Beyond Submodular Maximization:

One-Sided Smoothness and Meta-Submodularity

by

Mehrdad Ghadiri

B.Sc., Sharif University of Technology, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2019

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Beyond Submodular Maximization: One-Sided Smoothness and Meta-Submodularity

submitted by Mehrdad Ghadiri in partial fulfillment of the requirements for

the degree of Master of Science

in Computer Science

Examining Committee:

Bruce Shepherd, Computer Science
Co-supervisor

Mark Schmidt, Computer Science
Co-supervisor

Nicholas Harvey, Computer Science
Additional Examiner

Abstract

While there are well-developed tools for maximizing a submodular function $f(S)$ subject to a matroid constraint $S \in \mathcal{M}$, there is much less work on the corresponding supermodular maximization problems. We develop new techniques for attacking these problems inspired by the continuous greedy method applied to the multi-linear extension of a submodular function. We first adapt the continuous greedy algorithm to work for general twice-continuously differentiable functions. Our results are based on a new notion of one-sided smoothness of an objective. Reminiscent of how Lipschitz smoothness bounds convergence rates in convex optimization, one-sided smoothness controls the approximability of maximizing a monotone, non-linear function. If $F : [0,1]^n \to \mathbb{R}_{\geq 0}$ is one-sided σ-smooth, then it yields an approximation factor depending only on σ. We apply the new algorithm to a broad class of quadratic supermodular functions arising in diversity maximization. We also develop new methods for rounding quadratics over a matroid polytope. These are based on extensions to swap rounding and approximate integer decomposition. Together with the adapted continuous greedy this leads to a $O(\sigma^{3/2})$-approximation. This is the best asymptotic approximation known for this class of diversity maximization and we give some evidence for why we believe it may be tight.

We then consider general (non-quadratic) functions. We give a broad parameterized family of monotone functions which include submodular functions and the just-discussed supermodular family of discrete quadratics. The new family is defined by restricting the one-sided smoothness condition to the boolean hypercube; such set functions are called *γ-meta-submodular*. We develop local search algorithms with approximation factors that depend only on γ. We show that the γ-meta-submodular families include well-known function classes including meta-submodular functions ($\gamma = 0$), proportionally submodular ($\gamma = 1$), and diversity functions based on negative-type distances or Jensen-Shannon divergence (both $\gamma = 2$) and (semi-)metric diversity functions.

We then focus on maximizing a specific 1-meta-submodular function in a distributed setting. This has applications in machine learning and recommender systems. As an application, we model the multi-label feature selection problem as such an optimization problem. This combined with our optimization algorithm leads to the first distributed multi-label feature selection method.

Lay Summary

In a wide variety of applications, one needs to find the (near) best value for a function. Submodular functions are a class of functions for which we can find such near best value. They have many applications in many different areas, including, but not limited to, machine learning, game theory, and automatic summarization. Maybe the most important reason for their nice behaviour is the diminishing return property. In this work, we investigate the optimization for functions that do not satisfy this property.

Preface

Chapters 2, 3, and 4 are based on a joint work with Richard Santiago and Bruce Shepherd. A version of this has been published on arXiv [34]. Chapter 5 is based on a joint work with Mark Schmidt and a version of it is published in AISTATS 2019 [33].

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Symbols

The next list describes several symbols that will be later used within the body of the document.

$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Real coordinate space of n dimensions
$\mathbb{E}[X]$	Expected value of X
$[n]$	$\{1, \ldots, n\}$
x^T	Transpose of the vector x
A^T	Transpose of the matrix A
$\mathbb{1}_S$	Characteristic vector of set S
x_i or $x(i)$	Coordinate i of the vector x
$\nabla F(x)$	Gradient of the function F at x
$\nabla^2 F(x)$	Hessian of the function F at x
$A(i, j)$	Entry i and j of the matrix A
$\nabla^2_{ij} F(x)$	Entry i and j of the Hessian. The second derivative of F with respect to x_i and x_j.
$\nabla_i F(x)$	Coordinate i of the gradient. The first derivative of F with respect to x_i.
$x \geq 0$ $(x > 0)$	$\forall i \in [n]$, $x_i \geq 0$ $(x_i > 0)$
$x(S)$	$\sum_{i \in S} x(i)$
$\|x\|_1$	$\sum_{i \in [n]} x(i)$
$supp(x)$	$\{i : x(i) \neq 0\}$
$\mathbf{0}$	The zero vector
$\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$	The standard unit vectors of $\mathbb{R}^n$
$x \vee y$	Coordinate-wise maximum of x and y

Acknowledgements

First of all, I would like to thank my supervisors Bruce Shepherd and Mark Schmidt whose guidance helped me in this research. This was a great opportunity for me to learn more about theoretical computer science, mathematical optimization, and machine learning from them. I am also very grateful to my examining committee member, Nick Harvey who took the time to read this thesis.

Next I would like to thank my lab-mates Coulter Beeson, Chris Liaw, Taylor Lundy, Sikander Randhawa, and Richard Santiago who created a nice environment of learning and research.

I am grateful for having friends who had a direct impact on my path and decisions: Abbas, Alireza, Reza, Saeed, Sahand, Samira, Sina, Yahya, and Zeinab. I am also very thankful to my friends in Vancouver who created a lot of fun and enjoyment: Ainaz, Ali, Amin, Borna, Farnoosh, Farzad, Marjan, Negar, Niki, Saeid, and Setareh.

Last but not least, I would like to express my gratitude to my family who always supported me during this degree and beyond: Morteza, Fatemeh, Mohammad, and Niloufar.

Dedication

To Leonhard, John, and Paul.

Chapter 1

Introduction

In optimization problems, given a function $f : \mathcal{D} \to \mathbb{R}$ and $\mathcal{C} \subseteq \mathcal{D}$, the goal is to find the element $x \in \mathcal{C}$ to *maximize/minimize* the function in $\mathcal{C}$. Optimization problems arise in many areas including but not limited to machine learning, scheduling, and resource allocation.

Optimization has many branches. Two of the most studied of these branches are *linear optimization* and *convex optimization*. A linear optimization problem in canonical form is stated as follows.

$$\begin{aligned} \text{maximize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ \text{and} \quad & x \geq 0, \end{aligned} \tag{1.1}$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, and the domain of x is $\mathbb{R}^n$. We call $c^T x$ the *objective function*; $Ax \leq b$ and $x \geq 0$ are called the *constraints*. We say the point x is feasible if $Ax \leq b$ and $x \geq 0$. The set of all feasible points is called the *feasible region/area*. A linear optimization problem is polynomial time solvable if there is a polynomial time separation oracle which given x, either confirms that x satisfies all the constraints or it returns one of the violated constraints (inequalities) [35, 36]. If such a separation oracle exists, for any $\epsilon > 0$, standard methods can find a solution whose difference with the optimal solution is at most ϵ. We call this an *additive approximation*.

Linear optimization is a subclass of the more general class of convex optimization problems. A function is called convex if

$$\frac{f(x) + f(y)}{2} \geq f(\frac{x+y}{2}), \tag{1.2}$$

for any x and y in the domain of f. For a vector z and function f, we say that f is *convex in z direction* if for any x and $x + \lambda z$ ($\lambda \geq 0$) in the domain of f,

$$\frac{f(x) + f(x + \lambda z)}{2} \geq f(\frac{x + (x + \lambda z)}{2}).$$

We define the concavity in a similar way as convexity. We say that a function f is *concave* if

$$\frac{f(x) + f(y)}{2} \leq f(\frac{x+y}{2}), \tag{1.3}$$

for any x and y in the domain of f. Similarly, we define the concavity in a direction. We say that a function is convex/concave in forward directions if it is convex/concave in z direction for any $z \geq 0$.

A function f is affine if there exists $c \in \mathbb{R}^n$ and $d \in \mathbb{R}$ that $f(x) = c^T x + d$. It is easy to check that any affine function is convex. A convex optimization problem in standard form is stated as the following.

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0, \ i = 1, \dots, m \\ \text{and} \quad & h_i(x) = 0, \ i = 1, \dots, p, \end{aligned} \tag{1.4}$$

where f and g_i's are convex functions and h_i's are affine functions. Many classes of convex optimization problems admit polynomial time algorithms that find an additive approximation.

Linear optimization and convex optimization problems possess a lot of nice properties which are exploited to find solutions very close to the optimal solution. Perhaps the most fundamental property is that the feasible region and the objective function are continuous. This allows, for example, the use of *gradient methods*. In contrast when the domain of the objective function is discrete, it is not possible to use gradient methods in a straitforward way. Problems of this form are considered *discrete optimization* problems. One important subclass of these problems is *combinatorial optimization* in which the domain of the objective function is the power set (i.e., the set of all subsets) of a ground set. Such functions are called *set functions*.

A combinatorial optimization problem is usually stated as the following

$$\begin{aligned} \text{maximize} \quad & f(S) \\ \text{subject to} \quad & S \in \mathcal{I}, \end{aligned} \tag{1.5}$$

where $f : 2^{[n]} \to \mathbb{R}$ and $\mathcal{I} \subseteq 2^{[n]}$. Usually $\mathcal{I}$ is a combinatorial family of subsets (e.g., a matroid).

Modular (linear) functions are extensively studied in the context of combinatorial optimization. A function $f : 2^{[n]} \to \mathbb{R}$ is modular if

$$f(S) = \sum_{s \in S} f(\{s\}), \tag{1.6}$$

for any $S \in 2^{[n]}$. The *traveling salesman* problem is an example of a combinatorial optimization problem with a modular objective function. In this problem f is defined on the power set of the set of edges of a complete graph $G = (V, E)$. Each edge $e \in E$ has a weight. Because f is modular, these weights determine f. In the traveling salesman problem, the family of feasible sets $\mathcal{I}$ is all the cycles of size $|V|$. One can see that the size of $\mathcal{I}$ is exponential in terms of $|V|$ and $|E|$ and we suspect that this problem is not easy to solve. This suspicion is actually correct in the sense that it has been shown that the traveling salesman problem is NP-hard.

Many problems in combinatorial optimization are NP-hard and therefore there is not any known polynomial time algorithm that can find the optimal solution. In the absence of such an algorithm, combinatorial optimization research has often focused on *multiplicative approximation algorithms* or in short approximation algorithms. Let $\mathcal{P}$ be an optimization problem with a non-negative objective function $f : 2^{[n]} \to \mathbb{R}_{\geq 0}$ and for an instance of $\mathcal{P}$ like P, let $\texttt{OPT}(P)$ be the optimal solution of P and $\texttt{ALG}(P)$ be the output of algorithm $\texttt{ALG}$ on P. We call $\texttt{ALG}$ an α-approximation algorithm for $\mathcal{P}$ if for any instance P, its returned solution satisfies the following.

$$\frac{1}{\alpha} f(\texttt{ALG}(P)) \leq f(\texttt{OPT}(P)) \leq \alpha f(\texttt{ALG}(P)) \tag{1.7}$$

The reason for defining such a multiplicative approximation is that many combinatorial optimization problems do not admit an additive approximation (e.g., the traveling salesman problem).

1.1 Submodular Functions and Matroids

Submodular functions are another class of set functions that are extensively studied in combinatorial optimization. Recently, many applications are found for these functions stemming from machine learning, social networks, recommendation systems, etc [53]. A function $f : 2^{[n]} \to \mathbb{R}$ is called *submodular* if

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T), \tag{1.8}$$

for any $S, T \subseteq [n]$. It is easy to see that any modular function is also a submodular function. Similarly we say that a function is *supermodular* if

$$f(S) + f(T) \leq f(S \cup T) + f(S \cap T), \tag{1.9}$$

for any $S, T \subseteq [n]$. It has been shown that a function is submodular if and only if

$$f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T), \tag{1.10}$$

for any $S \subseteq T \subseteq [n]$ and $i \in [n] \setminus T$ [70]. This is called the *diminishing return property.* There are many examples of set functions that admit this diminishing return property. One important example is the class of coverage functions which includes the set cover problem, vertex cover problem, etc.

This property also occurs in a class of combinatorial structures called *matroids.* A pair $\mathcal{M} = ([n], \mathcal{I})$ is a matroid if $\mathcal{I} \subseteq 2^{[n]}$ is a family of subsets that satisfies the following conditions: 1) if $S \in \mathcal{I}$ and $T \subseteq S$ then $T \in \mathcal{I}$ (*hereditary property*); 2) if $S, T \in \mathcal{I}$ and $|S| < |T|$ then there exist $i \in T \setminus S$ such that $S \cup \{i\} \in \mathcal{I}$ (*exchange property*) [71].

The subsets in $\mathcal{I}$ are called *independent sets* of the matroid $\mathcal{M}$ and the subsets in $2^{[n]} \setminus \mathcal{I}$ are called *dependent sets* of $\mathcal{M}$. A *base* of $\mathcal{M}$ is a maximal independent set. A *circuit* of $\mathcal{M}$ is a minimal dependent set. By the exchange property, it is easy to show that all the bases of a matroid have an equal size. The size of a base is called the *rank* of the matroid $\mathcal{M}$. We define the *rank function* $r : 2^{[n]} \to \mathbb{Z}_{\geq 0}$ of a matroid as the following. For any $S \subseteq [n]$, $r(S)$ is equal to the size of the largest independent set of the matroid which is a subset of S. It has been shown that the rank function of a matroid is submodular [71].

1.2 Continuous Relaxations of Set Functions

As mentioned, working in a continuous space has many advantages. Because of this, it is a common practice in combinatorial optimization to transform the problem to a continuous space, find the optimum solution (or a solution close to the optimum) in the continuous space, and transform back the continuous solution to the discrete space. This last step is often called the *rounding*.

Usually this transformation to the continuous space is done by using a continuous relaxation of the set function. Let $f : \{0,1\}^{[n]} \to \mathbb{R}$ (or $f : 2^{[n]} \to \mathbb{R}$) be a set function. The function $F : [0,1]^{[n]} \to \mathbb{R}$ is a *continuous relaxation* (or *continuous extension*) of f if

$$F(\mathbb{1}_S) = f(S), \tag{1.11}$$

for any $S \subseteq [n]$. Two important continuous relaxations considered in the study of submodular functions are the *Lovasz extension* and the *multi-linear extension*.

The Lovasz extension of a set function is defined as

$$F^L(x) = \mathbb{E}[f(\{i : x_i \geq \lambda\})], \tag{1.12}$$

where the expectation is over λ sampled from a uniform distribution on $[0,1]$. It is shown that F^L is convex if and only if f is submodular [60]. This relaxation is usually used for the problem of minimizing a submodular function. The multi-linear extension is another important continuous relaxation of set functions which is used for submodular maximization and plays an important role in our results.

Definition 1 (Multi-linear extension)**.** *Let $p_x(R)$ be the probability of picking the set R with respect to x if each element $v \in [n]$ is picked independently with probability x_v. In other words*

$$p_x(R) = \prod_{v \in R} x_v \prod_{v \in [n] \setminus R} (1 - x_v). \tag{1.13}$$

Then the multi-linear extension of $f : 2^{[n]} \to \mathbb{R}$ is

$$F(x) = \sum_{R \subseteq [n]} f(R) p_x(R) = \mathbb{E}_{R \sim x}[f(R)]. \tag{1.14}$$

The multi-linear extension can be viewed as the expected value of the function if the input set is picked randomly with respect to x. In general, the multi-linear extension of a submodular function is neither convex nor concave. However, it admits convexity/concavity properties in specific directions which has been exploited for the maximization of submodular functions. More specifically, it is concave in the forward directions and for any $i, j \in [n]$, it is convex in $\boldsymbol{e}_i - \boldsymbol{e}_j$ direction [18].

1.3 Convex Polytopes

In addition to extending the domain of the function to a continuous space, we need to extend the feasible (search) region to the continuous space. For this, we first need some notations. We define the *characteristic vector* of a subset $S \subseteq [n]$ as the following.

$$\mathbb{1}_S := \begin{cases} (\mathbb{1}_S)_i = 1, & i \in S \\ (\mathbb{1}_S)_i = 0, & i \notin S \end{cases}$$

With this, we can consider the feasible subsets as points/vectors in the continuous space $[0,1]^{[n]}$ but these points still form a discrete set. The *convex hull* of a set of points like Q is defined as the following.

$$C_Q := \{\sum_{p\in Q} \lambda_p p : \sum_{p\in P} \lambda_p = 1, \lambda \geq 0\}$$

A *convex polytope* is the convex hull of finitely many points. A convex polytope also can be represented as the intersection of finitely many half-spaces. For brevity, we call a convex polytope just a polytope. For a family of subsets $\mathcal{I}$, we call the convex hull of the characteristic vectors of elements of $\mathcal{I}$ its corresponding polytope. This polytope is the continuous extension of the feasible space we use in this work but in general, any convex subset of $\mathbb{R}^{[n]}$ that contains the corresponding polytope is a convex continuous extension of the feasible space. A polytope $\mathcal{Q}$ is a convex corner or a downwards-closed polytope if it satisfies the following conditions. 1) If $x \in \mathcal{Q}$ then $x \geq 0$; 2) If $x \in \mathcal{Q}$ and $0 \leq y \leq x$ then $y \in \mathcal{Q}$. If $\mathcal{I}$ has the hereditary property, its corresponding polytope is a downwards-closed polytope.

1.4 Motivation

In the past decade, the catalogue of algorithms available to combinatorial optimizers has been substantially extended to new settings which allow submodular objective functions. For instance, while classical work [31, 63, 64] already established a $\frac{1}{2}$-approximation for maximizing a non-negative monotone submodular function subject to a matroid constraint, it was not until recently when the work from [18, 81] achieved a tight $(1-\frac{1}{e})$-approximation for this problem. The latter required the development of new continuous optimization machinery for the associated multi-linear relaxation. These developments in submodular maximization were occurring at the same time that researchers found a wealth of new applications for these models [16, 28, 42, 46, 49, 54, 57, 58, 67, 76].

The related supermodular maximization models (submodular minimization) also offer an abundance of applications, but they appeared to be highly intractable even under simple cardinality constraints [77]. One exception came from a specific model for diversity maximization. Given a set function $f(S)$ which measures the 'diversity' amongst elements of a set S, a problem of broad interest is to find a set S of maximum diversity subject to a prescribed bound on its cardinality $|S| \leq k$, or more generally, subject to a matroid $\mathcal{M}$ constraint:

$$(DivMax) \qquad \max\{f(S) : S \in \mathcal{M}\}.$$

One class of diversity functions that has wide applications in machine learning are the so-called *remote-clique functions* [1, 33, 85]. These are based on having a dis-similarity measure $d(u,v)$ between each pair of objects u, v in the ground set. The corresponding max-sum problem is then to maximize $f(S) := \sum_{u,v\in S} A(u,v)$ [21, 51]. If $A(u,v) \geq 0$, then one easily checks that f is supermodular. We sometimes abuse nomenclature and conflate A with its associated diversity function f. These functions are essentially a special case of what we term *discrete quadratic functions*. Namely, a function which is the restriction of a quadratic $\frac{x^TAx}{2} + b^Tx$ to the boolean hypercube (A is symmetric, non-negative, 0-diagonal, and $b \geq 0$). Our results regarding these functions is of potential interest for non-convex quadratic programming.

Discrete quadratic diversity functions are a very broad family and the associated problem DivMax is ostensibly intractable in the sense that it includes the densest subgraph problem [11]. However, for *metric diversity functions* (remote-clique function when A forms a metric), there is a 2-approximation subject to a cardinality constraint [39, 68]. Moreover, this has been generalized to the case of matroid constraint [1, 11]. They give a 10.22-approximation for maximizing these functions subject to a matroid constraint. In [11], Borodin et al consider the maximization of the sum of a monotone submodular function and a *metric* diversity function subject to a matroid constraint. They show that the local search algorithm achieves a 2-approximation. Their technique carries over to the weaker notion of σ*-semi-metric* diversity functions (that is, satisfying a σ-approximate triangle inequality for $\sigma \geq 1$); in [84] this analysis is shown to yield a 2σ-approximation under a cardinality constraint and a $2\sigma^2$-approximation under a matroid constraint. More generally, Borodin et al. [12, 13] introduce the class of *proportionally submodular* (monotone) functions which include these metric diversity functions as well as monotone submodular functions. Ultimately, we define the parameterized class of γ-meta submodular funtcions which includes these well-known function classes including meta-submodular functions ($\gamma = 0$), proportionally submodular ($\gamma = 1$), and diversity functions based on negative-type distances or Jensen-Shannon divergence (both $\gamma = 2$) and (semi-)metric diversity functions.

The preceding results motivate the key impetus for our work, namely, to explain and explore the reasons for the fortunate cases when supermodular maximization is actually tractable. We argue that a one-sided smoothness parameter governs the degree to which we can approximate these problems. Two driving questions become: (Q1) *Find a parameterized family of supermodular functions which contains metric, and more generally σ-semi-metric, diversity functions and remains tractable in terms of σ.* (Q2) A second motivating question is *to find a parameterized tractable family of monotone set functions which includes all monotone submodular functions and the aforementioned diversity functions.*

In Chapter 2, we investigate these questions from a continuous optimization perspective. In Chapter 3, we connect these with the corresponding discrete optimization problems by presenting rounding algorithms. In Chapter 4, we focus on discrete algorithms but interestingly, we use continuous properties the analysis of these algorithms. In Chapter 5, we study the maximization of a special class of our functions in the distributed setting and investigate one of its applications.

1.5 Preliminaries

In this section, we discuss some basic properties of set functions and their multi-linear extensions which are widely used in this work. The proofs are mainly algebraic and the reader can skip them.

We widely use the gradient and the Hessian of the multi-linear extension in the analysis of our results. The following lemma gives a description of the gradient and the Hessian at point x.

Lemma 1 ([81])**.** *Let F be the multi-linear extension of a set function f. Then*

$$\nabla_i F(x) = \sum_{R\subseteq[n]-i} \big[(f(R+i)-f(R)) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i)} (1-x_v)\big], \tag{1.15}$$

and

$$\nabla^2_{ij} F(x) = \sum_{R\subseteq[n]-i-j} [f(R+i+j)-f(R+i)-f(R+j)+f(R)] \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i+j)} (1-x_v) \tag{1.16}$$

Proof. One can see that

$$\begin{aligned} F(x) &= \sum_{R\subseteq[n]} f(R)p_x(R) \\ &= x_i \sum_{R\subseteq[n]-i} \big[f(R+i) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i)} (1-x_v)\big] \\ &+ (1-x_i) \sum_{R\subseteq[n]-i} \big[f(R) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i)} (1-x_v)\big]. \end{aligned}$$

Therefore we have

$$\nabla_i F(x) = \sum_{R\subseteq[n]-i} \big[(f(R+i)-f(R)) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i)} (1-x_v)\big].$$

For the second part of the lemma, we use a similar formulation of $F(x)$.

$$\begin{aligned} F(x) = x_i x_j &\sum_{R\subseteq[n]-i-j} f(R+i+j) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i+j)} (1-x_v) \\ + x_i(1-x_j) &\sum_{R\subseteq[n]-i-j} f(R+i) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i+j)} (1-x_v) \\ + (1-x_i)x_j &\sum_{R\subseteq[n]-i-j} f(R+j) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i+j)} (1-x_v) \\ + (1-x_i)(1-x_j) &\sum_{R\subseteq[n]-i-j} f(R) \prod_{v\in R} x_v \prod_{v\in[n]\setminus(R+i+j)} (1-x_v) \end{aligned}$$

This formulation implies the second part of the lemma. □

The following lemma gives a description of the gradient and the Hessian on integral points.

Lemma 2. *Let F be the multi-linear extension of a set function f. Then*

$$\nabla_i F(\mathbb{1}_S) = f(S+i) - f(S-i), \tag{1.17}$$

and

$$\begin{aligned} \nabla^2_{ij} F(\mathbb{1}_S) &= \nabla_j F(\mathbb{1}_{S+i}) - \nabla_j F(\mathbb{1}_{S-i}) \\ &= f(S+i+j) - f(S+i-j) - f(S-i+j) + f(S-i-j) \end{aligned} \tag{1.18}$$

Proof. When $x = \mathbb{1}_S$, the only non-zero summand of (1.15) in Lemma 1 is for $R = S - i$. Therefore

$$\nabla_i F(\mathbb{1}_S) = f((S-i)+i) - f(S-i).$$

The result is concluded by noting that $S + i = (S - i) + i$. For the second part, we can see that when $x = \mathbb{1}_S$, the only non-zero summand of (1.16) in Lemma 1 is for $R = S - i - j$. Therefore

$$\nabla^2_{ij} F(\mathbb{1}_S) = f((S-i-j)+i+j) - f((S-i-j)+i) - f((S-i-j)+j) + f(S-i-j).$$

Checking that $(S-i-j)+i+j = S+i+j$, $(S-i-j)+i = S+i-j$, and $(S-i-j)+j = S-i+j$ concludes the result. $\square$

For the brevity of notation, we define the following.

Definition 2. *We define $B_i(S) := \nabla_i F(\mathbb{1}_S) = f(S+i) - f(S-i)$ which is called the marginal gain of adding i to S. We also define*

$$A_{ij}(S) := \nabla^2_{ij} F(\mathbb{1}_S) = f(S+i+j) - f(S+i-j) - f(S-i+j) + f(S-i-j),$$

which is called a second-order difference for $i, j \in [n]$.

One can see that a function is submodular if and only if $A_{ij}(S) \leq 0$ for any $i, j \in [n]$, and $S \subseteq [n]$ [71]. The following result formulates the gradient and the Hessian of the multi-linear extension in terms of the new notation.

Corollary 1. *Let F be the multi-linear extension of a set function f. Then*

$$\nabla_i F(x) = \sum_{R \subseteq [n]} B_i(R) p_x(R),$$

and

$$\nabla^2_{ij} F(x) = \sum_{R \subseteq [n]} A_{ij}(R) p_x(R)$$

Proof. First note that $B_i(R+i) = B_i(R)$. Now by Lemma 1 we have

$$\begin{aligned}
\nabla_i F(x) &= \sum_{R \subseteq [n]-i} (f(R+i) - f(R)) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i)} (1 - x_v) \\
&= x_i \sum_{R \subseteq [n]-i} (f(R+i) - f(R)) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i)} (1 - x_v) \\
&\quad + (1 - x_i) \sum_{R \subseteq [n]-i} (f(R+i) - f(R)) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i)} (1 - x_v),
\end{aligned}$$

where the second equality holds because $x_i + (1 - x_i) = 1$. Hence,

$$\begin{aligned}
\nabla_i F(x) &= \sum_{R \subseteq [n]-i} (f(R+i) - f(R)) \prod_{v \in R+i} x_v \prod_{v \in [n] \setminus (R+i)} (1 - x_v) \\
&+ \sum_{R \subseteq [n]-i} (f(R+i) - f(R)) \prod_{v \in R} x_v \prod_{v \in [n] \setminus R} (1 - x_v) \\
&= \sum_{R \subseteq [n]-i} B_i(R+i) p_x(R+i) + \sum_{R \subseteq [n]-i} B_i(R) p_x(R) \\
&= \sum_{R \subseteq [n]} B_i(R) p_x(R).
\end{aligned}$$

For the second part, we note that $x_i x_j + (1 - x_i) x_j + x_i (1 - x_j) + (1 - x_i)(1 - x_j) = 1$, and $A_{ij}(R + i + j) = A_{ij}(R + i) = A_{ij}(R + j) = A_{ij}(R)$. Therefore by Lemma 1, we have

$$\begin{aligned}
\nabla^2_{ij} F(x) &= \sum_{R \subseteq [n]-i-j} (f(R+i+j) - f(R+i) - f(R+j) + f(R)) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i+j)} (1 - x_v) \\
&= x_i x_j \sum_{R \subseteq [n]-i-j} A_{ij}(R+i+j) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i+j)} (1 - x_v) \\
&+ (1 - x_i) x_j \sum_{R \subseteq [n]-i-j} A_{ij}(R+j) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i+j)} (1 - x_v) \\
&+ x_i (1 - x_j) \sum_{R \subseteq [n]-i-j} A_{ij}(R+i) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i+j)} (1 - x_v) \\
&+ (1 - x_i)(1 - x_j) \sum_{R \subseteq [n]-i-j} A_{ij}(R) \prod_{v \in R} x_v \prod_{v \in [n] \setminus (R+i+j)} (1 - x_v).
\end{aligned}$$

Hence,

$$\begin{aligned}
\nabla^2_{ij} F(x) &= \sum_{R \subseteq [n]-i-j} A_{ij}(R+i+j) \prod_{v \in R+i+j} x_v \prod_{v \in [n] \setminus R} (1 - x_v) \\
&+ \sum_{R \subseteq [n]-i-j} A_{ij}(R+j) \prod_{v \in R+j} x_v \prod_{v \in [n] \setminus (R+i)} (1 - x_v) \\
&+ \sum_{R \subseteq [n]-i-j} A_{ij}(R+i) \prod_{v \in R+i} x_v \prod_{v \in V \setminus (R+i)} (1 - x_v) \\
&+ \sum_{R \subseteq [n]-i-j} A_{ij}(R) \prod_{v \in R} x_v \prod_{v \in [n] \setminus R} (1 - x_v),
\end{aligned}$$

and therefore

$$
\begin{aligned}
\nabla^2_{ij} F(x) = & \sum_{R \subseteq [n]-i-j} A_{ij}(R+i+j) p_x(R+i+j) + \sum_{R \subseteq [n]-i-j} A_{ij}(R+j) p_x(R+j) \\
& + \sum_{R \subseteq [n]-i-j} A_{ij}(R+i) p_x(R+i) + \sum_{R \subseteq [n]-i-j} A_{ij}(R) p_x(R) \\
& = \sum_{R \subseteq [n]} A_{ij}(R) p_x(R).
\end{aligned}
$$

□

The following result describes the connection between the terms A_{ij} and B_i. One can see it as a discrete integral formula.

Lemma 3. *Let $f : 2^{[n]} \to \mathbb{R}$ be a normalized set function (i.e., $f(\emptyset) = 0$), $i \in [n]$, and $R = \{v_1, \dots, v_r\} \subseteq [n]$. Moreover, let $R_m = \{v_1, \dots, v_m\}$ for $1 \leq m \leq r$ and $R_0 = \emptyset$. Then*

$$
B_i(R) = f(\{i\}) + \sum_{j=1}^{r} A_{iv_j}(R_{j-1}). \tag{1.19}
$$

Proof. First, we consider the case where $i \notin R$. Then $B_i(R) = f(R+i) - f(R)$ and the right hand side of (1.19) is equal to

$$
\begin{aligned}
& f(R_{r-1}+i+v_r) - f(R_{r-1}-i+v_r) - f(R_{r-1}+i-v_r) + f(R_{r-1}-i-v_r) \\
& + f(R_{r-2}+i+v_{r-1}) - f(R_{r-2}-i+v_{r-1}) - f(R_{r-2}+i-v_{r-1}) + f(R_{r-2}-i-v_{r-1}) \\
& + \cdots \\
& + f(R_1+i+v_2) - f(R_1-i+v_2) - f(R_1+i-v_2) + f(R_1-i-v_2) \\
& + f(R_0+i+v_1) - f(R_0-i+v_1) - f(R_0+i-v_1) + f(R_0-i-v_1) \\
& + f(\{i\}) \\
& = f(R+i) - f(R) - f(R_{r-1}+i) + f(R_{r-1}) \\
& + f(R_{r-1}+i) - f(R_{r-1}) - f(R_{r-2}+i) + f(R_{r-2}) \\
& + \cdots \\
& + f(R_2+i) - f(R_2) - f(R_1+i) + f(R_1) \\
& + f(R_1+i) - f(R_1) - f(R_0+i) + f(R_0) \\
& + f(\{i\}) \\
& = f(R+i) - f(R)
\end{aligned}
$$

The last equality holds because the third and the fourth elements of each line cancel out the first and the second element of the next line (except for the last two lines), respectively. For the last two lines, note that $f(R_0) = f(\emptyset) = 0$ and $f(R_0+i) = f(\{i\})$.

Now, we consider the case that $i \in R$. Let $i = v_j$. Then $B_i(R) = f(R) - f(R - i)$ and the right hand side of (1.19) is equal to

$$\begin{aligned}
&f(R_{r-1} + i + v_r) - f(R_{r-1} - i + v_r) - f(R_{r-1} + i - v_r) + f(R_{r-1} - i - v_r) \\
&+ f(R_{r-2} + i + v_{r-1}) - f(R_{r-2} - i + v_{r-1}) - f(R_{r-2} + i - v_{r-1}) + f(R_{r-2} - i - v_{r-1}) \\
&+ \cdots \\
&+ f(R_j + i + v_{j+1}) - f(R_j - i + v_{j+1}) - f(R_j + i - v_{j+1}) + f(R_j - i - v_{j+1}) \\
&+ f(R_{j-1} + i + v_j) - f(R_{j-1} - i + v_j) - f(R_{j-1} + i - v_j) + f(R_{j-1} - i - v_j) \\
&+ f(R_{j-2} + i + v_{j-1}) - f(R_{j-2} - i + v_{j-1}) - f(R_{j-2} + i - v_{j-1}) + f(R_{j-2} - i - v_{j-1}) \\
&+ \cdots \\
&+ f(R_1 + i + v_2) - f(R_1 - i + v_2) - f(R_1 + i - v_2) + f(R_1 - i - v_2) \\
&+ f(R_0 + i + v_1) - f(R_0 - i + v_1) - f(R_0 + i - v_1) + f(R_0 - i - v_1) \\
&+ f(\{i\})
\end{aligned}$$

Now note that for any set S, $S + i - i \neq= S - i + i$. Hence the right hand side of (1.19) is equal to

$$\begin{aligned}
&f(R) - f(R - i) - f(R_{r-1}) + f(R_{r-1} - i) \\
&+ f(R_{r-1}) - f(R_{r-1} - i) - f(R_{r-2}) + f(R_{r-2} - i) \\
&+ \cdots \\
&+ f(R_{j+1}) - f(R_{j+1} - i) - f(R_j) + f(R_{j-1}) \\
&+ f(R_j) - f(R_j) - f(R_{j-1}) + f(R_{j-1}) \\
&+ f(R_j) - f(R_{j-1}) - f(R_{j-2} + i) + f(R_{j-2}) \\
&+ \cdots \\
&+ f(R_2 + i) - f(R_2) - f(R_1 + i) + f(R_1) \\
&+ f(R_1 + i) - f(R_1) - f(R_0 + i) + f(R_0) \\
&+ f(\{i\}) \\
&= f(R) - f(R - i).
\end{aligned}$$

Like before the equality holds because the last two terms of each line cancels out the first two terms of the next line except for the last two lines, the first $f(R_j)$ line and the $f(R_{j+1})$ line. The terms of the first $f(R_j)$ line cancel each other out, while the last two terms of the $f(R_{j+1})$ line cancel the first two terms of the second $f(R_j)$ line. $\square$

1.6 Our Contributions

In Chapter 2, we introduce the notion of *one-sided smoothness* which generalizes the *convexity in the forward direction* property of the multi-linear extension of submodular functions. We first see a couple of examples for these functions. Then we show this property is enough to find an approximate solution for the maximum of an arbitrary monotone continuous function subject to a downwards-closed polytope. We improve our approximation factor for a class of functions that admit non-positive third-order derivatives. We then investigate this newly introduced smoothness property on the sub-domains of the function. We show that the multi-linear extension of any monotone set function satisfies the one-sided smoothness condition on some sub-domain of the function and using this, we present an approximation algorithm for maximizing such multi-linear extension. We also investigate the sub-domain smoothness for the class of meta-submodular functions defined by Kleinberg et al.

