of Mathematics* 8.3 (1956) 399–404. *(Cited on page 8.)*

[11] A.M. Frieze, A cost function property for plant location problems. *Mathematical Programming* 7 (1974) 245–248. *(Cited on page 6.)*

[12] S. Fujishige, Personal reminiscence: combinatorial and discrete optimization problems in which I have been interested. *Jpn. J. Ind. Appl. Math.* 29(3) (2012) 357–384. *(Cited on page 16.)*

[13] B. Goldengorin, Maximization of submodular functions: theory and enumeration algorithms. *European J. Oper. Res.* 198(1) (2009) 102–112. *(Cited on page 17.)*

[14] G. Grimmett and D. Welsh, *Probability: An Introduction*. Oxford University Press, Oxford, UK, 1986. *(Cited on page 3.)*

[15] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1987. *(Cited on page 16.)*

[16] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid algorithm and its consequences in combinatorial optimization. *Combinatorica* 1 (1981) 499 – 513. *(Cited on page 16.)*

[17] S. Iwata, A faster scaling algorithm for minimizing submodular functions. *SIAM J. Comput.* 32 (2002) 833 – 840. *(Cited on page 17.)*

[18] D. Kempe, J. Kleinberg, and E. Tardos, Maximizing the Spread of Influence through a Social Network. *KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003) 137–146. *(Cited on page 9.)*

[19] L. Khachiyan, E. Boros, K. Elbassioni, and V. Gurvich, Generating all minimal integral solutions to monotone $\vee$, $\wedge$-systems of linear, transversal and polymatroid inequalities. *Mathematical Foundations of Computer Science* 2005 (2005) 556–567. *(Cited on page 10.)*

[20] A. Krause and C. Guestrin, *Beyond Convexity – Submodularity in Machine Learning*, `http://submodularity.org/submodularity-slides.pdf`. (Viewed on: 4/7/2015.) *(Cited on pages 9 and 16.)*

[21] A. Krause and D. Golovin, Submodular Function Maximization. *Tractability: Practical Approaches to Hard Problems* 3 (2012) 71–104. *(Cited on pages 1, 13, 16, 17, and 18.)*

[22] J. Lee, Techniques for submodular maximization. *Discrete Geometry and Optimization, Fields Inst. Commun.* 69 (2013) 163–177. *(Cited on pages 1 and 16.)*

[23] Y.T. Lee, A. Sidford and S.C. Wong, A Faster Cutting Plane Method and its Implications for Combinatorial and Convex Optimization. 2015, submitted.arXiv preprint arXiv:1508.04874 *(Cited on page 17.)*

[24] L. Lovász, Submodular functions and convexity. *Mathematical programming: the state of the art* (1982) 235–257. *(Cited on pages 13, 15, and 16.)*

[25] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher, An analysis of approximations for maximizing submodular set functions I. *Math. Programming* 14 (1978) 177–188. *(Cited on pages 9, 17, and 19.)*

[26] J.B. Orlin, A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical programming* 118 (2, Ser. A) (2009) 237–251. *(Cited on page 17.)*

[27] A. Recski, *Matroid Theroy and its Applications in Electric Network Theory and in Statics*, Algorithms and Combinatorics,vol. 6. Springer, Berlin, 1989. *(Cited on page 16.)*

[28] A.P. Ruszczyński, *Nonlinear optimization*, Princeton university press, vol.13, 2006. *(Cited on page 16.)*

[29] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Combin. Theory* B 80 (2000) 346 – 355. *(Cited on page 17.)*

[30] A. Schrijver, Combinatorial Optimization. Polyhedra and Efficiency. In: Algorithms and Combinatorics , *Vol. B*, Springer-Verlag, 1995, Chapters 39–69. *(Cited on pages 6, 9, and 10.)*

[31] J. 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*, ACM, 2008.67 – 74 *(Cited on page 17.)*

[32] J. Vygen, A note on Schrijver's submodular function minimization algorithm. *J. Combin. Theory* B 88 (2003) 399 – 402. *(Cited on page 17.)*

[33] D.B. West, *Introduction to Graph Theory, Vol. 2.*, Upper Saddle River: Prentice Hall, 2001. *(Cited on page 7.)*

[34] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica* 2(4) (1982) 385–393. *(Cited on pages 9 and 20.)*