In Chapter 3, we mostly focus on the integrality gap of the multi-linear extension of one-sided smooth functions and also rounding algorithms for finding integral solutions. We first investigate the integrality gap of the multi-linear extension of a general one-sided smooth function. Then we present a rounding algorithm for functions with non-positive third-order derivatives when the problem is subject to a cardinality constraint. Then we present two rounding algorithms for functions with zero third-order derivative (quadratic multi-linear functions). Finally we show that our rounding gap for these functions is almost tight.

In Chapter 4, by restricting the one-sided smoothness condition to only integral points and specific directions, we introduce a new class of set functions called γ-meta-submodular (for $\gamma = 0$, it is equivalent to meta-submodular functions defined by Kleinberg et al). We show that this class of functions contains the set functions with a one-sided smooth multi-linear extension. We give various examples for γ-meta-submodular functions. Then we give algorithms inspired by the local search (local swap) algorithm for finding an approximate solution for these functions subject to a matroid constraint. Like before we improve our approximation factor for functions with non-positive third-order difference.

In Chapter 5, we consider the maximization of γ-meta-submodular functions in distributed and streaming settings. We give an approximation algorithm for a specific 1-meta-submodular function subject to a cardinality constraint in these settings. We see that even this specific example of γ-meta-submodular functions has interesting applications in machine learning. More specifically, we show that the multi-label feature selection problem can be modeled as such an optimization problem. This modeling combined with our distributed algorithm results in the first distributed method for the multi-label feature selection problem. We then empirically compare our method with centralized multi-label feature selection methods and see that its performance is comparable or in some cases is even better than current centralized multi-label feature selection methods.

At the end of each chapter, we mention some potential directions for future research.

Chapter 2

One-Sided Smoothness

In 1978 Fisher et al. [31, 63, 64] gave a 1/2-approximation for $\max\{f(S) : S \in \mathcal{M}\}$ where $\mathcal{M}$ is a matroid and f is non-negative monotone submodular. In the special case of uniform matroids, $\mathcal{M} = \{S : |S| \leq k\}$, they gave a, provably tight, $(1 - 1/e)$-approximation. Whether this ratio could be achieved for general matroids remained open for 35 years. Partly motivated by interest in the submodular welfare problem, Calinescu, Chekuri, Pál and Vondrak [18, 81] gave such a $(1 - 1/e)$-approximation algorithm. This was based on a new (non-convex) relaxation followed by an elegant application of lossless pipage rounding of the fractional solution to a vertex of the matroid polytope. We examine both phases of their framework for clues to the question (Q1) on supermodular maximization.

At the heart of their approach is the problem of maximizing the multi-linear extension of a submodular set function over a downwards-closed polytope. Submodularity in this context ensures some nice properties for the multi-linear extension. For instance, concavity along the forward directions is used to bound a Taylor series expansion in the continuous greedy analysis [81]. Since non-submodular multi-linear extensions will not have this concavity property, we propose a "smoothness" condition which guarantees an alternative bound based on Taylor series. A continuously twice differentiable function $F : [0, 1]^n \to \mathbb{R}$ is called *one-sided σ-smooth* at $x \neq \vec{0}$ if for any $u \in [0, 1]^n$

$$u^T \nabla^2 F(x) u \leq \sigma \cdot \frac{||u||_1}{||x||_1} u^T \nabla F(x).$$

We call such a function F *one-sided σ-smooth* if it is σ-smooth at *any non-zero point* of its domain. One can see that this property captures the concavity in the forward direction for $\sigma = 0$. As we see, approximation algorithms exist for maximizing these nonlinear functions due to a bound on their second derivatives in terms of their gradient. This is the essential ingredient in several of the main results. The following result describes a property of one-sided smoothness that plays a key role in the analysis of both our continuous and discrete (local search) algorithms.

Lemma 4. *Let $x \in [0, 1]^n \setminus \{\vec{0}\}$, $u \in [0, 1]^n$ and $\epsilon > 0$ such that $x + \epsilon u \in [0, 1]^n$. Let $F : [0, 1]^n \to \mathbb{R}$ be a non-negative, monotone function which is one-sided σ-smooth on $\{y | x + \epsilon u \geq y \geq x\}$. Then*

$$u^T \nabla F(x + \epsilon u) \leq \left(\frac{||x + \epsilon u||_1}{||x||_1}\right)^\sigma (u^T \nabla F(x)).$$

Proof. Let $g(t) := u^T \nabla F(x + tu)$. By the Chain Rule we have $g'(t) = u^T \nabla^2 F(x + tu) u$.

By one-sided σ-smoothness on $\{y|x+\epsilon u \geq y \geq x\}$, for any $0 \leq t \leq \epsilon$,

$$g'(t) = u^T \nabla^2 F(x+tu) u \leq \sigma \frac{||u||_1}{||x+tu||_1} u^T \nabla F(x+tu) = \sigma \frac{||u||_1}{||x+tu||_1} g(t) \leq \sigma \frac{||u||_1}{||x+tu||_1} (g(t)+c),$$

for any $c > 0$. Therefore, using that $g(t) + c > 0$ for all t (since $g(t) \geq 0$), we have

$$\frac{g'(t)}{g(t)+c} \leq \sigma \frac{||u||_1}{||x+tu||_1}. \tag{2.1}$$

We integrate both sides of (2.1) with respect to t. On the left hand side we get

$$\int_0^\epsilon \frac{g'(t)}{g(t)+c} dt = \ln(g(t)+c)\Big|_0^\epsilon = \ln(\frac{g(\epsilon)+c}{g(0)+c}),$$

and on the right hand side we get

$$\sigma \int_0^\epsilon \frac{||u||_1}{||x+tu||_1} dt = \sigma \ln(||x+tu||_1)\Big|_0^\epsilon = \sigma \ln(\frac{||x+\epsilon u||_1}{||x||_1}),$$

where we use that $||u||_1 = \sum_i u_i = \frac{d}{dt} \sum_i (x_i + tu_i) = \frac{d}{dt} ||x+tu||_1$.

Therefore $\ln(\frac{g(\epsilon)+c}{g(0)+c}) \leq \sigma \ln(\frac{||x+\epsilon u||_1}{||x||_1})$, and hence $g(\epsilon) + c \leq \left(\frac{||x+\epsilon u||_1}{||x||_1}\right)^\sigma (g(0)+c)$. Since this holds for any $c > 0$ taking the limit yields the desired result. □

Before going further, we give some examples of one-sided σ-smooth functions.

Proposition 1. *Let $f : 2^{[n]} \to \mathbb{R}$ and F be its multi-linear extension. Then f is submodular if and only if F is one-sided 0-smooth.*

Proof. A set function f is submodular if and only if $A_{ij}(S) \leq 0$ for all $S \subseteq [n]$ and $i, j \in [n]$ [71].

Let f be submodular. Then by Corollary 1, $\nabla^2_{ij} F(x) = \mathbb{E}_{R \sim x}[A_{ij}(R)] \leq 0$, for any $x \in [0,1]^n$. It follows that $u^T \nabla^2 F(x) u \leq 0$ for any $u \in [0,1]^n$, and thus F is one-sided 0-smooth.

For the opposite direction, let F be one-sided 0-smooth and let $u = \boldsymbol{e}_i + \boldsymbol{e}_j$. Then $u^T \nabla^2 F(x) u = 2\nabla^2_{ij} F(x) \leq 0$ for all $x \neq \mathbf{0}$. Moreover, by continuity of $\nabla^2 F(x)$, the inequality also holds at $x = \mathbf{0}$. We then have that $A_{ij}(S) = \nabla^2_{ij} F(\mathbb{1}_S) \leq 0$ for all $S \subseteq [n]$, and thus f is submodular. □

Another example is the multi-linear extension of discrete quadratic functions when the corresponding distance function is semi-metric. A distance function A is *σ-semi-metric* ($\sigma \geq 0$) if $A(i,j) \leq \sigma(A(i,k) + A(j,k))$ for any i, j, and k in $[n]$.

Proposition 2. *Let $A \in \mathbb{R}^{n \times n}$ be a symmetric, 0-diagonal matrix. Let $b \in \mathbb{R}^n$ and $b \geq 0$. Then $F(x) = \frac{1}{2} x^T A x + b^T x$ is one-sided 2σ-smooth if A is σ-semi-metric.*

Proof. Note that $\nabla^2 F(x) = A$ and $\nabla F(x) = Ax + b$. Therefore for any i, j we have

$$\begin{aligned}\sigma(\nabla_i F(x) + \nabla_j F(x)) &\geq \sigma(\sum_{k=1}^{n} A(i,k)x_k + \sum_{k=1}^{n} A(j,k)x_k) = \sum_{k=1}^{n} \sigma(A(i,k) + A(j,k))x_k \\ &\geq \sum_{k=1}^{n} A(i,j)x_k = ||x||_1 A(i,j) = ||x||_1 \nabla_{ij}^2 F(x),\end{aligned}$$

where the first inequality follows from $b \geq 0$ and the last inequality holds because A is σ-semi-metric. Now we have,

$$\begin{aligned}u^T \nabla^2 F(x) u &= \sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j \nabla_{ij}^2 F(x) \leq \frac{\sigma}{||x||_1} \sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j (\nabla_i F(x) + \nabla_j F(x)) \\ &= \frac{\sigma}{||x||_1} (\sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j \nabla_i F(x) + \sum_{i=1}^{n}\sum_{j=1}^{n} u_i u_j \nabla_j F(x)) \\ &= \frac{\sigma}{||x||_1} (\sum_{i=1}^{n} u_i \nabla_i F(x)(\sum_{j=1}^{n} u_j) + \sum_{i=1}^{n} u_i(\sum_{j=1}^{n} u_j \nabla_j F(x))) \\ &= \frac{\sigma}{||x||_1} (||u||_1 \sum_{i=1}^{n} u_i \nabla_i F(x) + ||u||_1 \sum_{j=1}^{n} u_j \nabla_j F(x)) \\ &= 2\sigma \left(\frac{||u||_1}{||x||_1}\right) (u^T \nabla F(x)).\end{aligned}$$

Hence F is one-sided 2σ-smooth. □

In the next section, we proceed to investigate the maximization of one-sided σ-smooth functions.

2.1 Maximizing One-Sided Smooth Functions and Jump-Start Continuous Greedy

We give an adaptation of the continuous greedy process which yields approximation factors that are upper-bounded by a function of the smoothness parameter σ. These results are used in a 2-phase (relax and round) algorithm for maximizing a discrete quadratic function. Interestingly, however, one-sided smoothness also plays a role in the analysis of a local search algorithm discussed in the next section.

Algorithm 2.1 is for maximizing a monotone one-sided σ-smooth function over a polytime separable downwards-closed polytope. Unlike the classical continuous greedy, our algorithm starts from a non-zero point, which allows us to take advantage of Lemma 4. Because of this, we call our algorithm *jump-start continuous greedy.* The algorithm actually can start from zero for downwards-closed polytopes in which the size of all of the maximal points are the same, e.g., matroids. The reason is that we know that the algorithm always chooses a maximal point

Algorithm 2.1: Jump-start continuous greedy

Input: A monotone one-sided σ-smooth function $F : [0,1]^n \to \mathbb{R}_{\geq 0}$, a polytime separable downwards-closed polytope P, and $c \in (0,1)$

$v^* \leftarrow \arg\max_{x \in P} ||x||_1$

$x(0) \leftarrow cv^*$

$v_{max}(x) \leftarrow \arg\max_{v \in P}\{v^T \nabla F(x)\}$

for $t \in [0,1]$ **do**

  Solve $x'(t) = (1-c)v_{max}(x(t))$ with boundary condition $x(0) = cv^*$

return $x(1)$;

and after some time our vector will be sizable compare to the maximum point in the polytope. So using a similar argument to ours, we can get the same bound for such polytopes even if we start from zero. However, this is not the case for all the polytopes and the size of their maximal points might be different. For these polytopes, we can't guarantee that starting from zero, the size of the vector gets large enough to be able to use Lemma 4. Therefore in general we need to start from a non-zero point that is within a constant factor of the maximum point in the polytope.

Theorem 1. *Let $F : [0,1]^n \to \mathbb{R}_{\geq 0}$ be a monotone one-sided σ-smooth function. Let $c \in (0,1)$ and P be a polytime separable, downwards-closed, polytope. If we run the jump-start continuous greedy process (Algorithm 2.1) then $x(1) \in P$ and $F(x(1)) \geq [1 - \exp(-(1-c)(\frac{c}{c+1})^\sigma)] \cdot OPT$ where $OPT := \max\{F(x) : x \in P\}$.*

Proof. For each $t \in [0,1]$ we have

$$x(t) = x(0) + (1-c)\int_0^t v_{max}(x(\tau))\, d\tau = cv^* + (1-c)\int_0^t v_{max}(x(\tau))\, d\tau. \tag{2.2}$$

Since P is convex and $v^* \in P$, we have that $x(t) \in P$ as long as $y(t) := \int_0^t v_{max}(x(\tau))\, d\tau \in P$. Given that each $v_{max}(x(\tau)) \in P$ and also $\vec{0} \in P$, it follows that $y(t)$ is a convex combination of points in P, and hence belongs to P.

Let $x^* \in P$ be such that $F(x^*) = OPT$. Also let $x \in \{x(t) : 0 \leq t \leq 1\}$ and $u = (x^* - x) \vee 0$, i.e., $x^* \vee x = x + u$. We have by Taylor's Theorem that for some $\epsilon \in (0,1)$:

$$\begin{aligned} F(x^* \vee x) = F(x) + u^T \nabla F(x + \epsilon u) &\leq F(x) + \left(\frac{||x + \epsilon u||_1}{||x||_1}\right)^\sigma u^T \nabla F(x) \\ &\leq F(x) + \left(\frac{||x + u||_1}{||x||_1}\right)^\sigma u^T \nabla F(x) \end{aligned}$$

where the first inequality follows from Lemma 4. Hence

$$u^T \nabla F(x) \geq \frac{1}{\left(\frac{||x+u||_1}{||x||_1}\right)^\sigma}(F(x \vee x^*) - F(x)) \geq \frac{1}{\left(\frac{||x+u||_1}{||x||_1}\right)^\sigma}(OPT - F(x)), \tag{2.3}$$

where the last inequality follows from monotonicity since then $F(x \vee x^*) \geq F(x^*) = OPT$. We also have that

$$v_{max}(x) \cdot \nabla F(x) \geq x^* \cdot \nabla F(x) \geq u \cdot \nabla F(x),$$

where the first inequality follows by definition of v_{max} and the fact that $x^* \in P$, and the second inequality from the fact that $x^* \geq u$ and $\nabla F \geq 0$. Combining this with (2.3) yields:

$$v_{max}(x) \cdot \nabla F(x) \geq \frac{1}{\left(\frac{||x+u||_1}{||x||_1}\right)^\sigma}(OPT - F(x)). \tag{2.4}$$

By the choice of $x(0)$ we have that $||x(0)||_1 \geq c||w||_1$ for any $w \in P$. Since $u \in P$ and $x(t)$ is non-decreasing in each component (because v_{max} is always non-negative), we thus have

$$\frac{||x+u||_1}{||x||_1} \leq 1 + \frac{||u||_1}{||x||_1} \leq 1 + \frac{||u||_1}{||x(0)||_1} \leq 1 + \frac{1}{c} = \frac{c+1}{c}.$$

Hence we deduce that

$$\frac{1}{\left(\frac{||x+u||_1}{||x||_1}\right)^\sigma} \geq (\frac{c}{c+1})^\sigma \tag{2.5}$$

for any $x \in \{x(t) : 0 \leq t \leq 1\}$. Let us define ρ to be the righthand side quantity above. Intuitively, (2.4) indicates that the direction v_{max} makes at least a ρ "fractional progress" towards OPT.

Moreover, we can use the Chain Rule to get

$$\frac{d}{dt}F(x(t)) = \nabla F(x(t)) \cdot x'(t) = \nabla F(x(t)) \cdot (1-c)v_{max}(x(t)) \geq \rho(1-c)[OPT - F(x(t))], \tag{2.6}$$

where the last inequality follows from (2.4) and (2.5).

We solve the above differential inequality by multiplying by $e^{\rho(1-c)t}$.

$$\begin{aligned}
\frac{d}{dt}[e^{\rho(1-c)t} \cdot F(x(t))] &= \rho(1-c)e^{\rho(1-c)t} \cdot F(x(t)) + e^{\rho(1-c)t} \cdot \frac{d}{dt}F(x(t)) \\
&\geq \rho(1-c)e^{\rho(1-c)t} \cdot F(x(t)) + \rho \cdot e^{\rho(1-c)t}(1-c)[OPT - F(x(t))] \\
&= \rho(1-c)e^{\rho(1-c)t} \cdot OPT.
\end{aligned}$$

where the inequality follows from Equation (2.6).

Integrating the LHS and RHS of the above equation between 0 and t we get

$$\begin{aligned}
e^{\rho(1-c)t} \cdot F(x(t)) - e^0 \cdot F(x(0)) &\geq \rho(1-c)OPT \int_0^t e^{\rho(1-c)\tau} d\tau \\
&= \rho(1-c)OPT \cdot [\frac{e^{\rho(1-c)t}}{\rho(1-c)} - \frac{1}{\rho(1-c)}] = OPT \cdot [e^{\rho(1-c)t} - 1].
\end{aligned}$$

Hence

$$F(x(t)) \quad \geq \quad [1 - \frac{1}{e^{\rho(1-c)t}}]OPT + \frac{F(x(0))}{e^{\rho(1-c)t}} \geq [1 - \frac{1}{e^{\rho(1-c)t}}]OPT,$$

where the last inequality follows from the fact that F is non-negative. Substituting $t = 1$ and $\rho = (\frac{c}{c+1})^\sigma$ gives the desired result. □

Our bound depends on the constant number c and also σ. Since c is part of the input, one question is what is the best value for c. This is answered in the following result.

Proposition 3. *For any $\sigma > 0$ the best approximation guarantee in Theorem 1 is attained at*

$$c = \frac{\sqrt{\sigma^2 + 6\sigma + 1} - (\sigma + 1)}{2}.$$

Proof. We need to find the maximizer of $g(c) = (1-c)(\frac{c}{c+1})^\sigma$ where $c \in [0,1]$. Hence, we solve $g'(c) = 0$.

$$\begin{aligned}
&g'(c) = \frac{\sigma c^{\sigma-1}(c+1)^\sigma - (\sigma+1)c^\sigma(c+1)^\sigma - \sigma(c+1)^{\sigma-1}c^\sigma + \sigma(c+1)^{\sigma-1}c^{\sigma+1}}{(c+1)^{2\sigma}} = 0 \\
&\Rightarrow \sigma c^{\sigma-1}(c+1)^{\sigma-1} - \sigma c^\sigma(c+1)^{\sigma-1} = c^\sigma(c+1)^\sigma \\
&\Rightarrow \sigma c^{\sigma-1}(c+1)^{\sigma-1}(1-c) = c^\sigma(c+1)^\sigma \\
&\Rightarrow \sigma(1-c) = c(c+1) \Rightarrow c^2 + (1+\sigma)c - \sigma = 0 \Rightarrow c = \frac{-(\sigma+1) \pm \sqrt{\sigma^2 + 6\sigma + 1}}{2}
\end{aligned}$$

The only solution in $(0,1)$ is $\frac{-(\sigma+1)+\sqrt{\sigma^2+6\sigma+1}}{2}$ and this yields the proposition. ☐

2.2 Jump-Start Continuous Greedy for Second-Order Smooth Functions

If we assume that σ is a constant, then the jump-start continuous greedy finds a constant factor approximation. However, the approximation factor is exponential in σ, so one immediate question is that if it is possible to improve it to a polynomial in terms of σ. We conjecture that this is possible for general one-sided σ-smooth functions. More specifically, we conjecture that it should be cubic in terms of σ. As a more immediate answer, we improve it to linear for a special class of one-sided σ-smooth functions that admit non-positive third-order derivative. This can be considered as having one-sided 0-smoothness for the derivatives. In terms of set functions, it is related to the class of second-order submodular functions considered in [52]. These are the functions that their marginal gains (derivatives) admit submodularity.

Theorem 2. *Let $F : [0,1]^n \to \mathbb{R}_{\geq 0}$ be a monotone one-sided σ-smooth function with non-positive third order partial derivatives. Let $c \in (0,1)$ and P be a polytime separable, downwards-closed, polytope. If we run the jump-start continuous greedy process (Algorithm 2.1) then $x(1) \in P$ and $F(x(1)) \geq [1 - \exp(-\frac{2c(1-c)}{2c+\sigma})] \cdot OPT$ where $OPT := \max\{F(x) : x \in P\}$. In particular, taking $c = 1/2$ we get $F(x(1)) \geq [1 - \exp(-\frac{1}{2\sigma+2})] \cdot OPT$ and so $F(x(1)) \geq \frac{1}{2\sigma+3} \cdot OPT$ (since $e^x \geq x+1$ for $x < 1$).*

Proof. For each $t \in [0,1]$ we have

$$x(t) = x(0) + (1-c)\int_0^t v_{max}(x(\tau))\,d\tau = cv^* + (1-c)\int_0^t v_{max}(x(\tau))\,d\tau. \tag{2.7}$$

Since P is convex and $v^* \in P$, we have that $x(t) \in P$ as long as $y(t) := \int_0^t v_{max}(x(\tau))\, d\tau \in P$. Given that each $v_{max}(x(\tau)) \in P$ and also $\vec{0} \in P$, it follows that $y(t)$ is a convex combination of points in P, and hence belongs to P.

Let $x^* \in P$ be such that $F(x^*) = OPT$. Also let $x \in \{x(t) : 0 \leq t \leq 1\}$ and $u = (x^* - x) \vee 0$, i.e., $x^* \vee x = x + u$. By Taylor's Theorem and non-positivity of the third order derivatives of F we have

$$\begin{aligned} F(x^* \vee x) &\leq F(x) + u^T \nabla F(x) + \frac{1}{2} u^T \nabla^2 F(x) u \leq F(x) + \Big(1 + \frac{\sigma ||u||}{2||x||}\Big) u^T \nabla F(x) \\ &\leq F(x) + \Big(1 + \frac{\sigma}{2c}\Big) u^T \nabla F(x), \end{aligned}$$

where the second inequality follows from smoothness, and the third from the fact that $||x(t)|| \geq ||x(0)|| = c||v^*|| \geq c||u||$. Thus

$$u^T \nabla F(x) \geq \Big(\frac{2c}{2c+\sigma}\Big)\Big(F(x \vee x^*) - F(x)\Big) \geq \Big(\frac{2c}{2c+\sigma}\Big)\Big(OPT - F(x)\Big), \tag{2.8}$$

where the last inequality follows from monotonicity. We also have that

$$v_{max}(x) \cdot \nabla F(x) \geq x^* \cdot \nabla F(x) \geq u \cdot \nabla F(x),$$

where the first inequality follows by definition of v_{max} and the fact that $x^* \in P$, and the second inequality from the fact that $x^* \geq u$ and $\nabla F \geq 0$. Combining this with (2.8) yields:

$$v_{max}(x) \cdot \nabla F(x) \geq \Big(\frac{2c}{2c+\sigma}\Big)\Big(OPT - F(x)\Big), \tag{2.9}$$

for any $x \in \{x(t) : 0 \leq t \leq 1\}$. Let us denote $\rho = 2c/(2c+\sigma)$. We can use the Chain Rule to get

$$\frac{d}{dt} F(x(t)) = \nabla F(x(t)) \cdot x'(t) = \nabla F(x(t)) \cdot (1-c) v_{max}(x(t)) \geq \rho(1-c)\Big[OPT - F(x(t))\Big], \tag{2.10}$$

where the last inequality follows from (2.9).

We solve the above differential inequality by multiplying by $e^{\rho(1-c)t}$.

$$\begin{aligned} \frac{d}{dt}[e^{\rho(1-c)t} \cdot F(x(t))] &= \rho(1-c)e^{\rho(1-c)t} \cdot F(x(t)) + e^{\rho(1-c)t} \cdot \frac{d}{dt} F(x(t)) \\ &\geq \rho(1-c)e^{\rho(1-c)t} \cdot F(x(t)) + \rho \cdot e^{\rho(1-c)t}(1-c)[OPT - F(x(t))] \\ &= \rho(1-c)e^{\rho(1-c)t} \cdot OPT. \end{aligned}$$

where the inequality follows from Equation (2.10).

Integrating the LHS and RHS of the above equation between 0 and t we get

$$\begin{aligned} e^{\rho(1-c)t} \cdot F(x(t)) - e^0 \cdot F(x(0)) &\geq \rho(1-c) OPT \int_0^t e^{\rho(1-c)\tau} d\tau \\ &= \rho(1-c) OPT \cdot \Big[\frac{e^{\rho(1-c)t}}{\rho(1-c)} - \frac{1}{\rho(1-c)}\Big] = OPT \cdot [e^{\rho(1-c)t} - 1]. \end{aligned}$$

Hence

$$F(x(t)) \quad \geq \quad [1 - \frac{1}{e^{\rho(1-c)t}}]OPT + \frac{F(x(0))}{e^{\rho(1-c)t}} \geq [1 - \frac{1}{e^{\rho(1-c)t}}]OPT,$$

where the last inequality follows from the fact that F is non-negative. Substituting $t = 1$ and $\rho = 2c/(2c + \sigma)$ gives the desired result. ☐

In the next section, we investigate the smoothness of the multi-linear extension of general monotone functions.

2.3 Sub-domain Smoothness of General Monotone Set Functions

In general, we do not need the smoothness on the whole domain of the function in order to be able to find an approximation. Hence, we can look at functions that only admit the one-sided smoothness on a specific subset of their domain. This sub-domain smoothness, for example, appears in the multi-linear extension of monotone set functions. In this section, we discuss the sub-domain smoothness of these functions and provide an adaptation of the jump-start continuous greedy algorithm that can be used for maximizing the multi-linear extension of a general monotone set function (Algorithm 2.2).

Proposition 4. *Let $f : 2^{[n]} \to \mathbb{R}$ be a non-negative, monotone function and F be its multi-linear extension. Let $x \in [0,1]^n$ such that $x_v > 0$ for each $v \in [n]$. Then there is a $\sigma \geq 0$, such that F is one-sided σ-smooth at x. Moreover, let $z \in [0,1]^n$ whose smallest component value is $z_{min} > 0$. Then F is $\frac{n}{z_{min}}$-smooth on $\{x : 1 \geq x \geq z\}$.*

Proof. Let $i, j \in [n]$. By Lemma 1 we have

$$\begin{aligned}\nabla^2_{ij} F(x) &= \sum_{R \subseteq [n]} A_{ij}(R) p_x(R) = \sum_{R \subseteq [n]} (B_i(R+j) - B_i(R-j)) p_x(R) \\ &= \sum_{R \subseteq [n]} B_i(R+j) p_x(R) - \sum_{R \subseteq [n]} B_i(R-j) p_x(R).\end{aligned}$$

We first show that there is $\gamma_{ij} > 0$ such that

$$||x||_1 \nabla^2_{ij} F(x) \leq \gamma_{ij}(\nabla_i F(x) + \nabla_j F(x)). \tag{2.11}$$

Since f is monotone, the right hand side is non-negative. Hence, if $\nabla^2_{ij} F(x)$ is non-positive, the inequality holds for any $\gamma_{ij} > 0$. Therefore, we assume that $\nabla^2_{ij} F(x)$ is positive which implies that $\sum_{R\subseteq[n]} B_i(R+j)p_x(R) > 0$ by monotonicity. Hence

$$
\begin{aligned}
0 &< \nabla^2_{ij}F(x) \le \sum_{R\subseteq[n]} B_i(R+j)p_x(R) \\
&= \sum_{R\subseteq[n]-j} B_i(R+j)p_x(R) + \sum_{R\subseteq[n]-j} B_i((R+j)+j)p_x(R+j) \\
&= \sum_{R\subseteq[n]-j} B_i(R+j)(p_x(R)+p_x(R+j)) = \sum_{R\subseteq[n]-j} B_i(R+j)(\frac{1-x_j}{x_j}p_x(R+j)+p_x(R+j)) \\
&= \sum_{R\subseteq[n]-j} B_i(R+j)(\frac{1}{x_j}p_x(R+j)) = \frac{1}{x_j}\sum_{R\subseteq[n]-j} B_i(R+j)p_x(R+j) \\
&\le \frac{1}{x_j}(\sum_{R\subseteq[n]-j} B_i(R)p_x(R) + \sum_{R\subseteq[n]-j} B_i(R+j)p_x(R+j)) = \frac{1}{x_j}\sum_{R\subseteq[n]} B_i(R)p_x(R) \\
&= \frac{1}{x_j}\nabla_i F(x).
\end{aligned}
$$

Hence, we conclude that $\nabla_i F(x) \ge \nabla^2_{ij}F(x)$ and so if $\nabla^2_{ij}F(x)$ is positive, then $\nabla_i F(x)+\nabla_j F(x)$ is also positive. Now, set $\gamma_{ij}=0$ if $\nabla^2_{ij}F(x)$ is non-positive and otherwise we set

$$
\gamma_{ij} = \frac{||x||_1 \nabla^2_{ij}F(x)}{\nabla_i F(x)+\nabla_j F(x)} \le ||x||_1 \frac{\nabla^2_{ij}F(x)}{(x_i+x_j)\nabla^2_{ij}F(x)} = \frac{||x||_1}{x_i+x_j}. \tag{2.12}
$$

Let $\gamma = 2\max_{\{i,j\}\subseteq[n]}\gamma_{ij}$. Then for $u\in[0,1]^n$, we have by (2.12)

$$
\begin{aligned}
u^T\nabla^2 F(x)u &= \sum_{i=1}^n\sum_{j=1}^n u_iu_j\nabla^2_{ij}F(x) \le \frac{1}{||x||_1}\sum_{i=1}^n\sum_{j=1}^n \gamma_{ij}u_iu_j(\nabla_i F(x)+\nabla_j F(x)) \\
&\le \frac{\gamma}{2}\frac{1}{||x||_1}\sum_{i=1}^n\sum_{j=1}^n u_iu_j(\nabla_i F(x)+\nabla_j F(x)) \\
&= \frac{\gamma}{2}\frac{1}{||x||_1}(\sum_{i=1}^n\sum_{j=1}^n u_iu_j\nabla_i F(x) + \sum_{i=1}^n\sum_{j=1}^n u_iu_j\nabla_j F(x)) \\
&= \frac{\gamma}{2}\frac{1}{||x||_1}(\sum_{i=1}^n u_i\nabla_i F(x)(\sum_{j=1}^n u_j) + \sum_{i=1}^n u_i(\sum_{j=1}^n u_j\nabla_j F(x))) \\
&= \frac{\gamma}{2}\frac{1}{||x||_1}(||u||_1\sum_{i=1}^n u_i\nabla_i F(x) + ||u||_1\sum_{j=1}^n u_j\nabla_j F(x)) \\
&= \gamma\left(\frac{||u||_1}{||x||_1}\right)(u^T\nabla F(x)).
\end{aligned}
$$

Now for the second part of the proof we must choose a γ that works for all $x\ge z$ and each i,j. By (2.12) it is sufficient to choose $\gamma = \max_{i,j}\{\frac{||x||_1}{x_i+x_j} : x\in[0,1]^n, x\ge z\} \le \frac{n}{z_{min}}$. □

Algorithm 2.2: Jump-start continuous greedy for monotone functions

Input: A monotone set function f, its multi-linear extension F, a polytime separable, downwards-closed polytope $P \subseteq [0,1]^n$ and $c \in (0,1)$.
$v^* \leftarrow \arg\max_{x \in P} ||x||_1$
$x(0) \leftarrow c(\frac{1}{||v^*||_1+1}\frac{\mathbb{1}_{[n]}}{n} + \frac{||v^*||_1}{||v^*||_1+1}v^*)$
$v_{max}(x) \leftarrow \arg\max_{v \in P}\{v^T \nabla F(x)\}$
for $t \in [0,1]$ **do**
  Solve $x'(t) = (1-c)v_{max}(x(t))$ with boundary condition $x(0) = c(\frac{1}{||v^*||_1+1}\frac{\mathbb{1}_{[n]}}{n} + \frac{||v^*||_1}{||v^*||_1+1}v^*)$
return $x(1)$;

Using this sub-domain smoothness property, we show that Algorithm 2.2 finds an approximation for the multi-linear extension of a general monotone set function. The bound is not good but it is interesting that such an algorithm works for such a general class of functions.

Theorem 3. *Let $f : 2^{[n]} \to \mathbb{R}$ be a non-negative, monotone set function and F be its multi-linear extension. Let $c \in (0,1)$ and P be a polytime separable, downwards-closed, convex polytope such that $\mathbb{1}_{\{i\}} \in P$ for any $i \in [n]$. Let σ be the one-sided smoothness parameter on $\{y | y \geq c(\frac{1}{||v^*||_1+1}\frac{\mathbb{1}_{[n]}}{n} + \frac{||v^*||_1}{||v^*||_1+1}v^*)\}$ where, $v^* = \arg\max_{x \in P} ||x||_1$. Then Algorithm 2.2 outputs $x(1) \in P$ such that*

$$F(x(1)) \geq [1 - \exp(-(1-c)(\frac{c}{c+2})^{\sigma})] \cdot OPT$$

where $OPT := \max\{F(x) : x \in P\}$.

Proof. We know that $\mathbb{1}_{\{i\}} \in P$ for any $i \in [n]$ and so a convex combination of these is also in the polytope which means $\frac{\mathbb{1}_{[n]}}{n} \in P$. Hence, since $v^* \in P$ and P is convex,

$$(\frac{1}{||v^*||_1+1}\frac{\mathbb{1}_{[n]}}{n} + \frac{||v^*||_1}{||v^*||_1+1}v^*) \in P.$$

For each $t \in [0,1]$ we have

$$x(t) = x(0) + (1-c)\int_0^t v_{max}(x(\tau))\,d\tau \tag{2.13}$$

Since P is convex and $(\frac{1}{||v^*||_1+1}\frac{\mathbb{1}_{[n]}}{n} + \frac{||v^*||_1}{||v^*||_1+1}v^*) \in P$, we have that $x(t) \in P$ as long as $y(t) := \int_0^t v_{max}(x(\tau))\,d\tau \in P$. Given that each $v_{max}(x(\tau)) \in P$ and also $\vec{0} \in P$, it follows that $y(t)$ is a convex combination of points in P, and hence belongs to P.

Let $x^* \in P$ be such that $F(x^*) = OPT$. Let $y \geq x(0)$ and $u = (x^* - y) \vee 0$, i.e., $x^* \vee y = y + u$. Note that all the coordinate of $x(0)$ are non-zero. We have by Taylor's Theorem that for some $\epsilon \in (0,1)$:

$$F(x^* \vee y) = F(y) + u^T \nabla F(y + \epsilon u) \leq F(y) + \left(\frac{||y+\epsilon u||_1}{||y||_1}\right)^{\sigma} u^T \nabla F(y)$$
$$\leq F(y) + \left(\frac{||y+u||_1}{||y||_1}\right)^{\sigma} u^T \nabla F(y)$$

where the first inequality follows from Proposition 4 and Lemma 4. Hence

$$u^T \nabla F(y) \geq \frac{1}{\left(\frac{||y+u||_1}{||y||_1}\right)^{\sigma}}(F(y \vee x^*) - F(y)) \geq \frac{1}{\left(\frac{||y+u||_1}{||y||_1}\right)^{\sigma}}(OPT - F(y)), \tag{2.14}$$

where the last inequality follows from monotonicity since then $F(y \vee x^*) \geq F(x^*) = OPT$. The definition of v_{max} implies that $v_{max}(y) \cdot \nabla F(y) \geq x^* \cdot \nabla F(y)$. Since f is monotonic, $\nabla F \geq 0$. Hence since $u = (x^* - y) \vee 0 \leq x^*$, we also have $x^* \cdot \nabla F(y) \geq u \cdot \nabla F(y)$. Combining these with (2.14) yields:

$$v_{max}(y) \cdot \nabla F(y) \geq \frac{1}{\left(\frac{||y+u||_1}{||y||_1}\right)^{\sigma}}(OPT - F(y)). \tag{2.15}$$

By the choice of $x(0)$ we have that for any $w \in P$,

$$\begin{aligned} ||x(0)||_1 &= ||c(\frac{1}{||v^*||_1 + 1}\frac{\mathbb{1}_{[n]}}{n} + \frac{||v^*||_1}{||v^*||_1 + 1}v^*)||_1 = c\frac{||v^*||_1^2 + 1}{||v^*||_1 + 1} = \frac{c}{2}\frac{2(||v^*||_1^2 + 1)}{||v^*||_1 + 1} \\ &\geq \frac{c}{2}||v^*||_1 \geq \frac{c}{2}||w||_1 \end{aligned}$$

Since $u \in P$ and $x(t)$ is non-decreasing in each component (because v_{max} is always non-negative), we thus have

$$\frac{||x(t) + u||_1}{||x(t)||_1} \leq 1 + \frac{||u||_1}{||x(t)||_1} \leq 1 + \frac{||u||_1}{||x(0)||_1} \leq 1 + \frac{2}{c} = \frac{c+2}{c}.$$

Hence we deduce that

$$\frac{1}{\left(\frac{||x(t)+u||_1}{||x(t)||_1}\right)^{\sigma}} \geq (\frac{c}{c+2})^{\sigma}$$

for all $x(t)$. Let us define ρ to be the righthand side quantity above. Intuitively, (2.15) indicates that the direction v_{max} makes at least a ρ "fractional progress" towards OPT.
Moreover, we can use the Chain Rule to get

$$\frac{d}{dt}F(x(t)) = \nabla F(x(t)) \cdot x'(t) = \nabla F(x(t)) \cdot (1-c)v_{max}(x(t)) \geq \rho(1-c)[OPT - F(x(t))], \tag{2.16}$$

where the last inequality follows from Equation (2.15).
We solve the above differential inequality by multiplying by $e^{\rho(1-c)t}$.

$$\begin{aligned} \frac{d}{dt}[e^{\rho(1-c)t} \cdot F(x(t))] &= \rho(1-c)e^{\rho(1-c)t} \cdot F(x(t)) + e^{\rho(1-c)t} \cdot \frac{d}{dt}F(x(t)) \\ &\geq \rho(1-c)e^{\rho(1-c)t} \cdot F(x(t)) + \rho \cdot e^{\rho(1-c)t}(1-c)[OPT - F(x(t))] \\ &= \rho(1-c)e^{\rho(1-c)t} \cdot OPT. \end{aligned}$$

where the inequality follows from Equation (2.16).

Integrating the LHS and RHS of the above equation between 0 and t we get

$$e^{\rho(1-c)t} \cdot F(x(t)) - e^0 \cdot F(x(0)) \geq \rho(1-c)OPT \int_0^t e^{\rho(1-c)\tau} d\tau$$
$$= \rho(1-c)OPT \cdot [\frac{e^{\rho(1-c)t}}{\rho(1-c)} - \frac{1}{\rho(1-c)}] = OPT \cdot [e^{\rho(1-c)t} - 1].$$

Hence

$$F(x(t)) \geq [1 - \frac{1}{e^{\rho(1-c)t}}]OPT + \frac{F(x(0))}{e^{\rho(1-c)t}} \geq [1 - \frac{1}{e^{\rho(1-c)t}}]OPT,$$

where the last inequality follows from the fact that F is nonnegative. Taking $t = 1$ we get

$$F(x(1)) \geq [1 - \frac{1}{e^{\rho(1-c)}}]OPT.$$

Substituting $\rho = (\frac{c}{c+2})^\sigma$ gives the desired result. $\square$

In the next section, we investigate the subdomain smoothness of *meta-submodular* functions defined by Kleinberg et al [48] and its implication for maximizing them subject to matroid constraints.

2.4 Maximizing Meta-Submodular Functions of Kleinberg et al

Kleinberg et al [48] defined meta-submodular functions as set functions that satisfy the submodular property for non-disjoint sets $S, T \subseteq [n]$.

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$$

As shown in [48], one can see that in terms of our notation, a meta-submodular function satisfies $A_{ij}(S) \leq 0$ for any non-empty $S \subseteq [n]$. This results in a sub-domain smoothness property for these functions that can be used for maximizing them subject to matroid constraints.

Proposition 5. *Let f be a non-negative, monotone, meta-submodular function of Kleinberg et al [48] and F be its multi-linear extension. Then for any $v \in [n]$, F is one-sided 0-smooth on $\{x \in [0,1]^n : x \geq \mathbb{1}_{\{v\}}\}$.*

Proof. By meta-submodularity, for any set R, we have $|R|A_{ij}(R) \leq 0$. This means that for any non-empty R, $A_{ij}(R) \leq 0$. Since $x_v = 1$, the probability of picking a set that does not include v is zero. Therefore, we have

$$\nabla^2_{ij} F(x) = \sum_{R \subseteq [n]} A_{ij}(R) p_x(R) = \sum_{R \subseteq [n]-v} A_{ij}(R+v) p_x(R+v) \leq 0.$$

Hence for $u \in [0,1]^n$,

$$u^T \nabla^2 F(x) u = 2 \sum_{\{i,j\} \subseteq [n]} u_i u_j \nabla^2_{ij} F(x) \leq 0.$$

$\square$

In the rest of this section, we provide an adaptation of the continuous greedy algorithm for maximizing these meta-submodular function over a polytime separable, downwards-closed polytope. We also show that the pipage rounding algorithm can be used to round the solution of the continuous greedy over a matroid polytope. This generalizes the result of Kleinberg et al [48] from uniform matroids to general matroid constraints. More specifically, it gives a $(1 - \frac{1}{e} - o(1))$-approximation for maximizing these meta-submodular functions subject to a matroid constraint.

Theorem 4. *There is a randomized* $(1-\frac{1}{e}-o(1))$*-approximation for maximizing a non-negative, monotone, meta-submodular function of Kleinberg et al subject to a matroid constraint.*

Given a matroid $\mathcal{M} = ([n], \mathcal{I})$, and an independent set $R \in \mathcal{I}$, we denote by $\mathcal{M}_R = ([n] - R, \mathcal{I}_R)$ the contraction of $\mathcal{M}$ by R. That is, $I \in \mathcal{I}_R$ if and only if $R \cup I \in \mathcal{I}$. We denote by $P_R \subseteq [0,1]^{[n]-R}$ its associated matroid polytope. We also define an *extended* version of P_R, as $\bar{P}_R = \{x \in [0,1]^n : x|_R = 0,\ x|_{[n]-R} \in P_R\}$, where $x|_R \in [0,1]^R$ denotes the restriction of x to the components in R. That is, $\bar{P}_R$ is obtained by extending the contracted polytope P_R to the original space $[0,1]^n$, and setting all components $x_i = 0$ for $i \in R$.

Theorem 5. *Let f be a non-negative monotone* 0*-meta submodular function and F be its multilinear extension. Let $\mathcal{M} = ([n], \mathcal{I})$ be a matroid, $P(\mathcal{M})$ its corresponding polytope, and $R \in \mathcal{I}$ an independent set. Then, the continuous greedy process described in Algorithm 2.3, outputs a vector $x \in P(\mathcal{M})$ satisfying $x \geq \mathbb{1}_R$ and*

$$F(x) \geq [1 - e^{-1}] \cdot OPT_R$$

where $OPT_R := \max\{F(x) : x \geq \mathbb{1}_R\}$.

Proof. For each $t \in [0,1]$ we have

$$x(t) = x(0) + \int_0^t v_{max}(x(\tau))\, d\tau = \mathbb{1}_R + \int_0^t v_{max}(x(\tau))\, d\tau. \tag{2.17}$$

Note that $x \in \bar{P}_R$ if and only if x is a convex combination $x = \sum_{i=1}^m \lambda_i \mathbb{1}_{S_i}$ of some independent sets $S_i \in \mathcal{I}_R$ (i.e. $R \cup S_i \in \mathcal{I}$). Thus, $\mathbb{1}_R + x = \mathbb{1}_R + \sum_{i=1}^m \lambda_i \mathbb{1}_{S_i} = \sum_{i=1}^m \lambda_i [\mathbb{1}_R + \mathbb{1}_{S_i}] \in P(\mathcal{M})$ since $\mathbb{1}_R + \mathbb{1}_{S_i} \in P(\mathcal{M})$ for each $i \in [m]$. Given that each $v_{max}(x(\tau)) \in \bar{P}_R$ for each τ, it follows that $(\int_0^t v_{max}(x(\tau))\, d\tau) \in \bar{P}_R$ and therefore $x(t) \in P(\mathcal{M})$. Moreover, it is clear that $x(t) \geq \mathbb{1}_R$.

Let $U := \{y + \mathbb{1}_R : y \in \bar{P}_R\}$, or equivalently, $U = \{x \in P(\mathcal{M}) : x|_R = \mathbb{1}_R\}$. Let $x, x^* \in U$ be such that $F(x^*) = OPT_R$ and $u = (x^* - x) \vee 0$, i.e., $x^* \vee x = x + u$. By Theorem 16, we know that F is one-sided 0-smooth at U. Hence, we have by Taylor's Theorem that for some $\epsilon \in (0,1)$:

$$F(x^* \vee x) = F(x) + u^T \nabla F(x + \epsilon u) \leq F(x) + \left(\frac{||x + \epsilon u||_1}{||x||_1} \right)^0 u^T \nabla F(x) = F(x) + u^T \nabla F(x)$$

where the inequality follows from Lemma 4. Hence

$$u^T \nabla F(x) \geq F(x \vee x^*) - F(x) \geq OPT_R - F(x). \tag{2.18}$$

Algorithm 2.3: Jump-start continuous greedy for contracted matroids

Input: A monotone set function f, its multi-linear extension F, a matroid $\mathcal{M}$, an independent set R, and its extended contracted polytope $\bar{P}_R$
$x(0) \leftarrow \mathbb{1}_R$
$v_{max}(x) \leftarrow \arg\max_{v \in \bar{P}_R} \{v^T \nabla F(x)\}$
for $t \in [0, 1]$ **do**
  Solve $x'(t) = v_{max}(x(t))$ with boundary condition $x(0) = \mathbb{1}_R$
return $x(1)$;

We also have that

$$v_{max}(x) \cdot \nabla F(x) \geq (x^* - \mathbb{1}_R) \cdot \nabla F(x) \geq u \cdot \nabla F(x),$$

where the first inequality follows by definition of v_{max} and the fact that $x^* - \mathbb{1}_R \in \bar{P}_R$, and the second inequality from the fact that $x^* - \mathbb{1}_R \geq u$ and $\nabla F \geq 0$. Combining this with (2.18) yields:

$$v_{max}(x) \cdot \nabla F(x) \geq OPT_R - F(x). \tag{2.19}$$

We can now use the Chain Rule to get

$$\frac{d}{dt} F(x(t)) = \nabla F(x(t)) \cdot x'(t) = \nabla F(x(t)) \cdot v_{max}(x(t)) \geq OPT_R - F(x(t)), \tag{2.20}$$

where the last inequality follows from Equation (2.19).

We solve the above differential inequality by multiplying by e^t.

$$\frac{d}{dt}[e^t \cdot F(x(t))] = e^t \cdot F(x(t)) + e^t \cdot \frac{d}{dt} F(x(t)) \geq e^t \cdot F(x(t)) + e^t[OPT_R - F(x(t))] = e^t \cdot OPT_R.$$

where the inequality follows from Equation (2.20).

Integrating the LHS and RHS of the above equation between 0 and t we get

$$e^t \cdot F(x(t)) - e^0 \cdot F(x(0)) \geq OPT_R \int_0^t e^\tau d\tau = OPT_R \cdot [e^t - 1].$$

Hence

$$F(x(t)) \geq [1 - \frac{1}{e^t}] OPT_R + \frac{F(x(0))}{e^t} \geq [1 - \frac{1}{e^t}] OPT_R,$$

where the last inequality follows from the fact that F is nonnegative. Taking $t = 1$ we get

$$F(x(1)) \quad \geq \quad [1 - \frac{1}{e}] OPT_R.$$

□

This now leads to the following result.

Algorithm 2.4: Refinement subroutine

Input: A vector $x \in [0,1]^n$ and two components $i, j \in \{1, 2, \ldots, n\}$
Let $\mathcal{S} = \{S \subseteq V : i \in S, j \notin S\}$
Compute $S^* = \arg\min_{S \in \mathcal{S}}[r(S) - x(S)]$ and let $\xi^* = r(S^*) - x(S^*)$
if $x_j < \xi^*$ **then**
  $x_i \leftarrow x_i + x_j, \;\; x_j \leftarrow 0, \;\; S' \leftarrow \{j\}$
else
  $x_i \leftarrow x_i + \xi^*, \;\; x_j \leftarrow x_j - \xi^*, \;\; S' \leftarrow S^*$
Output (x, S')

Corollary 2. *Let f be a non-negative monotone meta-submodular function of Kleinberg et al and F be its multi-linear extension. Let $\mathcal{M} = ([n], \mathcal{I})$ be a matroid, and $P(\mathcal{M})$ its corresponding polytope. For each $i \in [n]$, let x^i denote the output of Algorithm 2.3, run with $R = \{i\}$, and let $\bar{x} = \arg\max_{i \in [n]} F(x^i)$. Then $\bar{x} \in P(\mathcal{M})$ and*

$$F(\bar{x}) \geq [1 - e^{-1}] \cdot \max\{f(S) : S \in \mathcal{I}\}.$$

Proof. Let $O = \arg\max_{S \in \mathcal{I}} f(S)$ and $i \in O$. Then $\mathbb{1}_O \geq \mathbb{1}_{\{i\}}$, and hence

$$F(\bar{x}) \geq F(x^i) \geq (1 - \frac{1}{e}) \cdot \max\{F(x) : x \geq \mathbb{1}_{\{i\}}\} \geq (1 - \frac{1}{e}) F(\mathbb{1}_O) = (1 - \frac{1}{e}) f(O).$$

where the second inequality follows from Theorem 5. □

Hence, we can find a $(1 - 1/e)$-approximate fractional solution by running the continuous greedy process n times. By standard techniques (see [18, 81]), one may discretize the continuous greedy process to obtain a finite algorithm achieving a $(1 - 1/e - o(1))$-approximation. In fact, it may be the case that a more careful analysis provides a clean $(1 - 1/e)$-approximation.

We now discuss a randomized technique that allows to round efficiently in the matroid polytope. This rounding technique was initially introduced by Ageev and Sviridenko [3], and later adapted for matroid polytopes by Calinescu et al. [17]. This rounding procedure is known as randomized *pipage rounding* and we describe it in Algorithm 2.5 (also note that it uses Algorithm 2.4 as a subroutine).

By monotonicity we may assume that the output x^* of the continuous greedy algorithm is without loss of generality in the base polytope. We then have the following.

Theorem 6. *Let $f : 2^{[n]} \to \mathbb{R}_{\geq 0}$ be a meta-submodular set function of Kleinberg et al and $F : [0,1]^n \to \mathbb{R}_{\geq 0}$ its multilinear extension. Let $\mathcal{M}$ be a matroid and $x^* \in B(\mathcal{M})$ be the output of Corollary 2 over $\mathcal{M}$. Then Algorithm 2.5 outputs in polynomial time a random base B of $\mathcal{M}$ such that $\mathbb{E}[\mathbb{1}_B] = x^*$ and $\mathbb{E}[f(B)] \geq F(x^*)$.*

Proof. It is well known [17] that the randomized pipage rounding algorithm finishes in polynomial time. We next argue that there is no loss (on expectation) in the objective value during the rounding. Let x^* be the output of Corollary 2. Hence $x^*_{i^*} = 1$ for some $i^* \in [n]$, and by Proposition 5 it follows that F is 0-smooth over the region $\mathcal{R} := \{x \in [0,1]^n : x_{i^*} = 1\}$, that is, $\nabla^2_{ij} F(x) \leq 0$ for all $x \in \mathcal{R}$.

Algorithm 2.5: Pipage rounding

Input: A vector $x \in [0,1]^n$ and a matroid polytope $P(\mathcal{M})$

while x *not integral* **do**
- $S \leftarrow V$
- **while** S *has fractional variables* **do**
  - Choose $i, j \in S$ fractional
  - $(x^+, S^+) \leftarrow$ Refinement Subroutine (x, i, j)
  - $(x^-, S^-) \leftarrow$ Refinement Subroutine (x, j, i)
  - **if** $x = x^+ = x^-$ **then**
    - $S \leftarrow S \cap S^+$
  - **else**
    - $p \leftarrow \frac{||x^+ - x||}{||x^+ - x^-||}$
    - With probability p
      - $x \leftarrow x^-, \;\; S \leftarrow S \cap S^-$
    - Otherwise
      - $x \leftarrow x^+, \;\; S \leftarrow S \cap S^+$

Output x

Given any $x \in \mathcal{R}$ and $i^* \neq i, j \in [n]$, let $\phi_x(t) := F(x + t(\mathbb{1}_{\{i\}} - \mathbb{1}_{\{j\}}))$. Then $\phi_x''(t) = -2\nabla^2_{ij} F(x + t(\mathbb{1}_{\{i\}} - \mathbb{1}_{\{j\}})) \geq 0$, since $x + t(\mathbb{1}_{\{i\}} - \mathbb{1}_{\{j\}}) \in \mathcal{R}$. Hence ϕ_x is convex.

Let x be the current point during the rounding procedure, and i, j be the current changing coordinates. The next point is then given by $x' = x + t(\mathbb{1}_{\{i\}} - \mathbb{1}_{\{j\}})$, where t is a random variable such that $\mathbb{E}[t] = 0$. Then conditioning on the current point x and changing coordinates i, j, by Jensen's inequality we get $\mathbb{E}[F(x'|x, i, j)] = \mathbb{E}[\phi_x(t)] \geq \phi_x(0) = F(x)$. Since this is true for any choice of i, j that could be modified at that step, the result follows. $\square$

Note that Corollary 2 and Theorem 6 now prove Theorem 4.

2.5 Future Work

As we mentioned, we conjecture that a polynomial approximation in terms of σ is possible for a general one-sided σ-smooth function. This is the most important potential problem from this section that can be addressed. One way to attack this might be to improve the bound in Lemma 4.

Another interesting future direction is to find more examples of functions that admit one-sided smoothness or sub-domain smoothness. We discuss this a little more in Chapter 4.

Chapter 3

Integrality Gap and Rounding Algorithms

The continuous problem of maximizing a one-sided smooth function in and by itself is interesting. In combinatorial optimization, however, we are mostly interested in optimizing discrete set functions. Specifically we are interested in maximizing set functions whose multi-linear extensions are one-sided smooth. Therefore it is important to know how far the solutions to jump-start continuous greedy are from feasible discrete solutions. For a subclass of one-sided smooth functions, we approved this by designing rounding algorithms to actually find an integral solution based on a fractional solution. For general one-sided smooth functions, we investigate this directly.

Let $\mathcal{I}$ be a set system with the hereditary property and $P_{\mathcal{I}}$ be the convex hull of $\{\mathbb{1}_S : S \in \mathcal{I}\}$. For a function $F : [0,1]^{[n]} \to \mathbb{R}_{\geq 0}$, we say that the *integrality gap* of F over $P_{\mathcal{I}}$ is α if for any $x \in P_{\mathcal{I}}$, there is a $S \in \mathcal{I}$ such that $F(x) \leq \alpha F(\mathbb{1}_S)$. In this chapter, we first discuss the integrality gap of set functions with smooth multi-linear extension. We give an upper-bound for this that compares the best fractional solution with an approximate local optima in the discrete space. After this, we focus on subclasses of functions that have additional smoothness conditions on their first-order derivatives. We give three different rounding algorithms for these functions. We also give worst-case lower-bounds for the integrality gap of such functions. We also discuss the implications of our rounding results to inapproximability results.

3.1 Integrality Gap of One-sided Smooth Multi-Linear Extensions Over Matroids

Our first result bounds the integrality gap over a matroid polytope for the multi-linear extension of a set function when the multi-linear extension is one-sided smooth. The following result bounds the integrality gap of a set function with a ones-sided smooth multi-linear extension over a matroid. This result uses an approximate local optima in a discrete space for such a bound. Hence this implies that a local search (local swap) algorithm that finds an approximate local optima actually finds an approximation for the problem of maximizing a set function with a one-sided σ-smooth multi-linear extension subject to a matroid constraint. We investigate the local search algorithm for a more general class of function in Chapter 4; this requires new analysis and techniques.

We first provide a key lemma for bounding the Taylor series expansion of smooth multi-linear extension. Then we show that the local search algorithm finds a solution which is within $O(\sigma^2 2^\sigma)$-approximation of the optimal solution of the matroid polytope.

Lemma 5. *Let $F : [0,1]^n$ be a one-sided σ-smooth function where $F(\vec{0}) \geq 0$. Then $x^T \nabla F(x) \leq (\sigma+1)F(x)$ and $x^T \nabla^2 F(x) x \leq \sigma(\sigma+1)F(x)$.*

Proof. Given $x \in [0,1]^n$, let $h_x(t) = F(tx)$ and $g_x(t) = x^T \nabla F(tx)$ where $t \in \mathbb{R}$. Note that $g_x(t) = h'_x(t)$ and $x^T \nabla^2 F(tx) x = g'_x(t)$. Since F is one-sided σ-smooth, for $0 \leq t \leq 1$ we have

$$g'_x(t) = x^T \nabla^2 F(tx) x \leq \sigma(\frac{||x||_1}{||tx||_1})(x^T \nabla F(tx)) = \sigma \frac{1}{t} g_x(t).$$

Therefore,

$$t g'_x(t) \leq \sigma g_x(t),$$

and integrating both sides, we get

$$\int_0^1 t g'_x(t) dt \leq \int_0^1 \sigma g_x(t) dt.$$

Applying the integration by parts formula to the left hand side, we get

$$t g_x(t)\Big|_0^1 - \int_0^1 g_x(t) dt \leq \sigma \int_0^1 g_x(t) dt.$$

It follows that

$$1 \cdot g_x(1) - 0 \cdot g_x(0) = x^T \nabla F(x) \leq (\sigma+1) \int_0^1 g_x(t) dt.$$

By using $g_x(t) = h'_x(t)$ we have

$$x^T \nabla F(x) \leq (\sigma+1) \int_0^1 h'_x(t) dt = (\sigma+1)(h_x(1) - h_x(0)) = (\sigma+1)(F(x) - F(\vec{0})) = (\sigma+1)F(x).$$

By one-sided σ-smoothness we have

$$x^T \nabla^2 F(x) x \leq \sigma x^T \nabla F(x).$$

Hence,

$$x^T \nabla^2 F(x) x \leq \sigma(\sigma+1) F(x).$$

□

The following integrality gap result also implies that if the multi-linear extension of a set function is one-sided smooth then the approximate local search algorithm can be used for finding an approximate solution.

Theorem 7. *Let f be a non-negative, monotone set function such that its multi-linear extension F is one-sided σ-smooth, for some non-negative integer σ. Let $\mathcal{M} = ([n], \mathcal{I})$ be a matroid of rank r and P be its associated polytope. Let $x \in P$ be such that $||x||_1 = c$ where $c \in \{1, \ldots, r\}$. Let $S \in \mathcal{I}$ of size c be an approximate local optima such that $S \subseteq supp(x)$, i.e., for any $a \in S$ and $b \in supp(x) \setminus S$ such that $S - a + b \in \mathcal{I}$,*

$$(1 + \frac{\epsilon}{n^2}) f(S) \geq f(S - a + b),$$

where $\epsilon > 0$. Then if $\sigma = O(c)$, $F(x) \leq O(\sigma 2^\sigma) f(S)$ and if $\sigma = \omega(c)$, $F(x) \leq O(\sigma^2 2^\sigma) f(S)$.

Proof. Let $u = (\mathbb{1}_S \vee x) - \mathbb{1}_S$, i.e. $\mathbb{1}_S \vee x = \mathbb{1}_S + u$. It follows that $||u||_1 \leq ||x||_1 = c$. By Taylor's Theorem and Lemma 4 we have that for some $\epsilon \in (0, 1)$

$$F(\mathbb{1}_S \vee x) = F(\mathbb{1}_S + u) = F(\mathbb{1}_S) + u^T \nabla F(\mathbb{1}_S + \epsilon u) \leq F(\mathbb{1}_S) + u^T \nabla F(\mathbb{1}_S) \left(\frac{||\mathbb{1}_S + \epsilon u||_1}{||\mathbb{1}_S||_1}\right)^\sigma.$$

Using that $|S| = c$, $\epsilon \in (0, 1)$, and $||u||_1 \leq c$, we get

$$F(x) \leq F(\mathbb{1}_S \vee x) \leq F(\mathbb{1}_S) + u^T \nabla F(\mathbb{1}_S) \left(\frac{2c}{c}\right)^\sigma \leq f(S) + 2^\sigma u^T \nabla F(\mathbb{1}_S). \tag{3.1}$$

Let $e \in supp(u)$. Because of the exchange property, there is an $a \in S$ such that $S - a + e \in \mathcal{I}$. Because of the selection of S, we know that $(1 + \frac{\epsilon}{n^2}) f(S) \geq f(S - a + e)$. Hence $\frac{\epsilon}{n^2} f(S) + B_a(S - a) \geq B_e(S - a)$. Therefore, We have

$$\begin{aligned}
\nabla_e F(\mathbb{1}_S) = B_e(S) = B_e(S - a) + A_{ae}(S - a) &\leq B_e(S - a) + \sigma(\frac{B_e(S - a) + B_a(S - a)}{c - 1}) \\
&\leq \frac{c - 1 + 2\sigma}{c - 1} B_a(S - a) + \frac{(c - 1 + \sigma)\epsilon}{(c - 1)n^2} f(S)
\end{aligned}$$

Let $S = \{a_1, \ldots, a_c\}$ such that $B_{a_1}(S - a_1) \geq \cdots \geq B_{a_c}(S - a_c)$. Bounding $B_e(S)$ with $B_{a_i}(S - a_i)$ where i is large is better. Let $R_i = \{e^i_1, \ldots, e^i_{k_i}\}$ be the set of elements in $supp(u)$ that are exchangeable with a_i but are not exchangeable with any of $a_{i+1}, \ldots, a_c$. It is obvious that R_i's partition $supp(u)$. Let $t_i = \sum_{e \in R_i} u_e$. By contradiction, we show that if $i \leq c - 1$ then $\sum_{j=1}^i t_j \leq i$. We know that for $R \subseteq [n]$ and $y \in P$ we have $\sum_{e \in R} y_e \leq r_{\mathcal{M}}(R)$ where $r_{\mathcal{M}}$ is the rank function of the matroid. If $\sum_{j=1}^i t_j > i$ then $r_{\mathcal{M}}(\bigcup_{j=1}^i R_i) > i$. This means that there is $R \subseteq \bigcup_{j=1}^i R_i$ such that $|R| \geq i + 1$ and $R \in \mathcal{I}$. Now because of the exchange properties of matroids, we can add elements of S to R until they are the same size. Call this new set R'. Let $T_S = S \setminus R'$ and $T_R = R' \setminus S$. $|T_S| = |T_R| = i + 1$. Therefore, there is a perfect matching of exchangeablity between T_R and T_S [70]. This contradicts our assumption because elements in $\bigcup_{j=1}^i R_i$ are only exchangeable with $a_1, \ldots, a_i$. Now, we have

$$\begin{aligned}
u^T \nabla F(\mathbb{1}_S) = \sum_{e \in supp(u)} u_e \nabla_e F(\mathbb{1}_S) &\leq \sum_{j=1}^c \sum_{e \in R_i} u_e (\frac{c - 1 + 2\sigma}{c - 1} B_{a_j}(S - a_j) + \frac{(c - 1 + \sigma)\epsilon}{(c - 1)n^2} f(S)) \\
&= \sum_{j=1}^c t_j (\frac{c - 1 + 2\sigma}{c - 1} B_{a_j}(S - a_j) + \frac{(c - 1 + \sigma)\epsilon}{(c - 1)n^2} f(S)) \\
&= \frac{c - 1 + 2\sigma}{c - 1} (\sum_{j=1}^c t_j B_{a_j}(S - a_j)) + \frac{c(c - 1 + \sigma)\epsilon}{(c - 1)n^2} f(S).
\end{aligned} \tag{3.2}$$

By Lemma 5, we know that

$$\sum_{j=1}^c B_{a_j}(S - a_j) = \mathbb{1}_S^T \nabla F(\mathbb{1}_S) \leq (\sigma + 1) F(\mathbb{1}_S).$$

We also know that $B_{a_1}(S-a_1) \geq \cdots \geq B_{a_c}(S-a_c)$, $\sum_{j=1}^{c} t_j = ||u||_1 \leq c$, and $\sum_{j=1}^{i} t_j \leq i$ for $i = 1, \ldots, c-1$. Now, we show that

$$\sum_{j=1}^{c} t_j B_{a_j}(S-a_j) \leq (\sigma+1) f(S).$$

We try to find the maximizer of the above. Fix the value of $B_{a_j}(S-a_j)$'s. For any $j < k$, if we increase the value of t_j by ϵ and decrease the value of t_k by ϵ, the value of the summation will increase. This means that the maximum happens when $t_1, \ldots, t_{\lfloor ||u||_1 \rfloor}$ are equal to one and $t_{\lceil ||u||_1 \rceil}$ is equal to $||u||_1 - \lfloor ||u||_1 \rfloor$. Therefore,

$$\sum_{j=1}^{c} t_j B_{a_j}(S-a_j) \leq \sum_{j=1}^{c} B_{a_j}(S-a_j) \leq (\sigma+1) f(S).$$

Therefore, by (3.2), we have

$$u^T \nabla F(\mathbb{1}_S) \leq \frac{c-1+2\sigma}{c-1}(\sigma+1) f(S) + \frac{c(c-1+\sigma)\epsilon}{(c-1)n^2} f(S).$$

Hence, if $\sigma = O(c)$ then $u^T \nabla F(\mathbb{1}_S) \leq O(\sigma) f(S)$ and if $\sigma = \omega(c)$ then $u^T \nabla F(\mathbb{1}_S) \leq O(\sigma^2) f(S)$. Combining this with (3.1) yields the result. □

In the next sections, we discuss rounding algorithms for some subclasses of set functions with a one-sided smooth multi-linear extension.

3.2 Pipage Rounding for Second-Order Smooth Functions

In this section, we consider set functions that have a one-sided σ-smooth multi-linear extension and the third-order derivatives of their multi-linear extension are non-positive. As we mentioned before, these functions are related to second-order submodular function defined in [52].

Definition 3 ([52]). *A set functions $f : 2^{[n]} \to \mathbb{R}$ is called second-order-submodular if $B_i(S \cup R) - B_i(S) \geq B_i(T \cup R) - B_i(T)$ for any $S \subseteq T$, $R \subseteq [n] \setminus T$, and $i \in [n] \setminus (T \cup R)$.*

In Section 3.5, we show super-constant lower bounds for rounding discrete quadratics over matroids. This also implies super-constant lower bounds for the functions we consider in this section. However we show that for uniform matroids, there is a constant-factor rounding algorithm for such functions.

We analyze the pipage rounding algorithm (Algorithm 3.1) for our purpose. This algorithm, in each round, picks two fractional elements and makes (at lease) one of them integral. In case of submodular functions, this is a lossless procedure but as we will see in our case it is not. However we show that the loss is efficiently bounded.

Recall that given a vector $x \in [0,1]^n$ and $i \in [n]$, we denote by $x - i$ the vector resulting from setting the i'th coordinate of x to zero. That is, $(x-i)_j = x_j$ for all $j \neq i$ and $(x-i)_i = 0$.

Lemma 6. *Let f be a set function and F be its multi-linear extension. Let $x \in [0,1]^n$ and $i \neq j \in [n]$ such that $\nabla_i F(x-i-j) \geq \nabla_j F(x-i-j)$. Consider the vector $y = x + \epsilon(\boldsymbol{e}_i - \boldsymbol{e}_j)$, where $\boldsymbol{e}_i$ denotes the characteristic vector of $i \in [n]$, and $\epsilon = \min\{x_j, 1-x_i\}$. That is,*

$$y_k = \begin{cases} x_i + \epsilon = \min\{1, x_i + x_j\}, & k = i \\ x_j - \epsilon = \max\{0, x_i + x_j - 1\}, & k = j \\ x_k, & o.w. \end{cases}$$

Then $F(y) + \max\{0, x_i x_j \nabla^2_{ij} F(x)\} \geq F(x)$.

Proof. For any $z \in [0,1]^n$, we have

$$\begin{aligned}
F(z) &= \sum_{R \subseteq [n]} f(R) \prod_{v \in R} z_v \prod_{v \notin R} (1 - z_v) \\
&= z_i z_j \sum_{R \subseteq [n]-i-j} f(R+i+j) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&+ z_i (1 - z_j) \sum_{R \subseteq [n]-i-j} f(R+i) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&+ (1 - z_i) z_j \sum_{R \subseteq [n]-i-j} f(R+j) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&+ (1 - z_i)(1 - z_j) \sum_{R \subseteq [n]-i-j} f(R) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&= z_i z_j \sum_{R \subseteq [n]-i-j} \big(f(R+i+j) - f(R+i) - f(R+j) + f(R)\big) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&+ z_i \sum_{R \subseteq [n]-i-j} \big(f(R+i) - f(R)\big) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&+ z_j \sum_{R \subseteq [n]-i-j} \big(f(R+j) - f(R)\big) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&+ \sum_{R \subseteq [n]-i-j} f(R) \prod_{v \in R} z_v \prod_{v \notin R+i+j} (1 - z_v) \\
&= z_i z_j \nabla^2_{ij} F(z-i-j) + z_i \nabla_i F(z-i-j) + z_j \nabla_j F(z-i-j) + F(z-i-j).
\end{aligned}$$

Note that $x - i - j = y - i - j$. Also, by definition of ϵ we have $\epsilon \geq x_j - x_i$, and hence

$$y_i y_j = (x_i + \epsilon)(x_j - \epsilon) = x_i x_j + \epsilon(x_j - x_i - \epsilon) \leq x_i x_j.$$

It follows that

$$\begin{aligned}
F(x) &= x_i x_j \nabla^2_{ij} F(x-i-j) + x_i \nabla_i F(x-i-j) + x_j \nabla_j F(x-i-j) + F(x-i-j) \\
&= x_i x_j \nabla^2_{ij} F(y-i-j) + x_i \nabla_i F(y-i-j) + x_j \nabla_j F(y-i-j) + F(y-i-j) \\
&\leq x_i x_j \nabla^2_{ij} F(y-i-j) + y_i \nabla_i F(y-i-j) + y_j \nabla_j F(y-i-j) + F(y-i-j) \\
&= (x_i x_j - y_i y_j) \nabla^2_{ij} F(y-i-j) + y_i y_j \nabla^2_{ij} F(y-i-j) + y_i \nabla_i F(y-i-j) \\
&+ y_j \nabla_j F(y-i-j) + F(y-i-j) \\
&= (x_i x_j - y_i y_j) \nabla^2_{ij} F(x-i-j) + F(y) \\
&\leq (x_i x_j - y_i y_j) \max\{0, \nabla^2_{ij} F(x-i-j)\} + F(y) \\
&\leq x_i x_j \max\{0, \nabla^2_{ij} F(x-i-j)\} + F(y) \\
&= \max\{0, x_i x_j \nabla^2_{ij} F(x)\} + F(y),
\end{aligned}$$

where the first inequality follows from the assumption $\nabla_i F(x-i-j) \geq \nabla_j F(x-i-j)$, and the last equality follows from $\nabla^2_{ij} F(x-i-j) = \nabla^2_{ij} F(x)$ (see Lemma 1). □

Now, using this bound, we are able to analyze the total loss of the pipage rounding algorithm.

Theorem 8. *Let f be a non-negative, monotone, second-order-submodular function and F be its multi-linear extension. Let $x \in [0,1]^n$ such that $||x||_1 = k$. Then Algorithm 3.1 finds $S \subseteq [n]$ such that $|S| = k$ and $6f(S) \geq F(x)$.*

Proof. Let $z \in [0,1]^n$ and z^F be its fractional part (coordinates). Also let z' be z after one step of pipage rounding algorithm (Algorithm 8). By Lemma 6, we have

$$F(z) \leq F(z') + \max\{0, z_i z_j \nabla^2_{ij} F(z)\}. \tag{3.3}$$

By second-order-submodularity, Lemma 16, and monotonicity, we have

$$\frac{1}{2}(z^F)^T \nabla^2 F(z)(z^F) \leq \frac{1}{2}(z^F)^T \nabla^2 F(z^F)(z^F) \leq F(z^F) \leq F(z).$$

Therefore,

$$z_i z_j \nabla^2_{ij} F(z) = \min_{\{q,q'\} \subset supp(z^F)} z_q z_{q'} \nabla_{qq'} F(z) \leq \frac{1}{\binom{|supp(z^F)|}{2}} F(z)$$

Hence, by non-negativity of f, we have

$$\max\{0, z_i z_j \nabla^2_{ij} F(z)\} \leq \frac{1}{\binom{|supp(z^F)|}{2}} F(z)$$

Using this and (3.3), we have

$$\frac{\binom{|supp(z^F)|}{2} - 1}{\binom{|supp(z^F)|}{2}} F(z) \leq F(z') \tag{3.4}$$

Algorithm 3.1: Pipage rounding for functions with non-positive third-order derivative under cardinality constraint

Input: A fractional solution $x = (x_i) \in [0,1]^n$ where $\sum_{i \in [n]} x_i = k$.

while *the sum of fractional coordinates of x is greater than 2* **do**
    $x^F \leftarrow$ fractional coordinates of x;
    $\{i,j\} \leftarrow \arg\min_{\{q,q'\} \subset supp(x^F)} x_q x_{q'} \nabla_{qq'} F(x)$;
    if $x_i + x_j \leq 1$ **then**
        if $\nabla_i F(x - i - j) \geq \nabla_j F(x - i - j)$ **then**
            $x_i \leftarrow x_i + x_j$;
            $x_j \leftarrow 0$;
        else
            $x_j \leftarrow x_i + x_j$;
            $x_i \leftarrow 0$;
    else
        if $\nabla_i F(x - i - j) \geq \nabla_j F(x - i - j)$ **then**
            $x_j \leftarrow x_i + x_j - 1$;
            $x_i \leftarrow 1$;
        else
            $x_i \leftarrow x_i + x_j - 1$;
            $x_j \leftarrow 1$;
$x^F \leftarrow$ fractional coordinates of x;
$x^I \leftarrow$ integral coordinates of x;
$\{i,j\} \leftarrow \arg\max_{\{q,q'\} \subset supp(x^f)} (d(q,q') + g(q) + g(q') + \sum_{v \in supp(x^I)} (d(q,v) + d(q',v)))$;
$S \leftarrow supp(x^I)$;
$\{i,j\} \leftarrow \arg\max_{\{q,q'\} \subset supp(x^F)} B_q(S) + B_{q'}(S) + A_{qq'}(S)$;
$x_i \leftarrow 1$;
$x_j \leftarrow 1$;
for $q \in supp(x^F) - i - j$ **do**
    $x_q \leftarrow 0$;
return $supp(x)$;

Let x^1 be the initial vector in Algorithm 8 and x^{i+1} be the vector after i'th iteration of the loop. Also, let $n_i = |supp(x^i)|$. If the loop iterates t times, we have $n \geq n_1 > n_2 > \cdots > n_t \geq 3$ because in each iteration, the number of integral coordinate increases by at least 1, and $||x^t||_1 > 2$ (the loop's condition). By (3.4), for $i = 1, \ldots, t$, we have $F(x^{i+1}) \geq \frac{n_i^2 - n_i - 2}{n_i(n_i - 1)} F(x^i)$. Let x^{t+2} be the final vector in the algorithm (it is integral). We show that $F(x^{t+2}) \geq \frac{1}{2} F(x^{t+1})$. Let x^F be the fractional part of the x^{t+1}, x^I be its integral part, $S = supp(x^I)$, and

$$\{i,j\} = \underset{\{q,q'\} \subset supp(x^f)}{\arg\max} (B_q(S) + B_{q'}(S) + A_{qq'}(S)).$$

Note that $||x^F||_1 = 2$ because the norm of the fractional part decreases by at most 1 at any iteration and also it is always an integer. Therefore, because of the selection of i, j, we have

$$
\begin{aligned}
&\Big(\sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'}\Big)(B_i(S) + B_j(S) + A_{ij}(S)) \\
&\geq \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'}(B_q(S) + B_{q'}(S) + A_{qq'}(S)) \\
&= \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'} B_q(S) + \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'} B_{q'}(S) + \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'} A_{qq'}(S) \\
&= \sum_{q\in supp(x^F)} \sum_{\substack{q'\in supp(x^F)\\ q'\neq q}} x_q x_{q'} B_q(S) + \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'} A_{qq'}(S) \\
&= \sum_{q\in supp(x^F)} x_q(2-x_q)B_q(S) + \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'} A_{qq'}(S) \\
&\geq \sum_{q\in supp(x^F)} x_q B_q(S) + \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'} A_{qq'}(S) \\
&= (x^F)^T \nabla F(x^I) + \frac{1}{2}(x^F)^T \nabla F(x^I)(x^F)
\end{aligned}
$$

The second inequality holds because $||x^f||_1 = \sum_{q\in supp(x^F)} x_q = 2$ and x_q is fractional, i.e., $x_q < 1$. By the Lagrange multipliers' method and the fact that $\sum_{q\in supp(x^F)} x_q = 2$, we can conclude that

$$
\sum_{\{q,q\}\subset supp(x^F)} x_q x_{q'} \leq 2,
$$

and the equality happens when all $x_q = 2/(|supp(x^F)|)$. Using non-negativity and monotonicity of f, the Taylor's theorem, the above inequalities, and Lemma 3, we have

$$
\begin{aligned}
F(x^{t+1}) &= F(x^I) + (x^F)^T \nabla F(x^I) + \frac{1}{2}(x^F)^T \nabla F(x^I)(x^F) \\
&\leq 2F(x^I) + \Big(\sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'}\Big)(B_i(S) + B_j(S) + A_{ij}(S)) \\
&= \Big(2 - \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'}\Big)F(x^I) \\
&+ \Big(\sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'}\Big)(F(x^I) + B_i(S) + B_j(S) + A_{ij}(S)) \\
&= \Big(2 - \sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'}\Big)F(x^I) + \Big(\sum_{\{q,q'\}\subset supp(x^F)} x_q x_{q'}\Big)F(x^I + \mathbb{1}_{\{i,j\}}) \\
&\leq 2F(x^I + \mathbb{1}_{\{i,j\}}) = 2F(x^{t+2})
\end{aligned}
$$

By the above inequalities, we have

$$\begin{aligned}
F(x^{t+2}) &\geq (\prod_{i=1}^{t} \frac{\binom{n_i}{2}-1}{\binom{n_i}{2}})\frac{1}{2}F(x^1) \geq (\prod_{i=3}^{n} \frac{\binom{i}{2}-1}{\binom{i}{2}})\frac{1}{2}F(x^1) \geq (\prod_{i=3}^{n} \frac{i^2-i-2}{i(i-1)})\frac{1}{2}F(x^1) \\
&= (\prod_{i=3}^{n} \frac{(i+1)(i-2)}{i(i-1)})\frac{1}{2}F(x^1) = \frac{n+1}{3(n-1)}(\prod_{i=3}^{n-1} \frac{(i-1)(i+1)}{(i+1)(i-1)})\frac{1}{2}F(x^1) \\
&= \frac{n+1}{3(n-1)}\frac{1}{2}F(x^1) \geq \frac{1}{6}F(x^1).
\end{aligned}$$

□

In the following sections, we investigate rounding algorithms for general matroid constraints.

3.3 Quadratic Coverage Rounding for Discrete Quadratic Functions

In this section and the next one, we consider the discrete quadratic functions and their multi-linear extensions. Let $\mathcal{M} = ([n], \mathcal{I})$ be a matroid and $P_{\mathcal{M}}$ be its polytope. We consider the integrality gap for the quadratic program: $\max F(x) : x \in P_{\mathcal{M}}$. Here F is a non-negative, quadratic multi-linear function $F(x) = \frac{1}{2}x^T Ax + b^T x$ such that $A, b \geq 0$ and A is a symmetric, zero diagonal matrix.

This class is of interest for a variety of reasons. It is a natural family since these are just restrictions to the hypercube of quadratic forms $\frac{1}{2}x^T Ax + b^T x$. This family also coincides with the class of *second-order-modular* functions introduced in [52].

Definition 4 ([52])**.** *A set functions $f : 2^{[n]} \to \mathbb{R}$ is called second-order modular if $B_i(S \cup R) - B_i(S) = B_i(T \cup R) - B_i(T)$ for any $S \subseteq T$, $R \subseteq [n] \setminus T$, and $i \in [n] \setminus (T \cup R)$.*

The following lemma characterize the structure of second-order modular functions.

Lemma 7. *f is a second-order modular function if and only if there exist symmetric $d : [n] \times [n] \to \mathbb{R}$, and $g : [n] \to \mathbb{R}$ such that*

$$f(R) = \sum_{\{i,j\} \subset R} d(i,j) + \sum_{i \in R} g(i).$$

If f is also supermodular (submodular), then d is non-negative (non-positive).

Proof. Sufficiency is easy since

$$\begin{aligned}
B_i(S \cup R) - B_i(S) &= (g(i) + \sum_{m \in S \cup R} d(m,i)) - (g(i) + \sum_{m \in S} d(m,i)) = \sum_{m \in R} d(m,i) \\
&= (g(i) + \sum_{m \in T \cup R} d(m,i)) - (g(i) + \sum_{m \in T} d(m,i)) \\
&= B_i(T \cup R) - B_i(T).
\end{aligned}$$

To prove necessity, we first show that if $i, j \in [n]$ and $S \subseteq [n] - i - j$ then, by second-order modularity

$$B_j(S+i) - B_j(S) = B_j([n]-j) - B_j([n]-i-j),$$

because $S \subseteq [n]-i$ ($[n]-i$ plays the role of T in the definition of second-order modular). Now, let $d(i,j) = B_j([n]-j) - B_j([n]-i-j)$ and $g(i) = B_i(\emptyset)$. Note that d is symmetric because

$$\begin{aligned}
d(i,j) &= B_j([n]-j) - B_j([n]-i-j) = (f([n]) - f([n]-j)) - (f([n]-i) - f([n]-i-j)) \\
&= (f([n]) - f([n]-i)) - (f([n]-j) - f([n]-i-j)) \\
&= B_i([n]-i) - B_i([n]-i-j) = d(j,i).
\end{aligned}$$

For any m, let $R_m = \{v_1, \ldots, v_m\}$, and set $R_0 = \emptyset$. Consider a set $R = \{v_1, \ldots, v_r\}$. Then we have

$$\begin{aligned}
f(R) &= \sum_{m=0}^{r-1} (f(R_m + v_{m+1}) - f(R_m)) = \sum_{m=0}^{r-1} B_{v_{m+1}}(R_m) \\
&= \sum_{m=0}^{r-1} (\sum_{t=1}^{m} (B_{v_{m+1}}(R_t) - B_{v_{m+1}}(R_{t-1})) + B_{v_{m+1}}(R_0)) && \text{telescoping sum} \\
&= \sum_{m=0}^{r-1} (\sum_{t=1}^{m} (B_{v_{m+1}}([n]-v_{m+1}) - B_{v_{m+1}}([n]-v_t-v_{m+1})) + B_{v_{m+1}}(R_0)) \\
&= \sum_{m=0}^{r-1} \sum_{t=1}^{m} d(v_t, v_{m+1}) + \sum_{m=0}^{r-1} g(v_{m+1}).
\end{aligned}$$

If f is supermodular, $i, j \in [n]$, and $R \subseteq [n] - i - j$, we have

$$f(R+i+j) - f(R+i) \geq f(R+j) - f(R).$$

Therefore,

$$g(j) + \sum_{v \in R+i} d(v,j) \geq g(j) + \sum_{v \in R} d(v,j),$$

which means $d(i,j) \geq 0$. Similarly, if f is submodular, d is non-positive. □

In the special case when $b = 0$ and $A(u,v)$ forms a metric, the discrete quadratic class corresponds to metric diversity functions and, as pointed out, the maximization problem over a matroid constraint has a 2-approximation [1, 11]. As we established in Proposition 2, discrete quadratics have interesting behaviour with respect to their one-sided smoothness. The previous mentioned metric diversity functions have one-sided smoothness $\sigma = 2$. *Negative type distances* are another important class of distance functions.

Definition 5. *Let $d : [n] \times [n] \to \mathbb{R}_{\geq 0}$ be a distance function with the corresponding distance matrix $D \in \mathbb{R}_{\geq 0}^{n \times n}$ where $D_{a,b} = d(a,b)$. We say d is a negative-type distance if for any $x \in \mathbb{R}^n$ with $||x||_1 = 0$ we have $x^T D x \leq 0$.*

If A is a negative type distance, then the corresponding problems have been shown to admit a PTAS [19, 20]. Another well-known distance measure is the Jensen-Shannon divergence used to measure dis-similarity of two probability distributions. Both JS and negative-type distances have associated smoothness parameter $\sigma = 4$.

Proposition 6. *Any negative-type distance $d : [n] \times [n] \to \mathbb{R}_{\geq 0}$ is 2-semi-metric.*

Proof. Let $x = 0.5e_a + 0.5e_b - e_c$. We know

$$x^T Dx = 0.5d(a, b) - d(a, c) - d(b, c) \leq 0.$$

Therefore $d(a, b) \leq 2d(a, c) + 2d(b, c)$ and d is 2-semi metric. □

Jensen-Shannon Divergence is a function which measures dis-similarity between probability distributions. It is well-known that if d is a JS measure, then $\sqrt{d}$ is a metric. Hence JS distances form a 2-semi-metric by the following result.

Proposition 7. *Let $d : [n] \times [n] \to \mathbb{R}_{\geq 0}$ be a distance function such that $\sqrt{d(\cdot, \cdot)}$ is a metric. Then $d(\cdot, \cdot)$ is a 2-semi-metric.*

Proof. By definition, we have

$$\sqrt{d(i, j)} \leq \sqrt{d(i, k)} + \sqrt{d(j, k)}.$$

Therefore,

$$d(i, j) \leq d(i, k) + d(j, k) + 2\sqrt{d(i, k)d(j, k)}.$$

We also know that

$$d(i, k) + d(j, k) - 2\sqrt{d(i, k)d(j, k)} = (\sqrt{d(i, k)} - \sqrt{d(j, k)})^2 \geq 0.$$

Hence,

$$d(i, j) \leq 2(d(i, k) + d(j, k)).$$

□

For general $\sigma \geq 0$, let $\mathcal{O}_\sigma$ denote the family of discrete quadratic functions which are one-sided σ-smooth. Now that we have seen some examples for such functions, we discuss the rounding algorithms for them.

Gaps for such quadratic programmes may be unbounded even for graphic matroids if we allow parallel edges. Fortunately these large gaps transpire due to a simple reason, namely that the matroids have very small circuits. This is encapsulated in the following integrality gap upper bound.

Theorem 9. *Let F be a non-negative, quadratic multi-linear polynomial and $\mathcal{M}$ be a matroid with rank r and minimum circuit size $c \geq 3$. If $x^* \in P_{\mathcal{M}}$, then there is an independent set I of $\mathcal{M}$ such that $(3 + \frac{2r}{c-2})F(\mathbb{1}_I) \geq F(x^*)$.*

We actually prove the following decomposition result. For $x^* \in P_{\mathcal{M}}$, we define the *coverage* of a pair u, v to be the quantity $x^*(u)x^*(v)$. Let $Cov \in \mathbb{R}^{\binom{n}{2}}$ be the vector with entries $Cov(u,v) = x^*(u)x^*(v)$. As F is quadratic it is linear in these coverage values and the vector x^*: $F(x^*) = \sum_{u \neq v}(\frac{A(u,v)}{2})Cov(u,v) + \sum_v b(v)x^*(v)$. For a set X we say its *coverage set* is $cov(X) = \{\{u,v\} : u,v \in X, u \neq v\}$. A *quadratic coverage* of x^* is a collection $\mathcal{C} = \{\mathbb{1}_{I_i}, \mu_i\}$ of weighted independent sets with properties (1) for each $u \neq v$, $\sum_{i:\{u,v\} \subseteq cov(I_i)} \mu_i \geq Cov(u,v)$, and (2) for each v, $\sum_{i:I_i \ni v} \mu_i \geq x^*(v)$. Recall that $A, b \geq 0$. It follows that $\sum_i \mu_i F(\mathbb{1}_{I_i}) \geq F(x^*)$ and hence if the *size* $\sum_i \mu_i \leq K$, then some I_i satisfies $F(\mathbb{1}_{I_i}) \geq \frac{F(x^*)}{K}$. This bound depends on the fact that entries of A are non-negative. By condition (1) of quadratic coverages, we have $\sum_i \mu_i \mathbb{1}_{cov(I_i)} \geq Cov$ and by condition (2), $\sum_i \mu_i \mathbb{1}_{I_i} \geq x^*$. Therefore, for such a collection we have $\sum_i \mu_i F(\mathbb{1}_{I_i}) \geq F(x^*)$. This reasoning shows that to deduce Theorem 9, it suffices to find a quadratic coverage with $\sum_i \mu_i \leq (3 + \frac{2r}{c-2})$.

Theorem 10. *Let $F(x) = \frac{1}{2}x^T Ax + b^T x$ be a non-negative, quadratic multi-linear polynomial and $\mathcal{M}$ be a matroid with rank $r = r([n])$ and minimum circuit size $c \geq 3$. If $x^* \in P_{\mathcal{M}}$, then it has a quadratic coverage of size at most $3 + \frac{2r}{c-2}$.*

Proof. We start with an arbitrary representation of x^* as a convex combination of independent sets: $\sum_i \lambda_i \mathbb{1}_{B_i}$.

First note that $Cov(u,v) = (\sum_{B_i \ni u} \lambda_i)(\sum_{B_j \ni v} \lambda_j) = \sum_{(i,j):B_i \ni u, B_j \ni v} \lambda_i \lambda_j$. Hence an ordered pair (B_i, B_j) contributes $\lambda_i \lambda_j$ to $Cov(u,v)$ if $u \in B_i, v \in B_j$. This implies that if $B_i = B_j$, then this contributes exactly λ_i^2 for every $u, v \in B_i$. If $B_i \neq B_j$, then the unordered pair $\{B_i, B_j\}$ contributes to coverages as follows. It contributes $2\lambda_i\lambda_j$ for every $u, v \in B_i \cap B_j$ and $\lambda_i \lambda_j$ for each $uv \in \delta(B_i - B_j, B_j - B_i, B_i \cap B_j)$. Here for disjoint node sets $X_1, X_2, \ldots, X_p$ we define $\delta(X_1, X_2, \ldots, X_p)$ to be the set of edges which have endpoints in distinct sets from the X_i's. Hence we can express the coverage vector Cov for x^* in $\mathbb{R}^{\binom{n}{2}}$ as:

$$\sum_i \lambda_i^2 \cdot \mathbb{1}_{cov(B_i)} + \sum_{i<j} \lambda_i \lambda_j \cdot (2 \cdot \mathbb{1}_{cov(B_i \cap B_j)} + \mathbb{1}_{\delta(B_i - B_j, B_j - B_i, B_i \cap B_j)}). \tag{3.5}$$

We now define a quadratic coverage, that is, a weighted collection of independent sets satisfying conditions (1) and (2). In particular, for each $i \leq j$ we define a family of independent sets $\mathcal{I}^{i,j}$ which will take care of all coverages associated with terms $\lambda_i \lambda_j$ in (3.5). In the case where $i = j$, this is easy. We just include the set B_i with weight $\mu_i = \lambda_i^2$. Now consider the case where $i < j$ which is trickier. For each set I in this family, we always associate the weight $\mu_I = \lambda_i \lambda_j$ and so this amounts to finding a family which satisfies

$$\sum_{I \in \mathcal{I}^{i,j}} \mathbb{1}_{cov(I)} \geq 2 \cdot \mathbb{1}_{cov(B_i \cap B_j)} + \mathbb{1}_{\delta(B_i - B_j, B_j - B_i, B_i \cap B_j)}. \tag{3.6}$$

We return to this construction later but we note that condition (2) will follow easily as long as we guarantee that for each v, i and $j \neq i$, if $B_i \ni v$, then the family $\mathcal{I}^{i,j}$ includes at least one set I which contains v. Since we have $\mu_I = \lambda_i \lambda_j$ for any such I, we derive the desired inequality (2): $\sum_{I \ni v} \mu_I \geq \sum_{B_i \ni v}(\sum_j \lambda_i \lambda_j) = \sum_{B_i \ni v} \lambda_i = x^*(v)$.

If we can achieve this construction so that $|\mathcal{I}^{i,j}| \leq K$ for each i, j, then we have a quadratic coverage whose size is $\sum_i \mu_i + \sum_{i<j} \sum_{I \in \mathcal{I}^{i,j}} \mu_I = \sum_i \lambda_i^2 + \sum_{i<j} \lambda_i \lambda_j |\mathcal{I}^{i,j}| \leq \sum_i \lambda_i^2 + \sum_{i<j} \lambda_i \lambda_j K \leq 1 + K/2$. The last inequality follows since the λ_i are a convex combination.

We now define $\mathcal{I}^{i,j}$ for a fixed pair i, j and show how to find the desired independent sets $\mathcal{I}^{i,j} = \{I_k^{i,j} : k = 1, 2, \ldots, K\}$, where K is defined later. First, if $|B_i \cap B_j| \geq 1$, then we include the sets B_i, B_j. This takes care of the double-coverage of pairs in $B_i \cap B_j$ as well as any pairs u, v with $u \in B_i \cap B_j$ and $v \in B_i \Delta B_j$. Let $S_{ij} = B_i \setminus B_j$ and $S_{ji} = B_j \setminus B_i$. Note that the excess coverage from these sets B_i, B_j is to contribute an extra $\lambda_i \lambda_j$ to each pair in $cov(S_{ij}) \cup cov(S_{ji})$. It now remains to cover the edges in $\delta(S_{ij}, S_{ji})$.

Let $t = \lfloor (c-1)/2 \rfloor$ and $m = |B_i \cap B_j| \geq 0$. Decompose $B_j \setminus B_i$ into $\ell = \lceil (r-m)/t \rceil$ disjoint independent sets by ripping out sets of size t greedily, possibly the last being smaller than t. Call these $C_1, C_2, \ldots, C_\ell$. For each $k \leq \ell$, we extend C_k to an independent set $R_k^{i,j}$ in $B_i \Delta B_j$ only adding elements from $B_i \setminus B_j$. Hence this set will have used all elements of B_i except a subset, call it Z_k, of size at most t. Let $C_k^{i,j} = Z_k \cup C_k$ and note that $|C_k^{i,j}| \leq 2t \leq c - 1$ and hence it is also independent. We now examine the pairs covered by $C_k^{i,j}, R_k^{i,j}$. Let $u \in C_k, v \in B_i \setminus B_j$, then either u, v is covered by $R_k^{i,j}$, or $v \in Z_k$ in which case it is covered by $C_k^{i,j}$.

Finally, we count the number of sets for a given family. There are two cases depending on whether $B_i \cap B_j = \emptyset$ or not. If the intersection is empty, then we just build $2\lceil \frac{r}{t} \rceil$. Since $t \geq \frac{c-2}{2}$, this is at most $2 \cdot (1 + \frac{2r}{c-2})$. In the other case we have $m \geq 1$, and we add the sets B_i, B_j up front and then we add $2\lceil \frac{r-m}{t} \rceil$ more sets. Hence the overall number of sets in this case is at most $2 + 2 \cdot (\frac{2r}{c-2} - \frac{2}{c-2} + 1)$.

It follows that $K \leq 2 \cdot (2 + \frac{2r}{c-2})$, and thus we have a quadratic coverage of size at most $1 + \frac{K}{2} \leq 3 + \frac{2r}{c-2}$, as we wanted to show.

□

The bound given in Theorem 9 is good for matroids that do not have small circuits. For example, in case of uniform matroids, the size of the smallest circuit is one more than the rank of the matroid. Therefore this bound is actually constant for uniform matroids. However for general matroids the rank could be large and the size of the smallest circuit be small. In the next section, we give another bound that works better for such matroids.

3.4 Swap Rounding For Quadratic Multi-Linear Extensions

The swap rounding algorithm is previously used for rounding modular and submodular functions over matroid polytopes and other combinatorial structures [22, 71]. In this section, we analyze a modified version of the swap rounding algorithm (Algorithm 3.2) and we show that it finds an integral solution which is an $O(1 + \frac{\sigma}{r})$-approximation of the initial fractional solution. This algorithm starts from a convex combination of the bases of the matroid and in each round, it merges two of the bases in the convex combination. Note that by Carathéodory's theorem, any maximal point in the matroid polytope can be written as the convex combination of characteristic vectors of $n + 1$ bases [37].

First we define some notation. Let $d(S) = \sum_{\{i,j\} \subseteq S} d(i, j)$ and $d(S, S') = \sum_{i \in S} \sum_{j \in S'} d(i, j)$ and $g(S) = \sum_{i \in S} g(i)$. The following result provides a decomposition of the multi-linear exten-

Algorithm 3.2: Swap rounding for monotone second-order-modular functions under matroid constraints

Input: A matroid $\mathcal{M} = ([n], \mathcal{I})$, its base polytope P, and a fractional solution $x \in P$. A set function $f(S) = \sum_{i \in S} g(i) + \sum_{\{i,j\} \subseteq S} d(i,j)$.
Find $\boldsymbol{\lambda}_1 = (\lambda_1, \lambda_2, \ldots, \lambda_p)$ and $\boldsymbol{I}_1 = (I_1, I_2, \ldots, I_p)$ such that $x = \sum_{i=1}^{p} \lambda_i I_i$, $\lambda_i \geq 0$ (for any i), $\sum_{i=1}^{p} \lambda_i = 1$, and I_i's are bases of the matroid;
$I'_1 \leftarrow I_1$;
$\lambda'_1 \leftarrow \lambda_1$;
for $k = 1, \ldots, p-1$ **do**
    $(I'_{k+1}, M_k) \leftarrow$ `MergeBases(`$\boldsymbol{I}_k$,$\boldsymbol{\lambda}_k$`)`;
    $\lambda'_{k+1} \leftarrow \lambda'_k + \lambda_{k+1}$;
    $\boldsymbol{I}_{k+1} \leftarrow (I'_{k+1}, I_{k+2}, \ldots, I_p)$;
    $\boldsymbol{\lambda}_{k+1} \leftarrow (\lambda'_{k+1}, \lambda_{k+2}, \ldots, \lambda_p)$;
$t \leftarrow \arg\max_{k=1,\ldots,p-1} \{\sum_{(i,j) \in M_k} d(i,j)\}$;
$(I^*, M^*) \leftarrow$ `MergeBases(`(I'_t, I_{t+1}),$(0.5, 0.5)$`)`;
return $\arg\max\{f(I^*), f(I'_p)\}$;

Function `MergeBases(`$\boldsymbol{I} = (I_1, I_2, \ldots, I_m)$, $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)$`):`
    $M \leftarrow \emptyset$;
    while $I_1 \neq I_2$ **do**
        Pick $i \in I_1 \setminus I_2$ and $j \in I_2 \setminus I_1$ such that $I_1 - i + j \in \mathcal{I}$ and $I_2 - j + i \in \mathcal{I}$;
        $M \leftarrow M \cup \{(i,j)\}$;
        if $g(i) + \lambda_1 d(i, I_1 - i) + \lambda_2 d(i, I_2 - j) + \sum_{k=3}^{m} \lambda_k d(i, I_k) \geq g(j) + \lambda_1 d(j, I_1 - i) + \lambda_2 d(j, I_2 - j) + \sum_{k=3}^{m} \lambda_k d(j, I_k)$ **then**
            $I_2 \leftarrow I_2 - j + i$;
        else
            $I_1 \leftarrow I_1 - i + j$;
    return (I_1, M);
End Function

sion of a quadratic function based on the convex decomposition of a point to the bases of the matroid.

Lemma 8. *Let $f(S) = \sum_{i \in S} g(i) + \sum_{\{i,j\} \subseteq S} d(i,j)$ where $g : [n] \to \mathbb{R}_{\geq 0}$ and $d : [n] \times [n] \to \mathbb{R}_{\geq 0}$ with $d(i,i) = 0$ for all $i \in [n]$. Let $b \in \mathbb{R}^n$ be a vector such that $b_i = g(i)$ and $A \in \mathbb{R}^{n \times n}$ be a matrix such that $A_{ij} = d(i,j)$. Then the multi-linear extension of f is $F(x) = \frac{1}{2} x^T A x + x^T b$. Moreover, if $x = \sum_{k=1}^{p} \lambda_k \mathbb{1}_{I_k}$ for some scalars λ_k's and subsets $I_k \subseteq [n]$, then*

$$F(x) = \sum_{k=1}^{p} \lambda_k g(I_k) + \sum_{k=1}^{p} \lambda_k^2 d(I_k) + \sum_{k=1}^{p-1} \sum_{\ell=k+1}^{p} \lambda_k \lambda_\ell d(I_k, I_\ell). \tag{3.7}$$

Proof. For the first part of the lemma note that

$$\begin{aligned}
F(x) &= \sum_{S\subseteq[n]} f(S)\prod_{k\in S} x_k \prod_{k\in[n]\setminus S}(1-x_k) = \sum_{S\subseteq[n]}(g(S)+d(S))\prod_{k\in S} x_k \prod_{k\in[n]\setminus S}(1-x_k) \\
&= \sum_{S\subseteq[n]}(\sum_{i\in S} g(i))\prod_{k\in S} x_k \prod_{k\in[n]\setminus S}(1-x_k) + \sum_{S\subseteq[n]}(\sum_{\{i,j\}\subseteq S} d(i,j))\prod_{k\in S} x_k \prod_{k\in[n]\setminus S}(1-x_k) \\
&= \sum_{i\in[n]} g(i) \sum_{\substack{S\subseteq[n]\\ i\in S}}(\prod_{k\in S} x_k \prod_{k\in[n]\setminus S}(1-x_k)) + \sum_{\{i,j\}\subseteq[n]} d(i,j) \sum_{\substack{S\subseteq[n]\\ \{i,j\}\subseteq S}}(\prod_{k\in S} x_k \prod_{k\in[n]\setminus S}(1-x_k)).
\end{aligned}$$

Hence

$$\begin{aligned}
F(x) = &\sum_{i\in[n]} g(i)x_i \sum_{S\subseteq[n]-i}(\prod_{k\in S} x_k \prod_{k\in[n]-i\setminus S}(1-x_k)) \\
&+ \sum_{\{i,j\}\subseteq[n]} d(i,j)x_ix_j \sum_{S\subseteq[n]-i-j}(\prod_{k\in S} x_k \prod_{k\in[n]-i-j\setminus S}(1-x_k)) \\
&= \sum_{i\in[n]} g(i)x_i + \sum_{\{i,j\}\subseteq[n]} d(i,j)x_ix_j = x^Tb + \frac{1}{2}x^TAx.
\end{aligned}$$

To see the second part, observe that

$$b^Tx = b^T(\sum_k \lambda_k \mathbb{1}_{I_k}) = \sum_k \lambda_k(b^T\mathbb{1}_{I_k}) = \sum_k \lambda_k g(I_k),$$

and

$$\begin{aligned}
x^TAx &= (\sum_{k=1}^{p}\lambda_k\mathbb{1}_{I_k})A(\sum_{\ell=1}^{p}\lambda_\ell\mathbb{1}_{I_\ell}) = \sum_{k,\ell=1}^{p}\lambda_k\lambda_\ell\mathbb{1}_{I_k}A\mathbb{1}_{I_\ell} = \sum_{k,\ell=1}^{p}\lambda_k\lambda_\ell d(I_k,I_\ell) \\
&= \sum_{k=1}^{p}\lambda_k^2 d(I_k,I_k) + 2\sum_{k<\ell}\lambda_k\lambda_\ell d(I_k,I_\ell) = 2\sum_{k=1}^{p}\lambda_k^2 d(I_k) + 2\sum_{k=1}^{p-1}\sum_{\ell=k+1}^{p}\lambda_k\lambda_\ell d(I_k,I_\ell).
\end{aligned}$$

□

Using this we can bound the loss of each merge in the swap rounding algorithm.

Lemma 9. *Let $\mathcal{M} = ([n],\mathcal{I})$ be a matroid and P be its corresponding base polytope. Let $F(z) = \frac{1}{2}z^TAz + z^Tb$ where $A, b \geq 0$ and A is a symmetric matrix such that its diagonal is zero. Let $f(S) = F(\mathbb{1}_S)$ for any $S \subseteq [n]$. Let $x = \sum_{i=1}^{p}\lambda_i\mathbb{1}_{I_i} \in P$ where I_i's are bases of the matroid, $\sum_{i=1}^{p}\lambda_i = 1$, and $\lambda_i \geq 0$, for $i = 1,\dots,p$. Let (I', M) be the output of* MergeBases *(defined in Algorithm 3.2) on $(I_1,\dots,I_p)$ and $(\lambda_1,\dots,\lambda_p)$. Let $y = (\lambda_1+\lambda_2)\mathbb{1}_{I'} + \sum_{i=3}^{p}\lambda_i\mathbb{1}_{I_i}$. Then $F(x) \leq F(y) + \lambda_1\lambda_2\sum_{(i,j)\in M} d(i,j)$.*

Proof. Let $I_1^0 = I_1$ and $I_2^0 = I_2$ (the original inputs of the function). Let I_1^m and I_2^m be the resulting I_1 and I_2 after the m-th iteration of the while loop. Let $x_m = \lambda_1\mathbb{1}_{I_1^m} + \lambda_2\mathbb{1}_{I_2^m} +$

$\sum_{k=3}^{p} \lambda_k \mathbb{1}_{I_k}$. Let i_m, j_m be the elements we pick at the m-th iteration of the loop. We show that $F(x_{m-1}) \leq F(x_m) + \lambda_1\lambda_2 d(i_m, j_m)$ and this yields the desired result using a simple recursion argument. Without loss of generality, we assume

$$\begin{aligned}
&g(i_m) + \lambda_1 d(i_m, I_1^{m-1} - i_m) + \lambda_2 d(i_m, I_2^{m-1} - j_m) + \sum_{k=3}^{p} \lambda_k d(i_m, I_k) \\
&\geq g(j_m) + \lambda_1 d(j_m, I_1^{m-1} - i_m) + \lambda_2 d(j_m, I_2^{m-1} - j_m) + \sum_{k=3}^{p} \lambda_k d(j_m, I_k)
\end{aligned} \tag{3.8}$$

We have

$$\begin{aligned}
F(x_{m-1}) &= \lambda_1 g(I_1^{m-1}) + \lambda_2 g(I_2^{m-1}) + \sum_{k=3}^{p} \lambda_k g(I_k) + \lambda_1^2 d(I_1^{m-1}) + \lambda_2^2 d(I_2^{m-1}) + \sum_{k=3}^{p} \lambda_k^2 d(I_k) \\
&+ \lambda_1\lambda_2 d(I_1^{m-1}, I_2^{m-1}) + \lambda_1 \sum_{k=3}^{p} \lambda_k d(I_1^{m-1}, I_k) \\
&+ \lambda_2 \sum_{k=3}^{p} \lambda_k d(I_2^{m-1}, I_k) + \sum_{k=3}^{p-1} \sum_{k'=k+1}^{p} \lambda_k \lambda_{k'} d(I_k, I_{k'}) \\
&= \lambda_1 g(I_1^{m-1}) + \lambda_2 g(I_2^{m-1} - j_m) + \sum_{k=3}^{p} \lambda_k g(I_k) + \lambda_1^2 d(I_1^{m-1}) + \lambda_2^2 d(I_2^{m-1} - j_m) + \sum_{k=3}^{p} \lambda_k^2 d(I_k) \\
&+ \lambda_1\lambda_2 d(I_1^{m-1}, I_2^{m-1} - j_m) + \lambda_1 \sum_{k=3}^{p} \lambda_k d(I_1^{m-1}, I_k) + \lambda_2 \sum_{k=3}^{p} \lambda_k d(I_2^{m-1} - j_m, I_k) \\
&+ \sum_{k=3}^{p-1} \sum_{k'=k+1}^{p} \lambda_k \lambda_{k'} d(I_k, I_{k'}) + \lambda_2 g(j_m) + \lambda_2^2 d(j_m, I_2^{m-1} - j_m) + \lambda_1\lambda_2 d(j_m, I_1^{m-1} - i_m) \\
&+ \lambda_2 \sum_{k=3}^{p} \lambda_k d(j_m, I_k) + \lambda_1\lambda_2 d(i_m, j_m) \\
&\leq \lambda_1 g(I_1^{m-1}) + \lambda_2 g(I_2^{m-1} - j_m) + \sum_{k=3}^{p} \lambda_k g(I_k) + \lambda_1^2 d(I_1^{m-1}) + \lambda_2^2 d(I_2^{m-1} - j_m) \\
&+ \sum_{k=3}^{p} \lambda_k^2 d(I_k) + \lambda_1\lambda_2 d(I_1^{m-1}, I_2^{m-1} - j_m) + \lambda_1 \sum_{k=3}^{p} \lambda_k d(I_1^{m-1}, I_k) + \lambda_2 \sum_{k=3}^{p} \lambda_k d(I_2^{m-1} - j_m, I_k) \\
&+ \sum_{k=3}^{p-1} \sum_{k'=k+1}^{p} \lambda_k \lambda_{k'} d(I_k, I_{k'}) + \lambda_2 g(i_m) + \lambda_2^2 d(i_m, I_2^{m-1} - j_m) + \lambda_1\lambda_2 d(i_m, I_1^{m-1} - i_m) \\
&+ \lambda_2 \sum_{k=3}^{p} \lambda_k d(i_m, I_k) + \lambda_1\lambda_2 d(i_m, j_m),
\end{aligned}$$

where the first equality follows from Lemma 8 and the inequality holds because of (3.8). By the above inequality, and the fact that $I_1^m = I_1^{m-1}$ and $I_2^m = I_2^{m-1} - j_m + i_m$, we have

$$\begin{aligned}F(x_{m-1}) \le \lambda_1 g(I_1^m) + \lambda_2 g(I_2^m) + \sum_{k=3}^{p} \lambda_k g(I_k) + \lambda_1^2 d(I_1^m) + \lambda_2^2 d(I_2^m) + \sum_{k=3}^{p} \lambda_k^2 d(I_k)\\ + \lambda_1\lambda_2 d(I_1^m, I_2^m) + \lambda_1 \sum_{k=3}^{p} \lambda_k d(I_1^m, I_k) + \lambda_2 \sum_{k=3}^{p} \lambda_k d(I_2^m, I_k)\\ + \sum_{k=3}^{p-1} \sum_{k'=k+1}^{p} \lambda_k \lambda_{k'} d(I_k, I_{k'}) + \lambda_1\lambda_2 d(i_m, j_m).\end{aligned}$$

By Lemma 8, the right hand side is equal to $F(x_m) + \lambda_1\lambda_2 d(i_m, j_m)$ and therefore

$$F(x_{m-1}) \le F(x_m) + \lambda_1\lambda_2 d(i_m, j_m).$$

□

Using these, we can bound the total loss of the swap rounding algorithm in the next theorem.

Theorem 11. *Let $\mathcal{M}([n], \mathcal{I})$ be a matroid of rank r and P be its corresponding base polytope. Let $F(z) = \frac{1}{2} z^T A z + z^T b$ where $A, b \ge 0$ and A is a symmetric matrix with zero diagonal that satisfies the σ-semi-metric inequality, i.e., $A_{ij} \le \sigma(A_{ik} + A_{jk})$. Let $f(S) = F(\mathbb{1}_S)$ for any $S \subseteq [n]$. Let $x \in P$ and S be the output of the modified swap rounding (Algorithm 3.2) on x. Then $F(x) \le O(1 + \frac{\sigma}{r}) f(S)$.*

Proof. Let $x = \sum_{i=1}^p \lambda_i \mathbb{1}_{I_i} \in P$ where I_i's are bases of the matroid, $\sum_{i=1}^p \lambda_i = 1$, and $\lambda_i \ge 0$, for $i = 1, \dots, p$. Let S be the output of the swap rounding (Algorithm 3.2) if it starts from $(I_1, \dots, I_p)$ and $(\lambda_1, \dots, \lambda_p)$. Let x_k denote the vector corresponding to $\boldsymbol{I}_k = (I_{k'}, I_{k+1}, \dots, I_p)$ and $\boldsymbol{\lambda}_k = (\lambda_{k'}, \lambda_{k+1}, \dots, \lambda_p)$, i.e. $x_k = \lambda'_k \mathbb{1}_{I'_k} + \sum_{i=k+1}^p \lambda_i \mathbb{1}_{I_i}$. By Lemma 9, for $k = 1, \dots, n-1$, we have

$$F(x_k) \le F(x_{k+1}) + \lambda'_k \lambda_{k+1} \sum_{(i,j)\in M_k} d(i,j) \le F(x_{k+1}) + \lambda'_k \lambda_{k+1} \sum_{(i,j)\in M_t} d(i,j),$$

where $t = \arg\max_{k=1,\dots,p-1} \{\sum_{(i,j)\in M_k} d(i,j)\}$. Therefore

$$\begin{aligned}F(x_1) &\le F(x_p) + (\sum_{k=1}^{p-1} \lambda'_k \lambda_{k+1}) \sum_{(i,j)\in M_t} d(i,j) = F(x_p) + (\sum_{k=1}^{p-1}\sum_{m=1}^{k} \lambda_m \lambda_{k+1}) \sum_{(i,j)\in M_t} d(i,j)\\ &\le F(x_p) + \frac{1}{2} \sum_{(i,j)\in M_t} d(i,j) = f(I'_p) + \frac{1}{2} \sum_{(i,j)\in M_t} d(i,j),\end{aligned} \tag{3.9}$$

where the last inequality holds since $2\sum_{k=1}^{p-1}\sum_{m=1}^{k} \lambda_m \lambda_{k+1} \le (\sum_{k=1}^{p} \lambda_k)^2 = 1$. Now, we bound the term $\sum_{(i,j)\in M_t} d(i,j)$. By definition of M_t, note that $M_t \subseteq I'_t \times I_{t+1}$. Using this and Lemma 8 it follows that

$$\sum_{(i,j)\in M_t} d(i,j) \le d(I'_t, I_{t+1}) \le 4 \cdot F(\frac{1}{2}\mathbb{1}_{I'_t} + \frac{1}{2}\mathbb{1}_{I_{t+1}}). \tag{3.10}$$

By Lemma 9 and the σ-semi-metric assumption, we also know that

$$F(\frac{1}{2}\mathbb{1}_{I'_t} + \frac{1}{2}\mathbb{1}_{I_{t+1}}) \leq F(\mathbb{1}_{I^*}) + \frac{1}{4}\sum_{(i,j)\in M^*} d(i,j)$$
$$\leq F(\mathbb{1}_{I^*}) + \frac{1}{4}\sum_{(i,j)\in M^*} \frac{\sigma}{r-1}\big(d(i, I'_t - i) + d(j, I'_t - i)\big). \tag{3.11}$$

Note that none of the edges of M^* is present in the right hand side summation. Therefore

$$\sum_{(i,j)\in M^*} (d(i, I'_t - i) + d(j, I'_t - i)) \leq d(I'_t) + d(I'_t, I_{t+1}) - \sum_{(i,j)\in M^*} d(i,j)$$
$$\leq 4 \cdot F(\frac{1}{2}\mathbb{1}_{I'_t} + \frac{1}{2}\mathbb{1}_{I_{t+1}}) - \sum_{(i,j)\in M^*} d(i,j) \leq 4F(\mathbb{1}_{I^*}) = 4f(I^*). \tag{3.12}$$

where the second inequality follows from Lemma 8 and the last inequality holds because of Lemma 9. Combining (3.10), (3.11), and (3.12), we get

$$\sum_{(i,j)\in M_t} d(i,j) \leq \big(4 + \frac{4\sigma}{r-1}\big) f(I^*). \tag{3.13}$$

Hence, by (3.9) and (3.13), we have

$$F(x_1) \leq f(I'_p) + \Big(2 + \frac{2\sigma}{r-1}\Big) f(I^*),$$

and this yields the result. □

In the last two sections, we provided two different rounding algorithms for discrete quadratic functions when we deal with general matroid constraints. We can run both algorithms and take the best of them. This immediately implies the following theorem.

Theorem 12 (Quadratic Integrality Gap over Matroid)**.** *Let $f \in \mathcal{O}_\sigma$ be a set function and F its multi-linear extension. Let $\mathcal{M}$ be a matroid of rank r, minimum circuit size c, and matroid polytope $P_\mathcal{M}$. Then there is a polytime algorithm which given $x^* \in P_\mathcal{M}$ produces an integral vector $\mathbb{1}_I \in P_\mathcal{M}$ such that $F(x^*) \leq O(\min\{\frac{r}{c-2}, 1 + \frac{\sigma}{r}\}) f(I) \leq O(\sqrt{\sigma}) f(I)$.*

These rounding algorithms can be combined with the jump-start continuous greedy (Theorem 2) to conclude the following.

Theorem 13. *There is an $O(\sigma^{3/2})$-approximation algorithm for maximizing $f \in \mathcal{O}_\sigma$ over a matroid.*

One immediate question is whether we can do better than the provided bounds. In the next two sections, we show that our bounds are almost tight for both phases of the algorithm: the jump-start continuous greedy and the rounding.

3.5 Integrality Gap Lower Bound for Discrete Quadratics

In this section, we describe an example that shows the integrality gap of a quadratic function with a σ-semi-metric distance over a matroid polytope is $\Omega(\min\{\frac{r}{c-2}, \frac{\sigma}{r}\})$ in the worst case, where r is the rank of the matroid and c is the size of the smallest circuit.

Proposition 8. *Let $k, t \in \mathbb{N}$ with $1 \leq t \leq k$. There exists a σ-semi-metric diversity function with multilinear extension F, and a matroid $\mathcal{M} = ([2k], \mathcal{I})$ with rank $r = k + t - 1$ and minimum circuit size $c = 2t$, where the integrality gap of $F(x)$ over the matroid polytope $P_{\mathcal{M}}$ is $\Omega(\min\{\frac{r}{c-2}, \frac{\sigma}{r}\})$.*

Proof. Let $S_i = \{2i-1, 2i\}$ for $1 \leq i \leq k$, and $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$. We define a matroid $\mathcal{M} = ([2k], \mathcal{I})$ in terms of its circuits as follows. A set C is a circuit of $\mathcal{M}$ if and only if C is the union of any t sets S_i. It is then clear that the minimum size c of a circuit is $2t$, and the rank r of the matroid is $k + t - 1$. For example, $\mathcal{M}$ could be the graphic matroid corresponding to the graph in Figure 3.1. Circuits here correspond to cycles of size 4, and the dashed lines show the coefficients of F that are equal to one.

Let

$$F(x) = \sum_{\{u,v\} \in \mathcal{S}} x_u x_v + \sum_{\{u,v\} \in ([2k] \times [2k]) \setminus \mathcal{S}} \frac{1}{\sigma} x_u x_v$$

. It is straightforward to see that F is the multilinear extension of a σ-semi-metric induced by a complete graph which has weight 1 on edges from $\mathcal{S}$ and weight $1/\sigma$ otherwise.

By definition of $\mathcal{M}$ and F, it is clear that any integral solution $x_I \in P_{\mathcal{M}}$ maximizing F will pick $t-1$ pairs from $\mathcal{S}$ and then singletons from other pairs. Therefore

$$\begin{aligned} F(x_I) &:= \max_{x \in P_{\mathcal{M}} \cap \{0,1\}^{2k}} F(x) = (t-1) + \frac{1}{\sigma}\Big(\binom{r}{2} - (t-1)\Big) = (1 - \frac{1}{\sigma})(t-1) + \frac{1}{\sigma}\binom{r}{2} \\ &= \frac{(\sigma-1)(c-2) + r(r-1)}{2\sigma}. \end{aligned}$$

On the other hand, $x_0 = \frac{k+t-1}{2k} \mathbb{1}_{[2k]} \in P_{\mathcal{M}}$ and

$$F(x_0) = k(\frac{k+t-1}{2k})^2 + \Big(\binom{2k}{2} - k\Big)\frac{1}{\sigma}(\frac{k+t-1}{2k})^2 = k\big(\frac{k+t-1}{2k}\big)^2\big(1 + \frac{2(k-1)}{\sigma}\big).$$

Using that $r = k + t - 1$ and $k = r - \frac{c}{2} + 1$ we have

$$k(\frac{k+t-1}{2k})^2 = \frac{r^2}{4(r - \frac{c}{2} + 1)} = \frac{r^2}{2(2r - c + 2)} \geq \frac{r}{4},$$

where the last inequality follows since $c \geq 2$. Hence, $F(x_0) \geq \frac{r}{4}(1 + \frac{2(k-1)}{\sigma})$. It follows that the integrality gap is at least

$$\frac{F(x_0)}{F(x_I)} \geq \frac{1}{2} \cdot \frac{\sigma r + 2r(k-1)}{(\sigma-1)(c-2) + r(r-1)} \geq \frac{1}{2} \cdot \frac{\sigma r}{\sigma(c-2) + r^2} \geq \frac{1}{4} \cdot \min\{\frac{r}{c-2}, \frac{\sigma}{r}\}.$$

$\square$

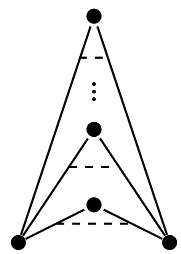


Figure 3.1: Lower bound of the integrality gap for quadratic functions.

3.6 Implications of the Rounding Results and Some Hardness Results

De Klerk [25] remarks that "approximation algorithms have been studied extensively for combinatorial optimization problems, but have not received the same attention for NP-hard continuous optimization problems". Theorem 12 implies a conditional hardness for maximizing quadratics over a simplex, as follows. Since the $O(\frac{r}{c-2})$ rounding does not depend on σ, it yields an $O(1)$ rounding for cardinality constraints. Since the multi-linear extension of the densest subgraph problem is of the form $x^T Ax$, the approximability of densest subgraph is within a constant factor of its continuous relaxation.

Corollary 3. *The continuous problem* $\max x^T Ax : x \in \Delta$ *is asymptotically as hard as the densest subgraph problem, where* Δ *is a simplex and* A *is a non-negative, symmetric matrix.*

The next result addresses the hardness of approximating the maximum of a σ-semi-metric diversity function.

Theorem 14. *Assuming the Planted Clique Conjecture: (1) for any constant* $\sigma \geq 1$*, it is hard to approximate the maximum of a* σ*-semi-metric function subject to a cardinality constraint within a factor of* $2\sigma - \epsilon$ *for any* $\epsilon > 0$ *and (2) for a super-constant* σ*, there is no constant factor (polytime) approximation algorithm for maximizing a* σ*-semi-metric function subject to a cardinality constraint.*

Proof. The Planted Clique problem asks for an algorithm to distinguish between the following graphs with probability of at least 3/4: 1) A graph drawn from $\mathcal{G}(n, 1/2)$, 2) A graph drawn from $\mathcal{G}(n, 1/2)$ and then a clique of size $n^{1/2-\delta}$ is planted in it ($\delta > 0$) [44]. The *planted clique conjecture* states that there is no polynomial time algorithm to do this task [6, 43]. It has been shown that assuming the planted clique conjecture, it is hard to approximate the maximum of a metric diversity function within a factor better than 2 [8, 11].

Given a graph G, in the *densest k-subgraph problem* we need to find an induced subgraph of size k with the maximum number of edges. Let R be a subset of vertices of G and $E(R)$ be the number of edges in the induced subgraph of R. The density of R is defined as $\rho(R) = E(R)/\binom{|R|}{2}$. Alon et al. [6] showed that if there is no polynomial time algorithm for the planted clique problem for a planted clique of size $n^{1/3}$, then there is no polynomial time algorithm for distinguishing between a graph G_1 of size n that contains a clique of size $n^{1/3}$, and a graph G_2 of the same size in which the density of every subset of vertices of size $n^{1/3}$ is at most $\delta > 0$.

We can reduce the densest k-subgraph problem to σ-semi-metric function maximization in the following way. Consider an instance of densest k-subgraph ($k = n^{1/3}$) on graph G with vertex set $[n]$. Create the distance function $d : [n] \times [n] \to \mathbb{R}$. If there is an edge between $i, j \in [n]$ in G, set $d(i,j) = 2\sigma$, otherwise set $d(i,j) = 1$. It is easy to see that this distance function is a σ-semi-metric. Let $f(R) = \sum_{\{i,j\} \subseteq R} d(i,j)$. If $|R| = k$, we have

$$f(R) = 2\sigma E(R) + (\binom{k}{2} - E(R)).$$

We know $\binom{k}{2} \geq E(R)$. Therefore

$$2\sigma E(R) \leq f(R) \leq 2\sigma E(R) + \binom{k}{2},$$

and dividing both sides by $2\sigma\binom{k}{2}$ we get

$$\rho(R) \leq \frac{f(R)}{2\sigma\binom{k}{2}} \leq \rho(R) + \frac{1}{2\sigma}. \tag{3.14}$$

It is easy to see that

$$\arg\max_{\substack{R \subseteq [n] \\ |R| = k}} \rho(R) = \arg\max_{\substack{R \subseteq [n] \\ |R| = k}} f(R).$$

Now, assume that for some fixed constant $c \geq 1$ there is a c-factor approximate algorithm for finding the maximum of σ-semi-metric function (σ is super-constant) and its output on G is S. Also, let

$$\mathtt{OPT} \in \arg\max_{\substack{R \subseteq [n] \\ |R| = k}} \rho(R).$$

We have

$$\rho(\mathtt{OPT}) \leq \frac{f(\mathtt{OPT})}{2\sigma\binom{k}{2}} \leq \frac{cf(S)}{2\sigma\binom{k}{2}} \leq c\rho(S) + \frac{c}{2\sigma}.$$

Since $\sigma \in \omega(1)$, for some n large enough we have that $\frac{c}{2\sigma} \leq \frac{1}{2}$. Hence $\rho(\mathtt{OPT}) \leq c\rho(S) + \frac{1}{2}$. Set $\delta = \frac{1}{4c}$ and note that $\delta > 0$ is a constant. If G is a graph in which the density of every subset of vertices of size k is at most δ, then clearly $\rho(S) \leq \delta$. If G is a graph that contains a clique of size k, then $1 = \rho(\mathtt{OPT}) \leq c\rho(S) + \frac{1}{2}$, which means $\rho(S) \geq \frac{1}{2c} = 2\delta$. This means that our c-factor approximate algorithm can distinguish between these two graphs which is in contrast with the planted clique conjecture.

For the first part, given any constant σ, assume there is a $(2\sigma - \epsilon)$-factor approximate algorithm for some $\epsilon > 0$ for finding the maximum of σ-semi-metric function. Denote its output on G by S, and let OPT be defined as above. We then have

$$\rho(\mathtt{OPT}) \leq \frac{f(\mathtt{OPT})}{2\sigma\binom{k}{2}} \leq \frac{(2\sigma - \epsilon)f(S)}{2\sigma\binom{k}{2}} \leq (2\sigma - \epsilon)\rho(S) + \frac{2\sigma - \epsilon}{2\sigma}.$$

Set $\delta = (\frac{1}{2\sigma-\epsilon} - \frac{1}{2\sigma})/2 = \frac{\epsilon}{4\sigma(2\sigma-\epsilon)}$, and note that $\delta > 0$ is a constant. If G is a graph in which the density of every subset of vertices of size k is at most δ then clearly $\rho(S) \leq \delta$. If G is a graph that contains a clique of size k then $1 = \rho(\texttt{OPT}) \leq (2\sigma - \epsilon)\rho(S) + \frac{2\sigma-\epsilon}{2\sigma}$ which means $\rho(S) \geq \frac{1}{2\sigma-\epsilon} - \frac{1}{2\sigma} = 2\delta$. This means that our $(2\sigma - \epsilon)$-factor approximate algorithm can distinguish between these two graphs which is in contrast with the planted clique conjecture and Alon et al. result. □

We note that the jump-start continuous greedy (Theorem 2) followed by the rounding (Theorem 9) gives an $O(\sigma)$-approximation in the case of uniform matroids. This approximation is asymptotically tight because the planted clique hardness result (Theorem 14) shows that we cannot expect an approximation better than 2σ in uniform matroids. Moreover, the $O(1)$ rounding for uniform matroids implies that the approximation factor of the discrete problem and the continuous problem are effectively (asymptotically) the same. Hence it is hard to approximate the continuous problem within a factor of $o(\sigma)$.

Corollary 4. *Let A be a matrix corresponding to a σ-semi-metric distance function. Then it is hard to approximate the continuous problem $\max x^T Ax : ||x||_1 \leq k$ within a factor of $o(\sigma)$. Moreover this implies that the analysis of the jump-start continuous greedy algorithm in Theorem 2 is asymptotically tight.*

3.7 Future Work

In this chapter, we discussed a rounding algorithm for a subclass of functions that have non-positive third-order derivatives, in the case of uniform matroid. Hence one question is to extend this to general matroids. We also discussed rounding algorithms for a subclass of functions that have zero third-order derivatives (discrete quadratics) in case of general matroids. An important open question is to generalize this to set functions with one-sided σ-smooth multilinear extensions without any additional condition on the third-order derivatives.

Chapter 4

Meta-Submodular Functions

In this section we study more general monotone set functions than the ones that have one-sided smooth multi-linear extensions. To motivate our approach we consider the definition of one-sided σ-smoothness restricted to only integral points of a function F instead of its whole domain. Namely, for any non-empty $S \subseteq [n]$: $u^T \nabla^2 F(\mathbb{1}_S) u \leq \sigma \cdot \frac{||u||_1}{|S|} u^T \nabla F(\mathbb{1}_S)$. If we also limit our attention to directions $u = \boldsymbol{e}_i + \boldsymbol{e}_j$, the inequality becomes

$$\nabla^2_{ij} F(\mathbb{1}_S) \leq \sigma \cdot (\frac{\nabla_i F(\mathbb{1}_S) + \nabla_j F(\mathbb{1}_S)}{|S|}). \tag{4.1}$$

Now suppose that F is the multi-linear extension of a set function $f : 2^{[n]} \to \mathbb{R}_{\geq 0}$, and so $F(\mathbb{1}_S) = f(S)$. One may show [81] that $\nabla_i F(\mathbb{1}_S) = f(S+i) - f(S-i)$ and $\nabla^2_{ij} F(\mathbb{1}_S) = f(S+i+j) - f(S+i-j) - f(S-i+j) + f(S-i-j) = \nabla_j F(\mathbb{1}_{S+i}) - \nabla_j F(\mathbb{1}_{S-i})$. To abbreviate notation we write $B_i(S) = \nabla_i F(\mathbb{1}_S)$ and $A_{ij}(S) = \nabla^2_{ij} F(\mathbb{1}_S)$ and so (4.1) becomes:

$$A_{ij}(S) \leq \sigma \cdot (\frac{B_i(S) + B_j(S)}{|S|}). \tag{4.2}$$

We call a set function f *σ-meta-submodular* if it satisfies this inequality for any $S \neq \emptyset$. One may view this as the discrete analogue of bounding the second-order term of a Taylor series by the corresponding first-order term. We primarily focus on monotone functions and so we denote by $\mathcal{G}_\sigma$ the family of non-negative, monotone set functions which are σ-meta-submodular. Note that since the B_i's are non-negative, we then have that $\mathcal{G}_\sigma \subseteq \mathcal{G}_{\sigma'}$ if $\sigma < \sigma'$.

We first discuss the structure around the meta-submodular family (see Fig. 4.1). Most importantly with respect to (Q2) is that $\mathcal{G}_\sigma$ includes all monotone submodular functions and σ-semi-metric diversity functions. More precisely, the 0-meta-submodular functions coincide with the class of meta-submodular functions defined by Kleinberg et al [48], which properly includes all submodular functions.

Proposition 9. *f is 0-meta-submodular if and only if it is meta-submodular (by Kleinberg et al. definition [48]).*

Proof. Kleinberg et al [48] show that a set function f is meta-submodular if and only if

$$f(S+i) - f(S) \geq f(T+i) - f(T), \ \ \forall \emptyset \neq S \subseteq T, \ \forall i \notin T.$$

The above is clearly equivalent to

$$f(S+i) - f(S) \geq f(S+j+i) - f(S+j), \ \ \forall S \neq \emptyset, \ \forall i \neq j \notin S. \tag{4.3}$$

Then

$$
\begin{aligned}
&f \text{ is 0-meta submodular}\\
\iff& A_{ij}(S) \le 0,\ \ \forall S \neq \emptyset,\ \forall i,j \in V\\
\iff& f(S+i+j) - f(S+i) - f(S+j) + f(S) \le 0,\ \ \forall S \neq \emptyset,\ \forall i,j \in V\\
\iff& f(S+i) - f(S) \ge f(S+j+i) - f(S+j),\ \ \forall S \neq \emptyset,\ \forall i,j \in V\\
\iff& f(S+i) - f(S) \ge f(S+j+i) - f(S+j),\ \ \forall S \neq \emptyset,\ \forall i \neq j \notin S\\
\iff& (4.3) \text{ holds.}
\end{aligned}
$$

□

Proposition 10. *Any second-order-modular function (Definition 4) with a σ-semi-metric distance function ($\sigma \ge 1$) and a non-negative modular function is a σ-meta submodular function.*

Proof. Let $f(R) = \sum_{q\in R} g(q) + \sum_{\{q,q'\}\subseteq R} d(q,q')$ be a second-order modular function (by Lemma 7, it has this form). The proof is by case analysis.

- If $i, j \notin R$, we have

$$
\begin{aligned}
|R|A_{ij}(R) &= |R|(f(R+i+j) - f(R+i-j) - f(R-i+j) + f(R-i-j))\\
&= |R|(\sum_{q\in R+i+j} g(q) + \sum_{\{q,q'\}\subseteq R+i+j} d(q,q') - \sum_{q\in R+i} g(q) - \sum_{\{q,q'\}\subseteq R+i} d(q,q')\\
&- \sum_{q\in R+j} g(q) - \sum_{\{q,q'\}\subseteq R+j} d(q,q') + \sum_{q\in R} g(q) + \sum_{\{q,q'\}\subseteq R} d(q,q'))\\
&= |R|d(i,j).
\end{aligned}
$$

  We also have

$$
\begin{aligned}
\sigma(B_i(R) + B_j(R)) &= \sigma(f(R+i) - f(R-i) + f(R+j) - f(R-i))\\
&= \sigma(\sum_{q\in R+i} g(q) + \sum_{\{q,q'\}\subseteq R+i} d(q,q') - \sum_{q\in R} g(q) - \sum_{\{q,q'\}\subseteq R} d(q,q')\\
&+ \sum_{q\in R+j} g(q) + \sum_{\{q,q'\}\subseteq R+j} d(q,q') - \sum_{q\in R} g(q) - \sum_{\{q,q'\}\subseteq R} d(q,q'))\\
&= \sigma g(i) + \sigma g(j) + \sigma \sum_{q\in R} d(i,q) + \sigma \sum_{q\in R} d(j,q).
\end{aligned}
$$

  Therefore $|R|A_{ij}(R) \le \sigma(B_i(R) + B_j(R))$ because g is non-negative and d is non-negative σ-semi-metric.

- If $i, j \in R$, we have

$$\begin{aligned}
|R|A_{ij}(R) &= |R|(f(R+i+j) - f(R+i-j) - f(R-i+j) + f(R-i-j)) \\
&= |R|(\sum_{q\in R} g(q) + \sum_{\{q,q'\}\subseteq R} d(q,q') - \sum_{q\in R-j} g(q) - \sum_{\{q,q'\}\subseteq R-j} d(q,q') \\
&- \sum_{q\in R-i} g(q) - \sum_{\{q,q'\}\subseteq R-i} d(q,q') + \sum_{q\in R-i-j} g(q) + \sum_{\{q,q'\}\subseteq R-i-j} d(q,q')) \\
&= |R|d(i,j).
\end{aligned}$$

We also have

$$\begin{aligned}
\sigma(B_i(R)+B_j(R)) &= \sigma(f(R+i) - f(R-i) + f(R+j) - f(R-i)) \\
&= \sigma(\sum_{q\in R} g(q) + \sum_{\{q,q'\}\subseteq R} d(q,q') - \sum_{q\in R-i} g(q) - \sum_{\{q,q'\}\subseteq R-i} d(q,q') \\
&+ \sum_{q\in R} g(q) + \sum_{\{q,q'\}\subseteq R} d(q,q') - \sum_{q\in R-j} g(q) - \sum_{\{q,q'\}\subseteq R-j} d(q,q')) \\
&= \sigma g(i) + \sigma g(j) + 2\sigma d(i,j) + \sigma \sum_{q\in R-i-j} d(i,q) + \sigma \sum_{q\in R-i-j} d(j,q).
\end{aligned}$$

Therefore $|R|A_{ij}(R) \leq \sigma(B_i(R)+B_j(R))$ because g is non-negative, d is non-negative σ-semi-metric, and $\sigma \geq 1$.

- If $i \in R$ and $j \notin R$, we have

$$\begin{aligned}
|R|A_{ij}(R) &= |R|(f(R+i+j) - f(R+i-j) - f(R-i+j) + f(R-i-j)) \\
&= |R|(\sum_{q\in R+j} g(q) + \sum_{\{q,q'\}\subseteq R+j} d(q,q') - \sum_{q\in R} g(q) - \sum_{\{q,q'\}\subseteq R} d(q,q') \\
&- \sum_{q\in R-i+j} g(q) - \sum_{\{q,q'\}\subseteq R-i+j} d(q,q') + \sum_{q\in R-i} g(q) + \sum_{\{q,q'\}\subseteq R-i} d(q,q')) \\
&= |R|d(i,j).
\end{aligned}$$

We also have

$$\begin{aligned}
\sigma(B_i(R)+B_j(R)) &= \sigma(f(R+i) - f(R-i) + f(R+j) - f(R-i)) \\
&= \sigma(\sum_{q\in R} g(q) + \sum_{\{q,q'\}\subseteq R} d(q,q') - \sum_{q\in R-i} g(q) - \sum_{\{q,q'\}\subseteq R-i} d(q,q') \\
&+ \sum_{q\in R+j} g(q) + \sum_{\{q,q'\}\subseteq R+j} d(q,q') - \sum_{q\in R} g(q) - \sum_{\{q,q'\}\subseteq R} d(q,q')) \\
&= \sigma g(i) + \sigma g(j) + \sigma d(i,j) + \sigma \sum_{q\in R-i} d(i,q) + \sigma \sum_{q\in R-i} d(j,q).
\end{aligned}$$

Therefore $|R|A_{ij}(R) \leq \sigma(B_i(R)+B_j(R))$ because g is non-negative, d is non-negative σ-semi-metric, and $\sigma \geq 1$.

□

Another property is that every proportional submodular function (cf. Borodin et al [13]) is 1-meta-submodular.

Proposition 11. *Any monotone propotionally submodular function is* 1*-meta-submodular.*

Proof. The proof is by case analysis.

- If $i, j \notin R$ then using weak submodularity property we have

$$(|R|+2)f(R) + (|R|)f(R+i+j) \leq (|R|+1)f(R+i) + (|R|+1)f(R+j),$$

which means

$$|R| \cdot (f(R) + f(R+i+j) - f(R+i) - f(R+j)) \leq f(R+i) + f(R+j) - 2f(R).$$

Hence

$$\begin{aligned}
&f(R+i+j) - f(R+i-j) - f(R+j-i) + f(R-i-j) \\
&= f(R+i+j) - f(R+i) - f(R+j) + f(R) \\
&\leq \frac{f(R+i) - f(R) + f(R+j) - f(R)}{|R|} \\
&= \frac{f(R+i) - f(R-i) + f(R+j) - f(R-j)}{|R|}.
\end{aligned}$$

- If $i, j \in R$ then using weak submodularity property we have

$$(|R|-2)f(R) + (|R|)f(R-i-j) \leq (|R|-1)f(R-i) + (|R|-1)f(R-j),$$

which means

$$|R| \cdot (f(R) + f(R-i-j) - f(R-i) - f(R-j)) \leq 2f(R) - f(R-i) - f(R-j).$$

Hence

$$\begin{aligned}
&f(R+i+j) - f(R+i-j) - f(R+j-i) + f(R-i-j) \\
&= f(R) - f(R-j) - f(R-i) + f(R-i-j) \\
&\leq \frac{f(R) - f(R-i) + f(R) - f(R-j)}{|R|} \\
&= \frac{f(R+i) - f(R-i) + f(R+j) - f(R-j)}{|R|}.
\end{aligned}$$

- If $i \in R$ and $j \notin R$ then using weak submodularity property we have

$$(|R|-1)f(R+j) + (|R|+1)f(R-i) \leq (|R|)f(R) + (|R|)f(R+j-i),$$

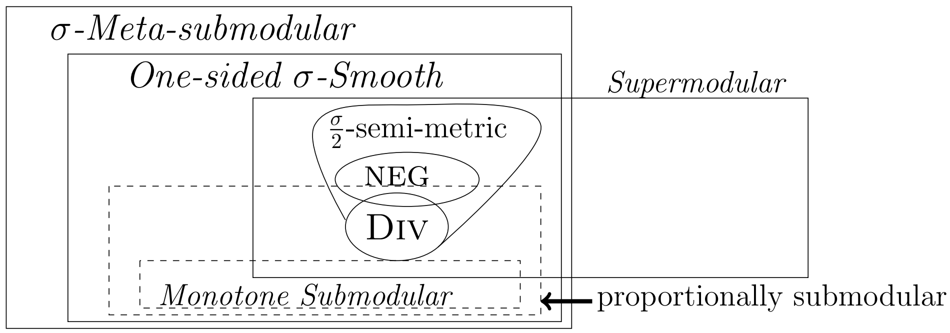


Figure 4.1: The meta-submodular families

which means

$$\begin{aligned}&|R| \cdot (f(R+j) + f(R-i) - f(R) - f(R+j-i)) \le f(R+j) - f(R-i)\\&= f(R+j) - f(R-j) + f(R+i) - f(R-i),\end{aligned}$$

where the equality is correct because $f(R) = f(R-j) = f(R+i)$. Hence

$$\begin{aligned}&f(R+i+j) - f(R+i-j) - f(R+j-i) + f(R-i-j)\\&= f(R+j) - f(R) - f(R+j-i) + f(R-i)\\&\le \frac{f(R+j) - f(R-i)}{|R|}\\&= \frac{f(R+i) - f(R-i) + f(R+j) - f(R-j)}{|R|}.\end{aligned}$$

□

Given the performance guarantees of continuous greedy for smooth functions, it is natural to study the smoothness of multi-linear extensions from the meta-submodular families. First, one can show that if the multi-linear extension of a set function is one-sided σ-smooth, then the set function itself is σ-meta-submodular.

Proposition 12. *Let f be a set function and F be its multi-linear extension. If F is one-sided γ-smooth, then f is γ-meta-submodular.*

Proof. Let non-empty $R \subseteq [n]$ and $i, j \in [n]$. Consider the inequality of one-sided γ-smoothness for $u = \mathbb{1}_{\{i,j\}}$ and $x = \mathbb{1}_R$:

$$2u_i u_j \nabla^2 F_{ij}(x) \le \gamma \frac{u_i + u_j}{||x||_1}(u_i \nabla_i F(x) + u_j \nabla_j F(x))$$

Since $u_i = u_j = 1$, $||x||_1 = |R|$, $\nabla^2 F_{ij}(x) = A_{ij}(R), and \nabla_i F(x) + \nabla_j F(x) = B_i(S) + B_j(S)$ we obtain the γ-meta-submodular inequality. □

The converse is not necessarily true however: the multi-linear extension of a σ-meta-submodular function is not always one-sided σ-smooth. Hence, we prefer to use a different parameter γ when referring to meta-submodularity. In other words we speak of γ-meta-submodular set functions and write $\mathcal{G}_\gamma$. One may think of γ as a discrete smoothness parameter. In the next section, we investigate the smoothness of γ-meta-submodular functions.

4.1 One-Sided Smoothness of γ-Meta-Submodular Functions

The following result shows that a set function's multi-linear extension is one-sided smooth whenever a stronger probabilistic version of (4.2) is satisfied. We call this the *expectation inequality* (4.4), where $R \sim x$ denotes a random set that contains element i independently with probability x_i.

Lemma 10 (Expectation Inequality)**.** *Let f be a non-negative, monotone set function and F be its multi-linear function. Let $x \in [0,1]^n$ and $\gamma \geq 0$. If for any $i, j \in [n]$ we have*

$$\mathbb{E}_{R\sim x}[|R|] \cdot \mathbb{E}_{R\sim x}[A_{ij}(R)] \leq \sigma \cdot (\mathbb{E}_{R\sim x}[B_i(R)] + \mathbb{E}_{R\sim x}[B_j(R)]), \tag{4.4}$$

or equivalently (by Lemma 1),

$$||x||_1 \nabla^2_{ij} F(x) \leq \gamma(\nabla_i F(x) + \nabla_j F(x)),$$

then F is one-sided 2γ-smooth at x.

Proof. We have

$$\begin{aligned}
u^T \nabla^2 F(x) u &= \sum_{i=1}^n \sum_{j=1}^n u_i u_j \nabla^2_{ij} F(x) \leq \frac{\gamma}{||x||_1} \sum_{i=1}^n \sum_{j=1}^n u_i u_j (\nabla_i F(x) + \nabla_j F(x)) \\
&= \frac{\gamma}{||x||_1} (\sum_{i=1}^n \sum_{j=1}^n u_i u_j \nabla_i F(x) + \sum_{i=1}^n \sum_{j=1}^n u_i u_j \nabla_j F(x)) \\
&= \frac{\gamma}{||x||_1} (\sum_{i=1}^n u_i \nabla_i F(x) (\sum_{j=1}^n u_j) + \sum_{i=1}^n u_i (\sum_{j=1}^n u_j \nabla_j F(x))) \\
&= \frac{\gamma}{||x||_1} (||u||_1 \sum_{i=1}^n u_i \nabla_i F(x) + ||u||_1 \sum_{j=1}^n u_j \nabla_j F(x)) \\
&= 2\gamma \left(\frac{||u||_1}{||x||_1} \right) (u^T \nabla F(x)).
\end{aligned}$$

□

We have proved that this inequality holds (modulo a constant factor) in the supermodular case, i.e., for the intersection of supermodular functions and γ-meta-submodular functions. To prove it, we need the next lemma.

Lemma 11. *Let $f : 2^{[n]} \to \mathbb{R}_+$ be a non-negative, monotone, supermodular, γ-meta-submodular set function. Let $x \in [0,1]^n \setminus \{\vec{0}\}$ and $R \subseteq [n]$ such that $1 \leq |R| < ||x||_1$. Then for all $i, j \in [n]$ we have*

$$(||x||_1 - |R|)A_{ij}(R)p_x(R) \leq 2\gamma \sum_{e\in[n]\setminus R} (\frac{B_i(R+e) + B_j(R+e)}{|R|+1})p_x(R+e).$$

Also, for the empty set,

$$(||x||_1)A_{ij}(\emptyset)p_x(\emptyset) \leq 2(\gamma+1) \sum_{e\in[n]} (B_i(\{e\}) + B_j(\{e\}))p_x(\{e\}).$$

Proof. Let $|R| = r$. Note that $r < n$ because $|R| = r < ||x||_1$. Also, note that if $x_e = 1$ for some $e \in [n] \setminus R$ then $p_x(R) = 0$, which means that the left hand side is zero. In that case, the inequality holds because f is monotone and the right hand side is non-negative. Hence, we assume that $x_e < 1$ for all $e \in [n] \setminus R$. We know that

$$\sum_{e\in[n]} x_e = ||x||_1.$$

Therefore, because each $x_e \leq 1$,

$$\sum_{e\in[n]\setminus R} x_e = ||x||_1 - \sum_{e\in R} x_e \geq ||x||_1 - \sum_{e\in R} 1 = ||x||_1 - |R|.$$

Hence, since $0 < 1 - x_e \leq 1$ for all $e \in [n] \setminus R$, we get

$$\begin{aligned}(||x||_1 - |R|)A_{ij}(R)p_x(R) &\leq \sum_{e\in[n]\setminus R} x_e A_{ij}(R)p_x(R) \\ &\leq \sum_{e\in[n]\setminus R} \frac{x_e}{1-x_e} A_{ij}(R)p_x(R) \\ &= \sum_{e\in[n]\setminus R} A_{ij}(R)p_x(R+e).\end{aligned}$$

Moreover, $2|R| \geq |R| + 1$ because $|R| \geq 1$, and we have

$$\sum_{e\in[n]\setminus R} A_{ij}(R)p_x(R+e) \leq 2 \sum_{e\in[n]\setminus R} \frac{|R|A_{ij}(R)}{|R|+1} p_x(R+e).$$

Using the γ-meta-submodularity and supermodularity we have

$$\begin{aligned}2 \sum_{e\in[n]\setminus R} \frac{|R|A_{ij}(R)}{|R|+1} p_x(R+e) &\leq 2\gamma \sum_{e\in[n]\setminus R} \frac{B_i(R) + B_j(R)}{|R|+1} p_x(R+e) \\ &\leq 2\gamma \sum_{e\in[n]\setminus R} \frac{B_i(R+e) + B_j(R+e)}{|R|+1} p_x(R+e)\end{aligned}$$

Combining all of these inequalities yields the first part of the lemma. For the second part of the lemma, we consider the set $\{i, j, e\}$. By Lemma 3 and the γ-meta-submodularity, we have

$$\begin{aligned} f(\{i,j,e\}) &= B_i(\{j,e\}) + B_j(\{e\}) + f(\{e\}) \\ &= A_{ij}(\{e\}) + B_i(\{e\}) + B_j(\{e\}) + f(\{e\}) \\ &\leq (\gamma+1)(B_i(\{e\}) + B_j(\{e\})) + f(\{e\}). \end{aligned}$$

Also, by Lemma 3, we have

$$\begin{aligned} f(\{i,j,e\}) &= B_i(\{j,e\}) + B_j(\{e\}) + f(\{e\}) \\ &= A_{ie}(\{j\}) + A_{ij}(\emptyset) + f(\{i\}) + B_j(\{e\}) + f(\{e\}). \end{aligned}$$

Therefore

$$A_{ie}(\{j\}) + A_{ij}(\emptyset) + f(\{i\}) + B_j(\{e\}) + f(\{e\}) \leq (\gamma+1)(B_i(\{e\}) + B_j(\{e\})) + f(\{e\}).$$

Hence, because f is non-negative, monotone and supermodular, it follows that

$$A_{ij}(\emptyset) \leq A_{ie}(\{j\}) + A_{ij}(\emptyset) + f(\{i\}) + B_j(\{e\}) \leq (\gamma+1)(B_i(\{e\}) + B_j(\{e\})). \tag{4.5}$$

Moreover, because f is non-negative and monotone, we have

$$\begin{aligned} A_{ij}(\emptyset) &= f(\{i,j\}) - f(\{i\}) - f(\{j\}) + f(\emptyset) = B_j(\{i\}) - f(\{j\}) \\ &\leq B_j(\{i\}) + B_i(\{i\}) \leq (\gamma+1)(B_j(\{i\}) + B_i(\{i\})), \end{aligned}$$

and

$$\begin{aligned} A_{ij}(\emptyset) &= f(\{i,j\}) - f(\{i\}) - f(\{j\}) + f(\emptyset) = B_i(\{j\}) - f(\{i\}) \\ &\leq B_i(\{j\}) + B_j(\{j\}) \leq (\gamma+1)(B_i(\{j\}) + B_j(\{j\})). \end{aligned}$$

If $x_e = 1$ for an $e \in [n]$ then $p_x(\emptyset) = 0$ and the inequality holds because the left hand side is zero and the right hand side is non-negative (since f is monotone). Therefore, we assume that $x_e < 1$ for all $e \in [n]$. Combining the above inequalities, we have

$$\begin{aligned} (||x||_1) A_{ij}(\emptyset) p_x(\emptyset) &= \sum_{e \in [n]} x_e A_{ij}(\emptyset) p_x(\emptyset) \\ &\leq \sum_{e \in [n]} \frac{x_e}{1-x_e} A_{ij}(\emptyset) p_x(\emptyset) \\ &= \sum_{e \in [n]} A_{ij}(\emptyset) p_x(\{e\}) \\ &\leq (\gamma+1) \sum_{e \in [n]} (B_i(\{e\}) + B_j(\{e\})) p_x(\{e\}), \end{aligned}$$

where the last inequality follows from (4.5). This completes the proof.

□

Now using this, we conclude the following.

Lemma 12. *Let f be a non-negative, monotone, supermodular, γ-meta-submodular set function and F be its multi-linear function. Then for any $x \in [0,1]^n \setminus \{\vec{0}\}$ and $i, j \in [n]$,*

$$||x||_1 \nabla^2_{ij} F(x) \leq (\max\{3\gamma, 2\gamma+1\})(\nabla_i F(x) + \nabla_j F(x)).$$

Proof. By using Lemma 11 for all the sets of size less than $||x||_1$, we can write

$$\begin{aligned}
&(||x||_1) A_{ij}(\emptyset) p_x(\emptyset) + \sum_{\substack{R \subseteq [n] \\ 1 \leq |R| < ||x||_1}} (||x||_1 - |R|) A_{ij}(R) p_x(R) \\
&\leq (\gamma+1) \sum_{e \in [n]} (B_i(\{e\}) + B_j(\{e\})) p_x(\{e\}) \\
&+ 2\gamma \sum_{\substack{R \subseteq [n] \\ 1 \leq |R| < ||x||_1}} \sum_{e \in [n] \setminus R} (\frac{B_i(R+e) + B_j(R+e)}{|R|+1}) p_x(R+e) \\
&= (\gamma+1) \sum_{e \in [n]} (B_i(\{e\}) + B_j(\{e\})) p_x(\{e\}) + 2\gamma \sum_{\substack{R \subseteq [n] \\ 2 \leq |R| < ||x||_1 + 1}} (B_i(R) + B_j(R)) p_x(R) \\
&\leq \max\{\gamma+1, 2\gamma\} \sum_{R \subseteq [n]} (B_i(R) + B_j(R)) p_x(R) \\
&= \max\{\gamma+1, 2\gamma\} (\nabla_i F(x) + \nabla_j F(x)),
\end{aligned} \tag{4.6}$$

where the equality follows from a simple counting argument, and in the last inequality we used the monotonicity of f (i.e., the B_i's are non-negative).
By γ-meta-submodularity, we also have that

$$\begin{aligned}
&\sum_{\substack{R \subseteq [n] \\ 1 \leq |R| < ||x||_1}} |R| A_{ij}(R) p_x(R) + \sum_{\substack{R \subseteq [n] \\ |R| \geq ||x||_1}} (||x||_1) A_{ij}(R) p_x(R) \\
&\leq \sum_{|R| \geq 1} |R| A_{ij}(R) p_x(R) \leq \sum_{|R| \geq 1} \gamma (B_i(R) + B_j(R)) p_x(R) \\
&\leq \sum_{R \subseteq [n]} \gamma (B_i(R) + B_j(R)) p_x(R) = \gamma (\nabla_i F(x) + \nabla_j F(x)).
\end{aligned} \tag{4.7}$$

By adding (4.6) and (4.7), we conclude that

$$||x||_1 \sum_{R \subseteq [n]} A_{ij}(R) p_x(R) = ||x||_1 \nabla^2_{ij} F(x) \leq \max\{2\gamma+1, 3\gamma\} (\nabla_i F(x) + \nabla_j F(x)).$$

$\square$

Th following theorem is a direct consequence of Lemma 10 and 12.

Theorem 15. *Let f be a supermodular function in $\mathcal{G}_\gamma$ and F be its multi-linear extension. Then F is one-sided* $(\max\{6\gamma, 4\gamma+2\})$*-smooth.*

We conjecture that this also holds even without the supermodular condition.

Conjecture 1. *Let $f \in \mathcal{G}_\gamma$ and F be its multi-linear extension where $\gamma > 0$. Then F is one-sided $O(\gamma)$-smooth.*

While we do not have the continuous greedy available to us for the general family $\mathcal{G}_\gamma$, ironically one may use a weakened smoothness property to analyze a local search algorithm for the discrete problem $\max f(S) : S \in \mathcal{M}$. The weakened property asks for f to be one-sided smooth on a subdomain which dominates some integral point $\mathbb{1}_S$.

Theorem 16. *Let $f \in \mathcal{G}_\gamma$ and F be its multi-linear extension. Let $\alpha \geq 1$ and $S \subseteq [n]$ be non-empty. Then F is one-sided $2\alpha\gamma$-smooth on $\{x | x \geq \mathbb{1}_S \,,\, ||x||_1 \leq \alpha|S|\}$.*

Proof. Let $y \in \{x | x \geq \mathbb{1}_S \,,\, ||x||_1 \leq c|S|\}$. First, we show that

$$||y||_1 \nabla^2_{ij} F(y) \leq \gamma c(\nabla_i F(y) + \nabla_j F(y)).$$

We know $\nabla^2_{ij} F(y) = \sum_{R \subseteq [n]} A_{ij}(R) p_y(R)$. Since $y \geq \mathbb{1}_S$, $p_y(R) = 0$ for any R that is not a superset of S. Therefore, $\nabla^2_{ij} F(y) = \sum_{R \subseteq [n] \setminus S} A_{ij}(S \cup R) p_y(S \cup R)$. We have

$$\begin{aligned}
||y||_1 \nabla^2_{ij} F(y) = ||y||_1 \sum_{R \subseteq [n] \setminus S} A_{ij}(S \cup R) p_y(S \cup R) &\leq c|S| \sum_{R \subseteq [n] \setminus S} A_{ij}(S \cup R) p_y(S \cup R) \\
&\leq \sum_{R \subseteq [n] \setminus S} \frac{\gamma c |S|}{|S \cup R|} (B_i(S \cup R) + B_j(S \cup R)) p_y(S \cup R) \\
&\leq \sum_{R \subseteq [n] \setminus S} \gamma c (B_i(S \cup R) + B_j(S \cup R)) p_y(S \cup R) \\
&\leq \gamma c(\nabla_i F(y) + \nabla_j F(y)).
\end{aligned}$$

Now, by Lemma 10, we conclude that F is one-sided $(2c\gamma)$-smooth at y. $\square$

This sub-domain smoothness property is used in a technical analysis to analyze a local search algorithm for maximizing γ-meta-submodular functions subject to matroid constraints. We discuss this in the next section.

4.2 Local Search for Maximizing Meta-Submodular Functions under Matroid Constraints

In this section, we give a very general answer to question (Q2), and for constant values of γ we obtain a new tractable parameterized class of functions. We show that a local search algorithm can be used for maximizing γ-meta-submodular functions subject to matroid constraints. We first need the following lemmas.

Lemma 13. *Let f be a non-negative, monotone, γ-meta-submodular function, F be its multi-linear function, $R \subset [n]$, and $x \in [0,1]^n$ such that $||x||_1 \leq |R|$. Let $u = \mathbb{1}_R \vee x - \mathbb{1}_R$. Then for $0 \leq \epsilon \leq 1$*

$$u^T \nabla F(\mathbb{1}_R + \epsilon u) \leq 2^{4\gamma} u^T \nabla F(\mathbb{1}_R)$$

Proof. By Theorem 16, we know that F is one-sided 4γ-smooth on $A = \{y | y \geq \mathbb{1}_R, ||y||_1 \leq 2|R|\}$. Therefore F is one-sided 4γ-smooth on $B = \{y | \mathbb{1}_R + \epsilon u \geq y \geq \mathbb{1}_R\}$ because $B \subseteq A$. Therefore, the desired result yields by Lemma 4. □

Lemma 14. *Let f be a non-negative, monotone, γ-meta submodular function and F be its multi-linear extension. Let $R \subseteq [n]$ such that $|R| \geq 2$. Then*

$$\mathbb{1}_R^T \nabla F(\mathbb{1}_R) = \sum_{i \in R} B_i(R - i) \leq (2(\frac{\lfloor \frac{|R|}{2} \rfloor^2 + \lceil \frac{|R|}{2} \rceil^2}{\lfloor \frac{|R|}{2} \rfloor \lceil \frac{|R|}{2} \rceil} + 2)\gamma + 2) f(R) \leq (9\gamma + 2) f(R)$$

Proof. Partition R into two sets of size $\lfloor \frac{|R|}{2} \rfloor$ and of size $\lceil \frac{|R|}{2} \rceil$ like S and T. Using Theorem 16, we know that F is one-sided $(2(\lfloor \frac{|R|}{2} \rfloor / \lceil \frac{|R|}{2} \rceil + 1)\gamma)$-smooth on $\{y | \mathbb{1}_T \leq y \leq \mathbb{1}_R\}$ and it is one-sided $(2(\lceil \frac{|R|}{2} \rceil / \lfloor \frac{|R|}{2} \rfloor + 1)\gamma)$-smooth on $\{y | \mathbb{1}_S \leq y \leq \mathbb{1}_R\}$. Let $c = 2(\lceil \frac{|R|}{2} \rceil / \lfloor \frac{|R|}{2} \rfloor + 1)\gamma$. We show that

$$\sum_{i \in T} B_i(R - i) \leq c f(R).$$

Let $h(t) = F(\mathbb{1}_S + t\mathbb{1}_T)$ and $g(t) = \mathbb{1}_T^T \nabla F(\mathbb{1}_S + t\mathbb{1}_T)$ where $0 \leq t \leq 1$. Note that $g(t) = h'(t)$ and $\mathbb{1}_T^T \nabla^2 F(\mathbb{1}_S + t\mathbb{1}_T)\mathbb{1}_T = g'(t)$. Since F is one-sided c-smooth at any given point $\mathbb{1}_S \leq y \leq \mathbb{1}_R$, we have

$$g'(t) = \mathbb{1}_T^T \nabla^2 F(\mathbb{1}_S + t\mathbb{1}_T)\mathbb{1}_T \leq c(\frac{||\mathbb{1}_T||_1}{||\mathbb{1}_S + t\mathbb{1}_T||_1})(\mathbb{1}_T^T \nabla F(\mathbb{1}_S + t\mathbb{1}_T)) \leq c\frac{1}{t} g(t).$$

Therefore, $tg'(t) \leq cg(t)$. Integrating both sides, we get

$$\int_0^1 tg'(t)dt \leq \int_0^1 cg(t)dt.$$

Applying the integration by parts formula to the left hand side, we get

$$tg(t)\Big|_0^1 - \int_0^1 g(t)dt \leq c\int_0^1 g(t)dt.$$

It follows that

$$1 \cdot g(1) - 0 \cdot g(0) = \mathbb{1}_T^T \nabla F(\mathbb{1}_S + \mathbb{1}_T) = \mathbb{1}_T^T \nabla F(\mathbb{1}_R) = \sum_{i \in T} B_i(R - i) \leq (c+1)\int_0^1 g(t)dt.$$

By using $g(t) = h'(t)$ we have

$$\begin{aligned}\sum_{i \in T} B_i(R - i) &\leq (c+1)\int_0^1 h'(t)dt = (c+1)(h(1) - h(0)) = (c+1)(F(\mathbb{1}_S + \mathbb{1}_T) - F(\mathbb{1}_S)) \\ &\leq (c+1)F(\mathbb{1}_R) = (c+1)f(R).\end{aligned}$$

This means that

$$\sum_{i\in T} B_i(R-i) \le (2(\lceil\frac{|R|}{2}\rceil/\lfloor\frac{|R|}{2}\rfloor+1)\gamma+1)f(R).$$

With the same argument we can conclude that

$$\sum_{i\in S} B_i(R-i) \le (2(\lfloor\frac{|R|}{2}\rfloor/\lceil\frac{|R|}{2}\rceil+1)\gamma+1)f(R),$$

and combining these inequalities yields the lemma. □

We now present the main theorem of this section.

Theorem 17. *Let $f \in \mathcal{G}_\gamma$ and $\mathcal{M} = ([n], \mathcal{I})$ be a matroid of rank r. Let $A \in \mathcal{I}$ be an optimum set, i.e., $A \in \arg\max_{R\in\mathcal{I}} f(R)$, and $S \in \mathcal{I}$ be an $(1+\frac{\epsilon}{n^2})$-approximate local optima, i.e., for any i and j such that $S-i+j \in \mathcal{I}$, $(1+\frac{\epsilon}{n^2})f(S) \ge f(S-i+j)$, where $\epsilon > 0$ is a constant. Then if $\gamma = O(r)$, $f(A) \le O(\gamma 2^{4\gamma})f(S)$ and if $\gamma = \omega(r)$, $f(A) \le O(\gamma^2 2^{4\gamma})f(S)$.*

Proof. Since f is monotone, we assume that $|S| = |A| = r$. Given the exchangeability property of matroids, there is a bijective mapping ([70]) $g : S \setminus A \to A \setminus S$ such that $S - i + g(i) \in \mathcal{I}$ where $i \in S \setminus A$. Since S is a $(1+\frac{\epsilon}{n^2})$-approximate local optima, for all $i \in S \setminus A$ we have $(1+\frac{\epsilon}{n^2})f(S) \ge f(S-i+g(i))$. That is, $\frac{\epsilon}{n^2}f(S) + B_i(S-i) \ge B_{g(i)}(S-i)$. Using this we get

$$\begin{aligned} B_{g(i)}(S) = B_{g(i)}(S-i) + A_{ig(i)}(S-i) &\le B_{g(i)}(S-i) + \gamma(\frac{B_{g(i)}(S-i)+B_i(S-i)}{r-1}) \\ &\le \frac{2\gamma+r-1}{r-1}B_i(S-i) + \frac{\epsilon(\gamma+r-1)}{(r-1)n^2}f(S), \end{aligned}$$

where the equality follows from Lemma 3 and the first inequality from γ-meta-submodularity. Therefore,

$$\sum_{i\in S\setminus A} B_{g(i)}(S) \le \frac{2\gamma+r-1}{r-1}\sum_{i\in S\setminus A} B_i(S-i) + o(1)f(S).$$

Now, by Taylor's Theorem, Lemma 13, and the above inequality, we have

$$\begin{aligned} f(S\cup A) = F(\mathbb{1}_S \vee \mathbb{1}_A) = F(\mathbb{1}_S + \mathbb{1}_{A\setminus S}) &= F(\mathbb{1}_S) + \mathbb{1}_{A\setminus S}^T \nabla F(\mathbb{1}_S + \epsilon' \mathbb{1}_{A\setminus S}) \\ &\le F(\mathbb{1}_S) + 2^{4\gamma}\mathbb{1}_{A\setminus S}^T \nabla F(\mathbb{1}_S) = F(\mathbb{1}_S) + 2^{4\gamma}\sum_{i\in S\setminus A} B_{g(i)}(S) \\ &\le (1 + 2^{4\gamma}\cdot o(1))f(S) + \frac{2\gamma+r-1}{r-1}2^{4\gamma}\sum_{i\in S\setminus A} B_i(S-i) \end{aligned}$$

Therefore, using the monotonicity of f and Lemma 14 we get

$$\begin{aligned} f(A) \le f(S\cup A) &\le (1+2^{4\gamma}\cdot o(1))f(S) + \frac{2\gamma+r-1}{r-1}2^{4\gamma}(9\gamma+2)f(S) \\ &= [\frac{2\gamma+r-1}{r-1}2^{4\gamma}(9\gamma+2) + 1 + 2^{4\gamma}\cdot o(1)]f(S). \end{aligned}$$

□

Algorithm 4.1: Local search under matroid constraint

Input: A set function f, a matroid $\mathcal{M} = ([n], \mathcal{I})$ with circuits of minimum cardinality c, and $\epsilon > 0$.
$S_0 \leftarrow \arg\max_{\{v,v'\}\in\mathcal{I}} f(\{v, v'\})$
$S \leftarrow$ a base of $\mathcal{M}$ that contains S_0
while *S is not an approximate local optima* **do**
  Find $i \in S$ and $j \in [n] \setminus S$ such that $S - i + j \in \mathcal{I}$ and $f(S - i + j) \geq (1 + \frac{\epsilon}{n^2})f(S)$
  $S \leftarrow S - i + j$
Create a complete weighted bipartite graph G with node sets S and $[n] \setminus S$, and edge weights $w(i, j) := A_{ij}(S)$ for each $i \in S$ and $j \notin S$. Find a maximum weighted matching M in G of (edge) cardinality $\frac{c-1}{2}$, and let S' denote the node set of M.
Return $\arg\max\{f(S), f(S')\}$

One important question is whether such a local search algorithm is polytime. We show that if the algorithms starts from the best pair, it performs a polynomial number of swaps. First we need the following the following lemma.

Lemma 15. *Let f be a non-negative, monotone, γ-meta submodular function and $\mathcal{M} = ([n], \mathcal{I})$ be a matroid of rank r. Let $A \in \mathcal{I}$ be an optimum set, i.e.,*

$$A \in \arg\max_{R\in\mathcal{I}} f(R),$$

and

$$S_0 \in \arg\max_{\{v,v'\}\in\mathcal{I}} f(\{v, v'\}).$$

Then $f(A) \leq O(r(\gamma + 1)^{r-2})f(S_0)$.

Proof. Let $A = \{a_1, \ldots, a_r\}$ and $A_i = \{a_1, \ldots, a_i\}$ for $1 \leq i \leq r$. By definition of S_0 we know that $f(A_2) \leq f(S_0)$. Now by induction we show that for any $2 \leq i < j \leq n$, $B_{a_j}(A_i) \leq O((\gamma + 1)^{i-1})f(S_0)$. The base case is $i = 2$. By definition of $f(S_0)$, monotonicity and meta submodularity of f, we have

$$\begin{aligned}B_{a_j}(A_2) &= B_{a_j}(A_1) + A_{a_2a_j}(A_1) \leq B_{a_j}(A_1) + \gamma(B_{a_j}(A_1) + B_{a_2}(A_1)) \leq (2\gamma + 1)f(S_0) \\ &\leq O(\gamma + 1)f(S_0).\end{aligned}$$

Now assume that for $k < j \leq n$, we have $B_{a_j}(A_k) \leq O(\gamma^{k-1})f(S_0)$. We want to show that for $k + 1 < j \leq n$, we have $B_{a_j}(A_{k+1}) \leq O(\gamma^k)f(S_0)$.

$$\begin{aligned}B_{a_j}(A_{k+1}) &= B_{a_j}(A_k) + A_{a_{k+1}a_j}(A_k) \leq B_{a_j}(A_k) + \frac{\gamma}{k}(B_{a_{k+1}}(A_k) + B_{a_j}(A_k)) \\ &\leq (1 + \frac{2\gamma}{k})O((\gamma + 1)^{k-1})f(s_0) \leq O((\gamma + 1)^k)f(S_0).\end{aligned}$$

We know that

$$f(A) = f(A_2) + \sum_{i=3}^{r} B_{a_i}(A_{i-1}) \leq f(S_0) + \sum_{i=3}^{r} O((\gamma + 1)^{i-2})f(S_0) \leq O(r(\gamma + 1)^{r-2})f(S_0)$$

□

Using this we can bound the number of swaps that the approximate local search algorithm needs.

Proposition 13. *Local search algorithm (Algorithm 4.1) runs in* $O(n^4(\log(r) + r\log(\gamma+1)/\epsilon)$ *time on a* γ*-meta submodular functions and a matorid of rank* r*.*

Proof. Cost of finding S_0 is $O(n^2)$. Also, each iteration of the while loop costs $O(n^2)$. Let S_k be the solution after k iterations and A be an optimum solution. By Lemma 15, we know

$$f(S_k) \leq (1 + \frac{\epsilon}{n^2})^k f(S_0) \leq f(A) \leq O(r(\gamma+1)^{r-2}) f(S_0).$$

Taking the logarithm, we have

$$k\ln(1 + \frac{\epsilon}{n^2}) \leq O(\ln(r) + (r-2)\ln(\gamma+1)).$$

Noting that $\frac{x-1}{x} \leq \ln x$ for any $x > 0$, we have

$$k(\frac{\epsilon}{n^2})/(\frac{n^2+\epsilon}{n^2}) \leq O(\ln(r) + (r-2)\ln(\gamma+1)).$$

This yields the result. □

As with the continuous setting (Theorem 2), one can improve the performance ratios by requiring additional (discrete smoothness) conditions on higher order (first derivative) terms. We discuss this in the next section.

4.3 Local Search for Second-Order-Submodular γ-Meta-Submodular Functions

As we have seen the discrete analog of $\nabla_i F$ is the marginal gain set function $B_i(S)$. The following result shows that if these set functions are submodular, then the exponential factor from Theorem 17 improves to a quadratic factor. We remark that submodularity of the B_i's is just the notion of *second-order-submodularity* introduced in [52], and is also equivalent to the non-positivity of the third-order partial derivatives of the multi-linear extension. We first provide a key lemma for bounding the Taylor series expansion of multi-linear extension of second-order-submodular functions.

Lemma 16. *Let* $f : 2^n \to \mathbb{R}$ *be a non-negative, second-order-submodular set function and* F *be its multi-linear extension. Then for any* $R \subseteq [n]$, $\sum_{u\in R} B_u(R) \leq 2f(R)$*. If* f *is also monotone then* $x \in [0,1]^n$, $x^T\nabla^2 F(x)x \leq 2F(x)$*.*

Proof. For the first part, without loss of generality, let $R = [k]$ (we can always relabel the elements so that this is true) and $R_i = [i]$. By Lemma 3, we have

$$\sum_{i\in R} B_i(R) = \sum_{i=1}^{k} \big(f(\{i\}) + \sum_{j=1}^{k} A_{ij}(R_{j-1})\big).$$

Since $B_i(R_i) = B_i(R_{i-1})$, and $f(R_0) = f(\emptyset) = 0$ we have

$$2f(R) = 2\sum_{i=1}^{k} B_i(R_i) = 2\sum_{i=1}^{k} \big(f(\{i\}) + \sum_{j=1}^{i} A_{ij}(R_{j-1})\big).$$

Moreover, note that

$$\sum_{i=1}^{k}\sum_{j=1}^{k} A_{ij}(R_{j-1}) \le 2\sum_{i=1}^{k}\sum_{j=1}^{i} A_{ij}(R_{j-1})$$

since

$$\begin{aligned}\sum_{i=1}^{k}\sum_{j=i+1}^{k} A_{ij}(R_{j-1}) &= \sum_{j=1}^{k}\sum_{i=1}^{j-1} A_{ij}(R_{j-1}) = \sum_{j=1}^{k}\sum_{i=1}^{j-1} A_{ji}(R_{j-1})\\ &\le \sum_{j=1}^{k}\sum_{i=1}^{j-1} A_{ji}(R_{i-1}) = \sum_{j=1}^{k}\sum_{i=1}^{j} A_{ji}(R_{i-1})\\ &= \sum_{i=1}^{k}\sum_{j=1}^{i} A_{ij}(R_{j-1}),\end{aligned}$$

where the second equality follows from the fact that $A_{ij}(S) = A_{ji}(S)$ for all $i, j \in [n]$ and $S \subseteq [n]$, and the third equality from the fact that $A_{ii}(S) = 0$ for all $i \in [n]$ and $S \subseteq [n]$. The inequality follows since $R_{j-1} \supseteq R_{i-1}$ and f is second-order-submodular.

By non-negativity we also have that $2f(\{i\}) \ge f(\{i\})$. This yields the first part of the lemma.

We now discuss the second part. By the Taylor's Theorem, non-negativity, monotononicity and second-order-submodularity, we have

$$F(x) = F(0) + x^T\nabla F(0) + \frac{1}{2}x^T\nabla^2 F(\epsilon x)x \ge \frac{1}{2}x^T\nabla^2 F(\epsilon x)x \ge \frac{1}{2}x^T\nabla^2 F(x)x.$$

□

In order to achieve a sub-quadratic approximation factor in Theorem 13 we also require the function to be supermodular. Moreover, the local search algorithm must be significantly adapted and find a maximum matching in the last step.

Theorem 18. *Let f be a γ-meta-submodular function which is also second order submodular (that is, f's marginal gains are submodular). Let $\mathcal{M} = ([n], \mathcal{I})$ be a matroid of rank r and minimum circuit size c. Let $A \in \mathcal{I}$ be an optimum set, i.e., $A \in \arg\max_{R\in\mathcal{I}} f(R)$, and $S \in \mathcal{I}$ be an $(1 + \frac{\epsilon}{n^2})$-approximate local optima, i.e., for any i and j such that $S - i + j \in \mathcal{I}$, $(1 + \frac{\epsilon}{n^2})f(S) \ge f(S - i + j)$, where $\epsilon > 0$ is a constant. Then $f(A) \le O(\gamma + \frac{\gamma^2}{r})f(S)$. So Algorithm 4.1 gives an $O(\gamma + \frac{\gamma^2}{r})$-approximation. If f is also supermodular then Algorithm 4.1 gives an $O(\min\{\gamma + \frac{\gamma^2}{r}, \frac{\gamma r}{c-1}\}) \le O(\gamma^{3/2})$-approximation.*

Proof. Since f is monotone, we assume that $|S| = |A| = r$. Given the exchangeability property of matroids, there is a bijective mapping ([70]) $g : S \setminus A \to A \setminus S$ such that $S - i + g(i) \in \mathcal{I}$ where $i \in S \setminus A$. Since S is a $(1 + \frac{\epsilon}{n^2})$-approximate local optima, for all $i \in S \setminus A$ we have $(1 + \frac{\epsilon}{n^2})f(S) \geq f(S - i + g(i))$. That is,

$$\frac{\epsilon}{n^2} f(S) + B_i(S - i) \geq B_{g(i)}(S - i). \tag{4.8}$$

Using this we get

$$\begin{aligned} B_{g(i)}(S) = B_{g(i)}(S - i) + A_{ig(i)}(S - i) &\leq B_{g(i)}(S - i) + \gamma\Big(\frac{B_{g(i)}(S - i) + B_i(S - i)}{r - 1}\Big) \\ &\leq \frac{2\gamma + r - 1}{r - 1} B_i(S - i) + \frac{\epsilon(\gamma + r - 1)}{(r - 1)n^2} f(S) = \Big(\frac{2\gamma}{r - 1} + 1\Big) B_i(S) + \frac{\epsilon(\gamma + r - 1)}{(r - 1)n^2} f(S), \end{aligned}$$

where the first equality follows from Lemma 3, the first inequality from γ-meta-submodularity, and the last equality from $B_i(S) = B_i(S - i)$ for all $i \in [n]$ and $S \subseteq [n]$. Thus,

$$\begin{aligned} \sum_{i \in S \setminus A} B_{g(i)}(S) &\leq \Big(\frac{2\gamma}{r - 1} + 1\Big) \sum_{i \in S \setminus A} B_i(S) + |S \setminus A| \cdot \frac{\epsilon(\gamma + r - 1)}{(r - 1)n^2} f(S) \\ &\leq \Big(\frac{2\gamma}{r - 1} + 1\Big) \sum_{i \in S} B_i(S) + \frac{\epsilon(\gamma + r - 1)}{(r - 1)n} f(S) \\ &\leq \Big(\frac{4\gamma}{r - 1} + 2 + o(1)\Big) \cdot f(S). \end{aligned}$$

where the second inequality follows from monotonicity (i.e. $B_i(S) \geq 0$), and the last one follows from Lemma 16.

Now, by Taylor's Theorem and the submodularity of the marginal gains of f (i.e. the non-positivity of the third order marginal gains), γ-meta submodularity, and the above inequality, we have

$$\begin{aligned} f(A) &\leq f(S \cup A) = F(\mathbb{1}_S \vee \mathbb{1}_A) = F(\mathbb{1}_S + \mathbb{1}_{A \setminus S}) \\ &\leq F(\mathbb{1}_S) + \mathbb{1}_{A \setminus S}^T \nabla F(\mathbb{1}_S) + \frac{1}{2} \mathbb{1}_{A \setminus S}^T \nabla^2 F(\mathbb{1}_S) \mathbb{1}_{A \setminus S} \\ &\leq F(\mathbb{1}_S) + \Big(1 + \frac{\gamma |A \setminus S|}{|S|}\Big) \mathbb{1}_{A \setminus S}^T \nabla F(\mathbb{1}_S) \leq F(\mathbb{1}_S) + (1 + \gamma) \mathbb{1}_{A \setminus S}^T \nabla F(\mathbb{1}_S) \\ &= F(\mathbb{1}_S) + (1 + \gamma) \sum_{i \in S \setminus A} B_{g(i)}(S) \leq \Big(\frac{4\gamma^2}{r - 1} + \gamma\big(\frac{4}{r - 1} + 2 + o(1)\big) + 3 + o(1)\Big) f(S) \\ &= O\Big(\frac{\gamma^2}{r} + \gamma\Big) f(S). \end{aligned}$$

Now, we assume that f is also supermodular. Let $S \cap S' = \{a_1, \ldots, a_p\}$ and $S' \setminus S = \{b_1, \ldots, b_p\}$ where $\{a_i, b_i\}$'s are the edges of the matching. Also, let $T_i = \{a_1, \ldots, a_i\}$ and $R_i = \{b_1, \ldots, b_i\}$. Then since M is a maximum weighted matching, we have

$$\sum_{i \in S \setminus A} A_{ig(i)}(S) \leq \frac{2 \cdot |S \setminus A|}{c - 1} \sum_{i=1}^{p} A_{a_i b_i}(S) \leq \frac{2r}{c - 1} \sum_{i=1}^{p} A_{a_i b_i}(S). \tag{4.9}$$

We also have that

$$
\begin{aligned}
f(S') &= \sum_{i=1}^{p}(f(T_i \cup R_i) - f(T_{i-1} \cup R_{i-1})) = \sum_{i=1}^{p}(B_{a_i}(T_{i-1} \cup R_{i-1}) + B_{b_i}(T_{i-1} \cup R_{i-1} + a_i)) \\
&= \sum_{i=1}^{p} \Big(B_{a_i}(T_{i-1} \cup R_{i-1}) + f(\{b_i\}) + \sum_{j=1}^{i} A_{b_i a_j}(T_{j-1}) + \sum_{j=1}^{i-1} A_{b_i b_j}(T_{i-1} + a_i \cup R_{j-1})\Big) \\
&= \sum_{i=1}^{p} \Big(B_{a_i}(T_{i-1} \cup R_{i-1}) + A_{b_i a_i}(T_{i-1}) + f(\{b_i\}) \\
&+ \sum_{j=1}^{i-1} A_{b_i a_j}(T_{j-1}) + \sum_{j=1}^{i-1} A_{b_i b_j}(T_{i-1} \cup R_{j-1} + a_i)\Big) \\
&\geq \sum_{i=1}^{p} A_{a_i b_i}(T_{i-1}) \geq \sum_{i=1}^{p} A_{a_i b_i}(S).
\end{aligned} \tag{4.10}
$$

where the third equality follows from Lemma 3, the first inequality from monotonocity and supermodularity (i.e. all the B_i and A_{ij} terms are non-negative), and the last inequality from second-order-submodularity and the fact that $T_i \subseteq S$ for any $i = 1, \ldots, p$.

Hence, by combining (4.9) and (4.10), we get

$$
\sum_{i \in S \setminus A} A_{ig(i)}(S - i) = \sum_{i \in S \setminus A} A_{ig(i)}(S) \leq \frac{2r}{c-1} \sum_{i=1}^{p} A_{a_i b_i}(S) \leq \frac{2r}{c-1} f(S'). \tag{4.11}
$$

Using Taylor's Theorem

$$
\begin{aligned}
f(A) &\leq f(S \cup A) = F(\mathbb{1}_S \vee \mathbb{1}_A) = F(\mathbb{1}_S + \mathbb{1}_{A \setminus S}) \\
&\leq F(\mathbb{1}_S) + \mathbb{1}_{A \setminus S}^T \nabla F(\mathbb{1}_S) + \frac{1}{2} \mathbb{1}_{A \setminus S}^T \nabla^2 F(\mathbb{1}_S) \mathbb{1}_{A \setminus S} \\
&\leq F(\mathbb{1}_S) + \Big(1 + \frac{\gamma |A - S|}{|S|}\Big) \mathbb{1}_{A \setminus S}^T \nabla F(\mathbb{1}_S) \leq F(\mathbb{1}_S) + (1 + \gamma) \mathbb{1}_{A \setminus S}^T \nabla F(\mathbb{1}_S) \\
&= F(\mathbb{1}_S) + (1 + \gamma) \sum_{i \in S \setminus A} B_{g(i)}(S) \\
&= f(S) + (1 + \gamma)(\sum_{i \in S \setminus A} B_{g(i)}(S - i) + \sum_{i \in S \setminus A} A_{ig(i)}(S - i)) \\
&\leq f(S) + (1 + \gamma)\Big(\frac{r\epsilon}{n^2} f(S) + \sum_{i \in S \setminus A} B_i(S - i) + \frac{2r}{c-1} f(S')\Big) \\
&\leq f(S) + (1 + \gamma)\Big(\frac{r\epsilon}{n^2} f(S) + 2f(S) + \frac{2r}{c-1} f(S')\Big) \leq O\big(\frac{\gamma r}{c-1}\big) \max\{f(S), f(S')\}.
\end{aligned}
$$

where the second inequality follows from second-order-submodularity (i.e. the non-positivity of the third order derivatives), the third inequality from γ-meta submodularity, the fifth inequality from (4.8) and (4.11), and the second to last inequality from Lemma 16.

We then have that if $r \leq \sqrt{\gamma}$ then $\gamma r \leq \gamma^{3/2}$, and if $r \geq \sqrt{\gamma}$ then $\frac{\gamma^2}{r} + \gamma \leq \gamma^{3/2}$. Therefore, $f(A) \leq O(\gamma^{3/2}) \max\{f(S), f(S')\}$.

$\square$

Let $\mathcal{S}_\gamma$ denote the class of functions $f \in \mathcal{G}_\gamma$ which are also supermodular and 2nd-order-submodular. Note that $\mathcal{S}_\gamma$ properly contains the family $\mathcal{O}_\gamma$ of discrete quadratic functions which are one-sided γ-smooth. By Theorem 18 there is an $O(\gamma^{3/2})$-approximation factor for functions in $\mathcal{S}_\gamma$, and hence this class provides our most general answer to question (Q1).

4.4 Future Work

We conjecture that the maximization of general γ-meta-submodular functions admits a cubic approximation in terms of γ. This is one of the main questions that can be addressed in future research. Another interesting avenue is to investigate the smoothness of multi-linear extension of meta-submodular-functions. We conjecture that for $\gamma > 0$, it is one-sided $O(\gamma)$-smooth (see Conjecture 1).

Chapter 5

Distributed Maximization of Meta-Submodular Functions

Many problems from different areas of machine learning and data mining can be modeled as an optimization problem that tries to maximize the sum of a sum-sum diversity function (which is the sum of the distances between all of the pairs in a given subset) and a non-negative monotone submodular function. Such a function is in the class of 1-meta-submodular functions. Examples include query diversification problem in the area of databases [26, 59], search result diversification [5, 29], and recommender systems [83]. The size of the datasets in these applications is growing rapidly, and there is a need for scalable methods to tackle these problems on huge datasets. Inspired by these applications, we propose an algorithm for approximately solving this optimization problem with a theoretical guarantee in distributed and steaming settings. Borodin et al. [14] presented a 0.5-approximation for this optimization problem in the centralized setting in which data can be stored and processed on a single machine. In this paper, we consider this problem for big data settings where the data cannot be stored on a single machine, or the processing time is too high for a single machine. We show that our algorithm achieves a 1/31-approximation. Note that solving this problem in a distributed or streaming setting is strictly harder than solving it in the centralized setting because, in the aforementioned settings, the algorithm does not use all of the data. As a result, our algorithm is $\frac{\sqrt{d/k}}{2}$ times faster in the distributed setting and it needs $\sqrt{d/k}$ times less memory in the streaming setting compared to the centralized setting, where d is the size of the ground set (for example, the number of features in the feature selection problem), and k is the number of machines (in the distributed setting) or is the number of partitions of the data (in the streaming setting). Therefore, our algorithm gives a worse approximate solution compared to the centralized method of Borodin et al. [14] but it is much faster and needs less memory. This trade-off might be interesting and useful in some applications.

One of the problems that can be modeled as such an optimization problem and is in need of scalable methods in modern applications is multi-label feature selection. The diversity part controls the redundancy of the selected features and the submodular part is to promote features that are relevant to the labels. A multi-label dataset is made up of a number of samples, features, and labels. Each sample is a set of values for the features and labels. Usually, labels have binary values. For example, if a patient has diabetes or not. Multi-label datasets can be found in different areas, including but not limited to semantic image annotation, protein and gene function studies, and text categorization [45]. Applications, number, and size of such datasets are growing very rapidly, and it is necessary to develop efficient and scalable methods to deal with them.

Feature selection is a fundamental problem in machine learning. Its goal is to decrease the dimensionality of a dataset in order to improve the learning accuracy, decrease the learning and prediction time, and prevent overfitting. There are three different categories of feature selection methods depending on their interaction with the learning methods. Filter methods select the features based on the intrinsic properties of the data and are totally independent of the learning method. Wrapper methods select the features according to the accuracy of a specific learning method, like SVMs. Finally, embedded methods select the features as a part of their learning procedure [38]. Decision trees and use of ℓ_0 and ℓ_1 regularization for feature selection fall into the latter. When the number of features is large, filter methods are a reasonable choice since they are fast, resistant to over-fitting, and independent of the learning model. Therefore, we can quickly select a number of features with filter methods and then try different learning methods to see which one fits the data better (possibly with wrapper or embedded feature selection methods). However, with millions of features, centralized filter methods are not applicable anymore. To deal with such huge datasets, we need scalable methods. Although there were efforts to develop scalable and distributed filter methods for single-label datasets [9, 85], to the best of our knowledge, there are no previous distributed multi-label feature selection method.

In this paper, we propose an information theoretic filter feature selection method for multi-label datasets that is usable in distributed, streaming, and centralized settings. In the centralized setting, all of the data is stored and can be processed on a single machine. In the distributed setting, the data is stored on multiple machines, and there is no shared memory between machines. In the streaming setting, although the computation is done on a single machine, this machine does not have enough memory to store all of the data at once. The data in our method is distributed vertically which means that the features are distributed between machines instead of samples (horizontal distribution). Feature selection is considered harder when the data is distributed vertically because we lose much information about the relations of the features [10]. However, when the number of instances is small, and the number of features is large (for example, biological or medical datasets) vertical distribution is the only reasonable choice. Our work can be seen as an extension of Borodin et al. [14] to distributed and streaming settings or an extension of Zadeh et al. [85] to multi-label data. However, our results cannot be derived from these previous works in a straightforward manner. The main results of this chapter are listed in the following.

- We present a greedy algorithm for maximizing the sum of a sum-sum diversity function and a non-negative monotone submodular function in the distributed and streaming settings. We prove that it achieves a constant factor approximation of the optimal solution.

- We formulate the multi-label feature selection problem as such a combinatorial optimization problem. Using this formulation we present information theoretic filter feature selection methods for distributed, steaming, and centralized settings. The distributed method is the first distributed multi-label feature selection method proposed in the literature.

- We perform an empirical study of the proposed distributed method and compare its results to different centralized multi-label feature selection methods. We show that the results of the distributed method are comparable to the current centralized methods in the literature. We also compare the runtime and the value of the objective function that our centralized and distributed methods achieve. Note that the centralized methods have access to the all of the data and can do computation on it. We do not expect that our distributed or streaming method to beat the centralized methods because it is not possible. However, we argue that our results are comparable to the results of centralized methods and our method is much faster (in case of the distributed setting) and needs much less memory (in case of the streaming setting). We compared our results with the centralized methods (this comparison is unfair to the distributed setting) in the literature because to the best of our knowledge there is no distributed multi-label feature selection method prior to this work.

5.1 Related Work

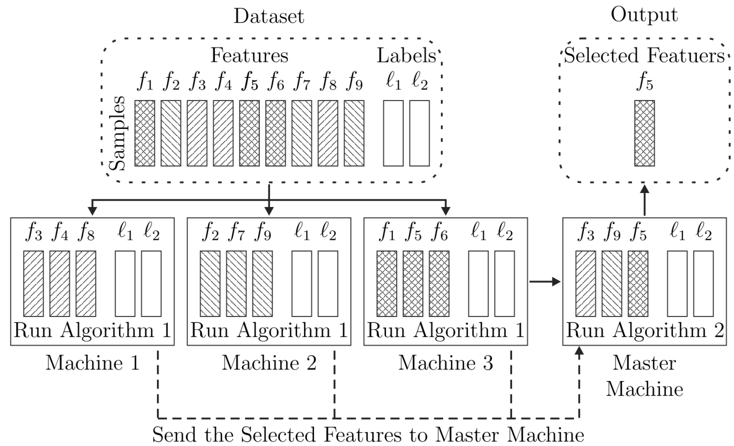
In this section, we review the previous works on different aspects of the problem including diversity maximization, submodular maximization, composable core-sets, and feature selection.

Diversity Maximization and Submodular Maximization

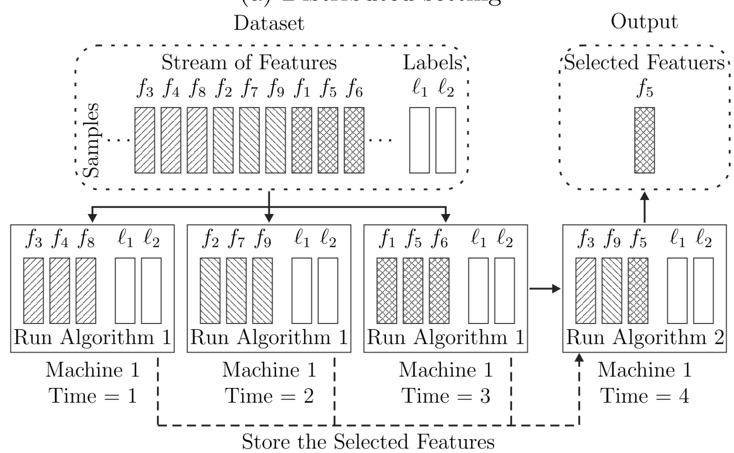
Usually, the diversity maximization problem is defined on a metric space of a set of points U with the goal of finding a subset of them which maximizes a diversity function subject to a constraint. For example, a cardinality constraint or a matroid constraint. If S is a subset of the points, the sum-sum diversity of S is $D(S) = 0.5 \sum_{x \in S} \sum_{y \in S} d(x, y)$ where $d(., .)$ is a metric distance. In the centralized setting, a simple greedy or local search algorithm can achieve a half approximation of the optimal solution subject to $|S| = k$ [1, 40]. TA better approximation factor is not achievable under the planted clique conjecture [8, 14].

Submodular functions are important concepts in machine learning and data mining with many applications. See Krause and Guestrin [55] for their applications. A submodular function is a set function with a diminishing marginal gain. A function $f : 2^U \to \mathbb{R}$ is submodular if $f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$ for any $A \subseteq B \subset U$, and $x \in U \setminus B$. It is monotone if $f(A) \leq f(B)$ and it is non-negative if $f(A) \geq 0$ for any $A \subseteq B \subseteq U$. Maximizing a monotone submodular function subject to a cardinality constraint is NP-hard but using a simple greedy algorithm we can achieve $(1 - \frac{1}{e})$ of the optimal solution. A better approximation factor is not achievable using a polynomial time algorithm unless P=NP [53].

Let U be a set and $f(.)$ be a submodular function defined on U and $d(., .)$ be a metric distance defined between pairs of elements of U. Borodin et al. [14] showed that in the centralized setting, using a simple greedy algorithm, we can achieve half of the optimal value for maximizing $f(S) + \lambda \sum_{\{u,v\}:u,v \in S} d(u, v)$ subject to $S \subseteq U$ and $|S| = k$. This result is extended to semi-metric distances in Zadeh and Ghadiri [84]. Similar problems are considered in Dasgupta et al. [24] where the diversity part can be other diversity functions. Namely, they considered the sum-sum diversity, the minimum spanning tree, and the minimum of distances between all pairs. They showed that the greedy algorithm achieves a constant factor approximation in all of these cases.



(a) Distributed setting



(b) Streaming setting

Figure 5.1: Algorithm 5.3 operating in big data settings.

Composable Core-sets

In computational geometry, a core-set is a small subset of points that approximately preserve a measure of the original set [2]. Composable core-sets extend this property to the combination of sets. Therefore, they can be used in a divide and conquer manner to find an approximate solution. Let U be a set, $f : 2^U \to \mathbb{R}$ be a set function on U, $(T^1, \ldots, T^m)$ be a random partitioning of elements of U, and k be a positive integer. Let $\texttt{OPT}(T) = \arg\max_{S \subseteq T, |S|=k} f(S)$ where $T \subseteq U$. Let $\texttt{ALG}$ be an algorithm which takes $T \subseteq U$ as an input and outputs $S \subseteq T$. For $\alpha > 0$, we call $\texttt{ALG}$ an α-approximate composable core-set with size k for f if the size of its output is k and $f(\texttt{OPT}(\texttt{ALG}(T^1) \cup \cdots \cup \texttt{ALG}(T^m))) \geq \alpha f(\texttt{OPT}(T^1 \cup \cdots \cup T^m))$ [41]. We call $\texttt{ALG}$ an α-approximate *randomized* composable core-set with size k for f if the size of its output is k and $\mathbb{E}[f(\texttt{OPT}(\texttt{ALG}(T^1) \cup \cdots \cup \texttt{ALG}(T^m)))] \geq \alpha f(\texttt{OPT}(T^1 \cup \cdots \cup T^m))$ [61]. Composable core-sets and randomized composable core-sets can be used in distributed settings (like the MapReduce framework) and streaming settings (see Figure 5.1).

Composable core-sets first were used to approximately solve several diversity maximization problems in distributed and streaming settings [41]. It resulted in an approximation algorithm for the sum-sum diversity maximization with an approximation factor of less than 0.01. This approximation factor is improved to $\frac{1}{12}$ in Aghamolaei et al. [4]. Randomized composable core-sets were first introduced to tackle submodular maximization problem in distributed and streaming settings which resulted in a 0.27-approximation algorithm for monotone submodular functions [61]. Then they were used to improve the approximation factor of the sum-sum diversity maximization from $\frac{1}{12}$ to 0.25 [85]. The randomized composable core-sets used in the latter case find the approximate solution with high probability instead of expectation.

There are a number of other works on distributed submodular maximization [7, 62]. Moreover, submodular and weak submodular functions are used for distributed *single-label* feature selection [47]. We should note that the discussed objective function in our work is neither submodular nor weak submodular. This is because of the diversity term of the function. An advantage of using this diversity function is that it is evaluated by a pairwise distance function. As a result, it is easy to evaluate our objective function on datasets with few samples. On the contrary, evaluating the pure submodular functions, that were used for feature selection in the literature, are quite hard and need a large amount of data and computing power.

Feature Selection and Multi-label Feature Selection

Filter feature selection methods select features independent of the learning algorithm. Hence, they are usually faster and immune to overfitting [38]. Mutual information based methods are a well-known family of filter methods. The best-known method of this kind for single-label feature selection is minimum redundancy and maximum relevance (mRMR) which tries to find a subset of features S that maximizes the following objective function using a greedy algorithm

$$\frac{1}{|S|}\sum_{x_i \in S} I(x_i, c) - \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i, x_j),$$

where $I(.,.)$ is the mutual information function, and c is the label vector [66]. The proposed method in this paper can be seen as a variation of mRMR which is capable of being used for multi-label feature selection in distributed, streaming, and centralized settings.

Although there have been great advancements in centralized feature selection, there are few works on distributed feature selection, and most of them distribute the data horizontally. Zadeh et al. [85] was the first work on the single-label vertically distributed feature selection that considered the redundancy of the features. Their method selects features using randomized composable core-sets in order to maximize a diversity function defined on the features. Although there are some similarities between the formulations presented in Zadeh et al. [85] and this work, we should note that the single-label formulation cannot be applied directly to multi-label datasets. Moreover, maximization of the functions and the analysis of the algorithms to prove the theoretical guarantee are completely different.

Most of the multi-label feature selection methods transform the data to a single-label form. Binary relevance (BR) and label powerset (LP) are two common ways to do so. BR methods consider each label separately and use a single-label feature selection method to select features for each label, and then they aggregate the selected features. A disadvantage of BR methods is that they cannot consider the relations of the labels. LP methods consider the multi-label dataset as one single-label multi-class dataset where each class of its single label are a possible combination of labels in the dataset (treating the labels as a binary string). Then they apply a single-label feature selection method. Although LP methods consider the relations of the labels, they have significant drawbacks. For example, some classes may end up with very few samples or none at all. Moreover, the method is biased toward the combination of the labels which exist in the training set [45]. Our proposed method does not transform the data to single-label data and is designed in a way to not suffer from the mentioned disadvantages.

5.2 Formulating the Multi-Label Feature Selection Problem

Let U be a set of d features and L be a set of t labels. We also have a set A of n instances each of which is a vector of observations for elements of $U \cup L$. The goal of *multi-label feature selection* is to find a small *non-redundant* subset of U which can *predict* labels in L accurately. In order to quantify redundancy it is natural to use a metric distance d over the feature set to measure dissimilarity. In our application (feature selection) we are particularly interested in the following metric distance. For any $u_i, u_j \in U$, we define

$$\begin{aligned} d(u_i, u_j) &= 1 - \frac{I(u_i, u_j)}{H(u_i, u_j)} \\ &= 1 - \frac{\sum_{x \in u_i, y \in u_j} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}}{-\sum_{x \in u_i, y \in u_j} p(x,y) \log p(x,y)}, \end{aligned}$$

where $H(.,.)$ is the joint entropy and $I(.,.)$ is the mutual information. This distance function is called *normalized* (values lie between 0 and 1) *variation of information* and it is a metric [65]. In Zadeh et al. [85], this distance function plus a modular function is used for single-label feature selection.

In order to quantify the predictive quality of the selected features, we define a non-negative monotone submodular function $g : 2^U \to \mathbb{R}$ which measures the relevance of the selected features to the labels. For any positive integer p, we define

$$g(S) = \sum_{\ell \in L} \max_{x \in S}{}^{\mathrm{p}} \{MI(x, \ell)\},$$

where $\max^{\mathrm{p}}{}_{x \in S}\{MI(x,\ell)\}$ is the sum of the p largest numbers in $\{MI(x,\ell) | x \in S\}$. Here $MI(x,\ell) = \frac{I(x,\ell)}{\sqrt{H(x)H(\ell)}}$ is the normalized mutual information where $H(.)$ is the entropy function and the value $MI(.,.)$ lies in $[0,1]$. Note that if we only have one label (i.e., $|L| = 1$), and $p = d$ (the number of all features of the dataset) then g will be exactly the modular function used in Zadeh et al. [85]. Therefore, our formulation is a generalization of theirs. Using the $\max^{\mathrm{p}}$ function, this formulation tries to select at least p relevant features for each label. In order to understand the importance of $\max^{\mathrm{p}}$ function, we discuss two extreme cases: $p = 1$ and $p = d$. If $p = 1$ then a feature that is somewhat relevant to all the features can dominate the $g(S)$ and prevent other features, that are highly relevant to one or few features, to get selected. If $p = d$ then a label that has a lot of relevant features can dominate $g(S)$ and prevent other labels to get relevant features, while a few features would be enough for predicting this label with a high accuracy. In the following lemma, we show that g has the nice properties we need in our model.

Lemma 17. *g is a non-negative, monotone, submodular function.*

Proof. Clearly g is non-negative and monotone. Since the sum of submodular functions is a submodular function, We only need to show that $\max^{\mathrm{p}}{}_{x\in S}\{MI(x,\ell)\}$ is submodular. We assume that $\max^{0}{}_{x\in S}\{MI(x,\ell)\} = 0$. Let $S \subseteq T \subset U$ and $a \in U \setminus T$. We show that

$$\begin{aligned}&\max_{x\in S\cup\{a\}}{}^{\mathrm{p}}\{MI(x,\ell)\} - \max_{x\in S}{}^{\mathrm{p}}\{MI(x,\ell)\}\\ &\qquad \geq \max_{x\in T\cup\{a\}}{}^{\mathrm{p}}\{MI(x,\ell)\} - \max_{x\in T}{}^{\mathrm{p}}\{MI(x,\ell)\}.\end{aligned}$$

We have two cases. If $MI(a,\ell)$ is not among the p largest numbers of $\{I(x,\ell)|x \in S \cup \{a\}\}$ then both sides of the above inequality are zero. If $MI(a,\ell)$ is among the p largest numbers of $\{I(x,\ell)|x \in S \cup \{a\}\}$ then the left hand side of the inequality is equal to $MI(a,\ell) - MI(b,\ell)$ where b is the p'th largest number in $\{I(x,\ell)|x \in S\}$. The right hand side is equal to $\max\{0, MI(a,\ell) - MI(c,\ell)\}$ where c is the p'th largest number in $\{I(x,\ell)|x \in T\}$. The p'th largest number in $\{I(x,\ell)|x \in T\}$ is greater than or equal to the p'th largest number in $\{I(x,\ell)|x \in S\}$ because $S \subseteq T$. Therefore, in this case $MI(a,\ell) - MI(b,\ell) \geq \max\{0, MI(a,\ell) - MI(c,\ell)\}$ and the inequality holds. □

If we define $f(S) = g(S) + \sum_{\{u,v\}\in S} d(u,v)$, then our feature selection model reduces to solving the following combinatorial optimization problem.

$$\max_{\substack{S\subseteq U\\|S|=k}} f(S) = \max_{\substack{S\subseteq U\\|S|=k}} \{g(S) + \sum_{\{u,v\}\in S} d(u,v)\}, \tag{5.1}$$

where $d(.,.)$ is a metric distance and $g(.)$ is a non-negative monotone submodular function. Moreover f is 1-meta-submodular. In the actual feature selection method we are free to scale the relative contributions of the diversity or submodular parts, since both metric and submodular functions are closed under multiplication by a positive constant. Hence, we use a weighted version of the objective function in our application.

The problem (5.1) is NP-hard but Borodin et al. [14] show that Algorithm 5.2 is a half approximation in the centralized setting. Note that this is a greedy algorithm under the objective where $g(S)$ is scaled by $\frac{1}{2}$. On the other hand, Algorithm 5.1 is a standard greedy algorithm for (5.1) and in the next section we show it is a constant factor randomized composable core-set for any functions f which are the sum of a sum-sum diversity function and a non-negative, monotone, submodular function. Combining these we conclude that Algorithm 5.3 is a constant factor approximation algorithm for maximizing f. Moreover, Algorithm 5.3 can be used both in distributed and streaming settings, as illustrated in Figure 5.1.

In our experiments, to select k features, we use the following function.

$$h(S) = (1-\lambda)\frac{k(k-1)}{2p|L|} g(S) + \lambda \sum_{x_i,x_j\in S} d(x_i,x_j). \tag{5.2}$$

Algorithm 5.1: Greedy

```
Input: Set of features U, set of labels L, number of features we want to select k.
Output: Set S ⊂ U with |S| = k.
S ← {arg max_{u∈U} g({u})};
forall 2 ≤ i ≤ k do
    u* ← arg max_{u∈U\S} g(S ∪ {u}) − g(S) + Σ_{x∈S} d(x, u);
    // This arg max has a consistent tiebreaking rule (see Definition 1).
    Add u* to S;
return S;
```

Algorithm 5.2: AltGreedy

```
Input: Set of features U, set of labels L, number of features we want to select k.
Output: Set S ⊂ U with |S| = k.
S ← {arg max_{u∈U} g({u})};
forall 2 ≤ i ≤ k do
    u* ← arg max_{u∈U\S} 1/2 (g(S ∪ {u}) − g(S)) + Σ_{x∈S} d(x, u);
    Add u* to S;
return S;
```

As discussed, the first term of $h(S)$ controls redundancy of the selected features and the second term is to promote features that are relevant to the labels. The term $\frac{k(k-1)}{2p|L|}$ is a normalization coefficient to make the range of both terms the same. Also, λ is a hyper-parameter which controls the effect of two criteria on the final function.

5.3 Maximizing the Sum of a Sobmodular Function and a Diversity Function

Let $f(S) = D(S)+g(S)$ be a set function defined on 2^U where $g(S)$ is a non-negative, monotone, submodular function and $D(S)$ is a sum-sum diversity function, i.e. $D(S) = \sum_{\{u,v\}\in S} d(u, v)$ where $d(.,.)$ is a metric distance. In this section, we show that Algorithm 5.1 is a constant factor randomized composable core-set with size k for f. We also show that running Algorithm 5.3 which is equivalent to running Algorithm 5.1 in each slave machine and then running Algorithm 5.2 in the master machine on the union of outputs of slave machines is a constant factor randomized approximation algorithm for maximizing f subject to a cardinality constraint.

Our proof follows from two key lemmas which bound the diversity and submodular portions of an optimal solution. We use O to denote a global optimum. To state the lemmas, we need the following notations. Let $\texttt{OPT}(T) = \arg\max_{R\subseteq T} f(R)$ subject to $|R| = k$. Let U be the set of all elements (for example, the set of all features for the feature selection problem) and

Algorithm 5.3: Distributed greedy

Input: Set of features U, set of labels L, number of features we want to select k, number of machines m.
Output: Set $S \subset U$ with $|S| = k$.
Randomly partition U into $(T_i)_{i=1}^m$;
forall $1 \leq i \leq m$ **do**
  $S_i \leftarrow$ output of $Greedy(T_i,\, L,\, k)$;
$S \leftarrow$ output of $AltGreedy(\cup_{i=1}^m S_i,\, L,\, k)$;
Return S;

$(T^1, \ldots, T^m)$ be a random partitioning of the elements of U.

We use the following key concept of a β-nice algorithm from Mirrokni and Zadimoghaddam [61] throughout our analysis.

Definition 6. *Let f be a set function on 2^U. Let* ***ALG*** *be an algorithm that given any $T \subseteq U$ outputs* ***ALG***$(T) \subseteq T$. *Let $t \in T \setminus$* ***ALG***(T). *For $\beta \in \mathbb{R}^+$, we call* ***ALG*** *a β-nice algorithm if it has the following properties.*

- $\texttt{ALG}(T) = \texttt{ALG}(T \setminus \{t\})$.
- $f(\texttt{ALG}(T) \cup \{t\}) - f(\texttt{ALG}(T)) \leq \beta \frac{f(\texttt{ALG}(T))}{k}$.

The intuition behind the first condition is simply that by removing an element of T which is not used in the algorithm's output, we do not change the output. This is effectively a condition on how we perform tiebreaking. The second condition helps to bound $f(\texttt{ALG}(T) \cup O)$ where O is a global optima. Our analysis heavily relies on the following theorem.

We use the following Theorem and techniques from a number of papers [4, 41, 61, 85] to prove the key lemmas. Even in cases where some parts of proofs are similar to previous work we include a complete proof for the sake of completeness. We should note that our analysis is not a straightforward combination of the ideas in the mentioned papers.

Theorem 19. *Let $k \geq 10$. Algorithm 5.1 is a 5-nice algorithm for $f(.) = D(.) + g(.)$. Also, if* ***ALG*** *is Algorithm 5.1, $T \subseteq U$, and $t \in T \setminus$* ***ALG***(T), *then* $\frac{4.5}{k-1} f(\texttt{ALG}(T)) \geq \sum_{x \in \texttt{ALG}(T)} d(t, x)$.

Proof. Let $\texttt{ALG}$ be the Algorithm 5.1, $T \subseteq U$, $t \in T \setminus \texttt{ALG}(T)$, and $x_1, \ldots, x_k$ be the elements that $\texttt{ALG}$ selected in the order of selection. Also, let $S_i = \{x_1, \ldots, x_i\}$ and $S_0 = \emptyset$.

For the first property of β-nice algorithms it is enough to have a consistent tiebreaking rule for $\texttt{ALG}$. It is sufficient to fix an ordering on all elements of U up front. If some iteration finds multiple elements with the same maximum marginal gain, then it should select earliest one in the a priori ordering.

Now we prove the second property of the β-nice algorithms for $\texttt{ALG}$. Because of the greedy selection of ALG, we have the following inequalities.

$$\Delta(x_1, S_0) \geq \Delta(t, S_0)$$
$$\Delta(x_2, S_1) + d(x_2, x_1) \geq d(t, x_1) + \Delta(t, S_1)$$
$$\Delta(x_3, S_2) + \sum_{i=1}^{2} d(x_3, x_i) \geq \sum_{i=1}^{2} d(t, x_i) + \Delta(t, S_2)$$
$$\cdots$$
$$\Delta(x_k, S_{k-1}) + \sum_{i=1}^{k-1} d(x_k, x_i) \geq \sum_{i=1}^{k-1} d(t, x_i) + \Delta(t, S_{k-1})$$

Adding these inequalities together gives the following inequality.

$$g(S_k) + D(S_k) \geq \sum_{i=1}^{k-1}(k-i)d(t, x_i) + \sum_{i=0}^{k-1} \Delta(t, S_i) \geq \sum_{i=1}^{k-1}(k-i)d(t, x_i) + k\Delta(t, S_k), \tag{5.3}$$

where the second inequality holds because of the submodularity of g. Note that

$$f(\texttt{ALG}(T) \cup \{x\}) - f(\texttt{ALG}(T)) = \Delta(t, \texttt{ALG}(T)) + \sum_{x \in \texttt{ALG}(T)} d(x, t). \tag{5.4}$$

One may thus note that if the right-hand side coefficients in (5.3) were all $k/2$ (instead of $k-i$) we would have 2-niceness of the algorithm. Our strategy is to achieve this by shifting some of the "weight" from coefficients where $k-i > k/2$ to coefficients $< k/2$. This uses the metric inequality since $d(x_{k-i}, x_i) + d(x_i, t) \geq d(x_{k-i}, t)$. Hence if we added $d(x_{k-i}, x_i)$ to both sides of (5.3), then we may increase the coefficient of $d(t, x_{k-i})$ by 1 at the expense of reducing the coefficient of $d(t, x_i)$ by 1.

We use this idea to fix all of the "small" components in bulk by adding a batch of *distinct* distances to both sides of (5.3). Since these distances are distinct, we increase the left-hand side by at most $D(S_k)$. In particular, the new left-hand side will be at most $2(g(S_k) + D(S_k))$.

The batch of distances we add to both sides of the inequality is $\sum_{i=\lceil \frac{k}{2} \rceil+1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor-1} d(x_i, x_j)$. Clearly these distances are distinct so we now need to make sure that the strategy produces the desired coefficients of terms $d(t, x_i)$. More formally, we claim that the following inequality holds.

Claim 1.

$$\sum_{i=1}^{k-1}(k-i)d(t, x_i) + \sum_{i=\lceil \frac{k}{2} \rceil+1}^{k} \sum_{j=1}^{i-\lfloor \frac{k}{2} \rfloor-1} d(x_i, x_j) \geq \sum_{i=1}^{k}(\lceil \frac{k}{2} \rceil - 1)d(t, x_i)$$

We prove this claim later. Using this we have the following.

$$
\begin{aligned}
2(g(S_k)+D(S_k)) &\geq g(S_k)+D(S_k)+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1} d(x_i,x_j) \\
&\geq \sum_{i=1}^{k-1}(k-i)d(t,x_i)+k\Delta(t,S_k)+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1} d(x_i,x_j) \\
&\geq \sum_{i=1}^{k}(\lceil\frac{k}{2}\rceil-1)d(t,x_i)+(\lceil\frac{k}{2}\rceil-1)\Delta(t,S_k)
\end{aligned}
$$

where the second inequalities holds because of the metric property, i.e. triangle inequality, and monotonicity of g. By using the above inequality, non-negativity of g, and (5.4) we have

$$
\begin{aligned}
\frac{2}{\lceil\frac{k}{2}\rceil-1}f(\texttt{ALG}(T)) = \frac{2}{\lceil\frac{k}{2}\rceil-1}(g(S_k)+D(S_k)) &\geq \sum_{i=1}^{k} d(t,x_i)+\Delta(t,S_k) \\
&= f(\texttt{ALG}(T)\cup\{t\})-f(\texttt{ALG}(T)).
\end{aligned}
$$

We can easily see that for $k\geq 10$, $\frac{5}{k}\geq\frac{2}{\lceil\frac{k}{2}\rceil-1}$ and $\frac{4.5}{k-1}\geq\frac{2}{\lceil\frac{k}{2}\rceil-1}$. Therefore, ALG is a 5-nice algorithm for f and because of monotonicity of g, $\frac{4.5}{k-1}f(\texttt{ALG}(T))\geq\sum_{i=1}^{k} d(t,x_i)$. $\square$

Now we prove Claim 1 to conclude Theorem 19.

Proof of Claim 1. Note that $k=\lceil\frac{k}{2}\rceil+\lfloor\frac{k}{2}\rfloor$ and $\lfloor\frac{k}{2}\rfloor+1\geq\lceil\frac{k}{2}\rceil$. First, we show that

$$
\sum_{j=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-j)d(t,x_j)=\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1} d(t,x_j). \tag{5.5}
$$

In the right hand side of (5.5), $d(t,x_j)$ appears in the inner summation when $i-\lfloor\frac{k}{2}\rfloor-1\geq j$ or equivalently, when $i\geq j+\lfloor\frac{k}{2}\rfloor+1$. We know that $k\geq i\geq\lceil\frac{k}{2}\rceil+1$. We also know that $j\geq 1$. Hence, $j+\lfloor\frac{k}{2}\rfloor+1\geq\lceil\frac{k}{2}\rceil+1$. Therefore, $d(t,x_j)$ definitely appears in the inner summation when $k\geq i\geq j+\lfloor\frac{k}{2}\rfloor+1$. This means that $d(t,x_j)$ appears $k-j-\lfloor\frac{k}{2}\rfloor=\lceil\frac{k}{2}\rceil-j$ many times in the right hand side of (5.5). Moreover, note that the index j in the right hand side of (1) ranges between 1 and $k-\lfloor\frac{k}{2}\rfloor-1$. Hence (5.5) holds. Let

$$
A=\sum_{i=k-\lfloor\frac{k}{2}\rfloor}^{k}(k-i)d(t,x_i)+\sum_{i=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-1)d(t,x_i).
$$

By decomposing $\sum_{i=1}^{k-1}(k-i)d(t,x_i)$ to three summations, noting that $(k-k)d(t,x_k)=0$, and using (5.5), we have

$$\sum_{i=1}^{k-1}(k-i)d(t,x_i) = \sum_{i=k-\lfloor\frac{k}{2}\rfloor}^{k}(k-i)d(t,x_i) + \sum_{i=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-1)d(t,x_i)$$
$$+\sum_{j=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(k-j-\lceil\frac{k}{2}\rceil+1)d(t,x_j) = A+\sum_{j=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lfloor\frac{k}{2}\rfloor-j+1)d(t,x_j)$$
$$\geq A+\sum_{j=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-j)d(t,x_j) = A+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(t,x_j).$$

Therefore, by the triangle inequality and the above statements, we have

$$\sum_{i=1}^{k-1}(k-i)d(t,x_i)+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(x_i,x_j) \geq A+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(t,x_j)$$
$$+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(x_i,x_j) = A+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}(d(t,x_j)+d(x_i,x_j))$$
$$\geq A+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}\sum_{j=1}^{i-\lfloor\frac{k}{2}\rfloor-1}d(t,x_i) = A+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}(i-\lfloor\frac{k}{2}\rfloor-1)d(t,x_i)$$
$$\geq A+\sum_{i=\lceil\frac{k}{2}\rceil+1}^{k}(i-\lfloor\frac{k}{2}\rfloor-1)d(t,x_i)+(\lceil\frac{k}{2}\rceil-\lfloor\frac{k}{2}\rfloor-1)d(t,x_{\lceil\frac{k}{2}\rceil})$$
$$= A+\sum_{i=\lceil\frac{k}{2}\rceil}^{k}(i-\lfloor\frac{k}{2}\rfloor-1)d(t,x_i) = \sum_{i=k-\lfloor\frac{k}{2}\rfloor}^{k}(k-i)d(t,x_i)+\sum_{i=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-1)d(t,x_i)$$
$$+\sum_{i=k-\lfloor\frac{k}{2}\rfloor}^{k}(i-\lfloor\frac{k}{2}\rfloor-1)d(t,x_i)$$
$$= \sum_{i=k-\lfloor\frac{k}{2}\rfloor}^{k}(k-i+i-\lfloor\frac{k}{2}\rfloor-1)d(t,x_i)+\sum_{i=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-1)d(t,x_i)$$
$$= \sum_{i=k-\lfloor\frac{k}{2}\rfloor}^{k}(\lceil\frac{k}{2}\rceil-1)d(t,x_i)+\sum_{i=1}^{k-\lfloor\frac{k}{2}\rfloor-1}(\lceil\frac{k}{2}\rceil-1)d(t,x_i) = \sum_{i=1}^{k}(\lceil\frac{k}{2}\rceil-1)d(t,x_i).$$

This yields the result. □

Our main result is that Algorithm 5.3 is a constant factor approximation algorithm.

We now proceed to bound the diversity part of the optimal solution. We re-use the key ideas from Aghamolaei et al. [4] to achieve this. Let O be an optimal solution for maximizing $f(S)$ subject to $S \subseteq U$ and $|S| = k$. Let $O^i = T^i \cap O$, $Q^i = O^i \setminus S^i$. So Q^i are the elements of O on machine I that were "missed" by S^i. Intuitively, we bound the damage to optimality by missing these elements by finding a low-weight matching between Q^i and S^i. The following normalization parameters are used in the next two lemmas: $r_i = \frac{f(S^i)}{\binom{k}{2}}$ and $r = \max_{i=1,\ldots,m} r_i$. Let $G^i(O^i \cup S^i, E)$ be a complete weighted graph. For $u, v \in O^i \cup S^i$, we use $d(u, v)$ as the edge weight in our matching problem.

Lemma 18. *There exists a bipartite matching between Q^i and S^i in G^i with a weight of at most $\frac{4.5}{2}|Q^i|r$ that covers all the Q^i.*

Proof. The number of all maximal bipartite matchings between Q^i and S^i is $\frac{k!}{(k-|Q^i|)!}$. Any of these matchings covers Q^i because $|Q^i| \leq |S^i|$. Each edge $\{q, x\}$ with $q \in Q^i$ and $x \in S^i$ is in $\frac{(k-1)!}{(k-|Q^i|)!}$ of these matchings. Hence the total weight of all matchings can be expressed as

$$\begin{aligned}
\frac{(k-1)!}{(k-|Q^i|)!} \sum_{q \in Q^i} \sum_{x \in S^i} d(q, x) &\leq \frac{(k-1)!}{(k-|Q^i|)!} \sum_{q \in Q^i} \frac{4.5}{k-1} f(S^i) \\
&\leq \frac{(k-1)!}{(k-|Q^i|)!} \sum_{q \in Q^i} \frac{4.5}{k-1} \binom{k}{2} r \\
&= \frac{(k-1)!}{(k-|Q^i|)!} |Q^i| \frac{4.5k}{2} r \\
&= \frac{k!}{(k-|Q^i|)!} \frac{4.5}{2} |Q^i| r
\end{aligned}$$

The first inequality is from Lemma 19 and the second by the definition of r. It follows that there exists a matching with a weight of at most $\frac{4.5}{2}|Q^i|r$. $\square$

We are now in position to upper bound the diversity portion of an optimal solution in terms of $f(\texttt{OPT}(\cup_i^m S^i))$.

Lemma 19. *Let ALG be Algorithm 5.1 and $S^i = \texttt{ALG}(T^i)$. Then $D(O) \leq 8.5 f(\texttt{OPT}(\cup_{i=1}^m S^i))$.*

Proof. Let M^i be the maximal bipartite matching between Q^i and S^i with a weight of less than or equal to $\frac{4.5}{2}|Q^i|r$. It exists because of Lemma 18. Let $M = \cup_{i=1}^m M^i$. Note that S_i's are disjoint and Q^i's are disjoint. This implies that M^i's are disjoint. Therefore, M is a matching between $\cup_{i=1}^m Q^i$ and $\cup_{i=1}^m S^i$ that covers all of $\cup_{i=1}^m Q^i$ with a weight of less than or equal to $\frac{4.5}{2} \sum_{i=1}^m |Q^i| r \leq \frac{4.5}{2}|O|r = \frac{4.5}{2} kr$.

Let $e : O \to \cup_{i=1}^m S^i$ be a mapping which maps any $o \in O \cap (\cup_{i=1}^m S^i)$ to itself and any $o \in (\cup_{i=1}^m Q^i)$ to its matched vertex in M. The weight of this mapping is less than or equal to the weight of M since $d(o, o) = 0$. Note that each vertex in the *range*(e) is mapped from at most two vertices in O. We use this fact in the second inequality below and use the triangle inequality in the first inequality. We have

$$\begin{aligned} D(O) &= \sum_{\{u,v\}\in O} d(u,v) \leq \sum_{\{u,v\}\in O} (d(u,e(u)) + d(e(u),e(v)) + d(e(v),v)) \\ &= (|O|-1)\sum_{u\in O} d(o,e(o)) + \sum_{\{u,v\}\in O} d(e(u),e(v)) \leq (k-1)\frac{4.5}{2}kr + 4D(range(e)) \\ &\leq 4.5\binom{k}{2}r + 4f(\mathtt{OPT}(\cup_{i=1}^m S^i)) \leq 8.5f(\mathtt{OPT}(\cup_{i=1}^m S^i)) \end{aligned}$$

□

Now, we proceed to bound $g(O)$ and the proofs of the next two lemmas follow those found in Mirrokni and Zadimoghaddam [61]. Let $o_1,\ldots,o_k$ be an ordering of elements of O. For $x = o_i \in O$ define $O_x = \{o_1,\ldots,o_{i-1}\}$ and $O_{o_1} = \emptyset$.

Lemma 20. *$g(O) \leq 6f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x) - \Delta(x,O_x\cup S^i))$.*

Proof. Note that $g(O) = g(O\cap(\cup_{i=1}^m S^i)) + \sum_{x\in O\setminus(\cup_{i=1}^m S^i)} \Delta(x, O_x \cup (O\cap(\cup_{i=1}^m S^i)))$. Therefore, using submodularity and monotonicity of g and 5-niceness of Algorithm 5.1, we have

$$\begin{aligned} g(O) &\leq f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{x\in O\setminus(\cup_{i=1}^m S^i)} \Delta(x,O_x) \\ &= f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i} (\Delta(x,O_x\cup S^i) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i)) \\ &\leq f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i} (\Delta(x,S^i) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i)) \\ &\leq f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i} (\frac{5}{k}f(S^i) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i)) \\ &\leq f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i} (\frac{5}{k}f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \Delta(x,O_x) - \Delta(x,O_x\cup S^i)) \\ &\leq f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + 5f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i} (\Delta(x,O_x) - \Delta(x,O_x\cup S^i)) \\ &\leq 6f(\mathtt{OPT}(\cup_{i=1}^m S^i)) + \sum_{i=1}^m \sum_{x\in O\cap T^i\setminus S^i} (\Delta(x,O_x) - \Delta(x,O_x\cup S^i)) \end{aligned}$$

□

In the next Lemma, we use the randomness of the partitioning of the data over machines and the first property of β-niceness.

Lemma 21. $\mathbb{E}[\sum_{i=1}^{m}\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x)-\Delta(x,O_x\cup S^i))]\leq \mathbb{E}[f(\mathit{OPT}(\cup_{i=1}^{m}S^i))]$.

Proof. We show that $\mathbb{E}[\sum_{i=1}^{m}\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x)-\Delta(x,O_x\cup S^i))]\leq \frac{\mathbb{E}[\sum_{i=1}^{m}g(S^i)]}{m}$ and the statement of the lemma follows from the fact that $\frac{\sum_{i=1}^{m}g(S^i)}{m}\leq f(\mathtt{OPT}(\cup_{i=1}^{m}S^i))$. We first establish an inequality

$$A:=\mathbb{E}[\sum_{i=1}^{m}\sum_{x\in O\cap T^i\setminus S^i}(\Delta(x,O_x)-\Delta(x,O_x\cup S^i))]\leq\frac{1}{m}B$$

where

$$B:=\mathbb{E}[\sum_{i=1}^{m}\sum_{x\in O}(\Delta(x,O_x)-\Delta(x,O_x\cup S^i))].$$

Let $\mathtt{ALG}$ be Algorithm 5.1. For $T\subseteq U$ and $x\in U$, let $q(x,T)=\Delta(x,O_x)-\Delta(x,O_x\cup\mathtt{ALG}(T))$. Let $P[.]$ be the probability mass function for the uniform distribution over m-partitions $\mathbb{P}=(T^1,\ldots,T^m)$ of U, and let $\mathbb{1}[x\notin\mathtt{ALG}(T\cup\{x\})]$ be a $0,1$ indicator function. Note that

$$P[T^i=T]=(\frac{1}{m})^{|T|}(1-\frac{1}{m})^{|U|-|T|}$$
$$P[T^i=T\cup\{x\}]=(\frac{1}{m})^{|T|+1}(1-\frac{1}{m})^{|U|-|T|-1}$$

Therefore

$$P[T^i=T\cup\{x\}]=\frac{P[T^i=T]+P[T^i=T\cup\{x\}]}{m}. \tag{5.6}$$

We have that

$$A=\sum_{i=1}^{m}\sum_{x\in O}\sum_{T\subseteq U\setminus\{x\}}P[T^i=T\cup\{x\}]\mathbb{1}[x\notin\mathtt{ALG}(T\cup\{x\})]q(x,T\cup\{x\})$$

$$\begin{aligned}B&=\sum_{i=1}^{m}\sum_{x\in O}\sum_{T\subseteq U\setminus\{x\}}(P[T^i=T\cup\{x\}]q(x,T\cup\{x\})+P[T^i=T]q(x,T))\\&\geq\sum_{i=1}^{m}\sum_{x\in O}\sum_{T\subseteq U\setminus\{x\}}\mathbb{1}[x\notin\mathtt{ALG}(T\cup\{x\})]q(x,T\cup\{x\})(P[T^i=T\cup\{x\}]\\&\qquad+P[T^i=T]).\end{aligned}$$

The last inequality holds because $q(.,.)$ is a non-negative function and multiplying it by $\mathbb{1}[x\notin\mathtt{ALG}(T\cup\{x\})]$ can only decrease the sum value. Also, $q(x,T)$ is replaced by $q(x,T\cup\{x\})$. It does not change the sum value because when $\mathbb{1}[x\notin\mathtt{ALG}(T\cup\{x\})]=1$, $q(x,T)=q(x,T\cup\{x\})$. We now deduce $A\leq B/m$ from (5.6).

Now note that $\sum_{x\in O}\Delta(x,O_x\cup S^i)=g(O\cup S^i)-g(S^i)$, and $\sum_{x\in O}\Delta(x,O_x)=g(O)$. Therefore, because of the monotonicity of g, we have for any i

$$\begin{aligned}&\sum_{x\in O}\Delta(x,O_x)-\Delta(x,O_x\cup S^i)\\&\qquad=g(O)-g(O\cup S^i)+g(S^i)\leq g(S^i).\end{aligned}$$

Hence $B \leq \frac{\mathbb{E}[\sum_{i=1}^{m} g(S^i)]}{m}$ and the lemma follows. $\square$

We now have the following follows directly from Lemmas 20, and 21.

Lemma 22. *Let* ALG *be Algorithm 5.1 and* $S^i = \texttt{ALG}(T^i)$. *Then* $g(O) \leq 6f(\texttt{OPT}(\cup_{i=1}^{m} S^i)) + \mathbb{E}[f(\texttt{OPT}(\cup_{i=1}^{m} S^i))]$.

Now using Lemmas 19 and 20, we can prove the following theorem.

Theorem 20. *Let* $k \geq 10$. *Algorithm 5.3 gives a* $\frac{1}{31}$*-approximate solution in expectation for maximizing* $f(S)$ *subject to* $|S| = k$.

Proof. Lemma 19 and 22 immediately yield $f(O) \leq 15.5\mathbb{E}[f(\texttt{OPT}(\cup_{i=1}^{m} S^i))]$. Based on Borodin et al. [14], we know that Algorithm 5.2 is a half approximation algorithm for maximizing f. Therefore, if ALG' is Algorithm 5.2 then $f(\texttt{OPT}(\cup_{i=1}^{m} S^i)) \leq 2f(\texttt{ALG'}(\cup_{i=1}^{m} S^i))$. Hence $f(O) \leq 31\mathbb{E}[f(\texttt{ALG'}(\cup_{i=1}^{m} S^i))]$ which is exactly the statement of the theorem. $\square$

In the next section, we evaluate the empirical performance of our algorithm for the distributed multi-label feature selection problem.

5.4 Empirical Results for Distributed Multi-Label Feature Selection

In this section, we investigate the performance of our method in practice. In the first experiment, we compare our distributed method with centralized multi-label feature selection methods in the literature on a classification task. We show that our method's performance is comparable to, or in some cases is even better than previous centralized methods. Next, we compare our distributed and centralized methods on two large datasets. We show that the distributed algorithm achieves almost the same objective function value and it is much faster. This implies that the distributed algorithm achieves a better approximation in practice compared to the theoretical guarantee.

Comparison to Centralized Methods

As mentioned in Section 5.1, most of the multi-label feature selection methods convert the multi-label dataset to one or multiple single-label datasets and then use single-label feature selection methods and then aggregate the results. Binary relevance (BR) and label powerset (LP) are the two best known of these conversions. Here, we combine these two conversion methods with two single-label feature selection methods which results in four different centralized feature selection methods. We considered ReliefF (RF) [50, 69] and information gain (IG) [86] for single-label methods. These methods compute a score for each feature and for aggregating their results in Binary Relevance conversion, it is enough to calculate the sum of the scores of each feature and use these scores for selecting features. These methods are used before in the literature for multi-label feature selection [23, 27, 72–74].

For comparison, we selected 10 to 100 features with each method and did a multi-label classification using BRKNN-b proposed in Xioufis et al. [82]. We did a 10-fold cross validation with five neighbors for BRKNN-b. We evaluated the classification outputs over five multi-label evaluation measures. They are subset accuracy, example-based accuracy, example-based F-measure, micro-averaged F-measure, and macro-averaged F-measure [45, 74].

Let n be the number of samples in the dataset, L_i be the set of labels for sample i that are 1 in the dataset, and L'_i be the set of labels for sample i that we predicted to be 1. Then the *subset accuracy* of the learning method is equal to

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(L_i, L'_i)$$

where $\mathbb{I}(,.,)$ is a 0, 1 indicator function and is equal to 1 when set L_i is equal to the set L'_i, and it is 0 otherwise. *Example-based accuracy* is equal to

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|L_i \cap L'_i|}{|L_i \cup L'_i|}.$$

Example-based F-measure is equal to

$$\frac{1}{n}\sum_{i=1}^{n}\frac{2|L_i \cap L'_i|}{|L_i| + |L'_i|}.$$

These evaluation measures are example-based. Micro-averaged F-measure and Macro-averaged F-measure are two label-based measures for multi-label classification. Let t be the number of labels in the dataset, E_i be the set of examples that their i'th label is equal to 1, and E'_i be the set of example that we predicted their i'th labels to be 1. Then *Micro-averaged F-measure* is equal to

$$\frac{1}{t}\sum_{i=1}^{t}\frac{2|E_i \cap E'_i|}{|E_i| + |E'_i|}.$$

Macro-averaged F-measure is equal to

$$\frac{2\sum_{i=1}^{t}|E_i \cap E'_i|}{\sum_{i=1}^{t}|E_i| + \sum_{i=1}^{t}|E'_i|}.$$

We used the Mulan library for the classification and computation of the evaluation measures [79]. We used a synthesized dataset and five real-world datasets. Their specifications are shown in table 5.1. The synthesized dataset made up of eight labels. Each label has two original features that repeated 50 times. One of the features has the same value as its label in half of the samples, and the other one has the same value as its label in a quarter of the samples. The results of this dataset show that our method outperforms other methods on a dataset with redundant features. The results of this experiments are shown in Figures 5.2 and 5.3. We named our method distributed greedy diversity plus submodular (DGDS) in the plots. The other methods are named based on the conversion method they use (i.e., BR or LP) and the feature selection method they use (i.e., RF or IG). In the experiments, we used $\lambda = 0.5$ and $\max^{10}$ for our method. Results of the distributed method fluctuate more compared to

Table 5.1: Specifications of the datasets.

Dataset Name	# Features	# Instances	# Labels	Reference
Synthesized	800	256	8	-
Corel5k	499	5000	374	[30]
Eurlex-ev	5000	19,348	3993	[32]
CAL500	68	502	174	[80]
Delicious	500	16,105	983	[78]
Scene	294	2407	6	[15]

Table 5.2: Comparison of the distributed and the centralized algorithms. "h" and "m" means hour and minute.

Dataset Name	Reference	# Features	# Instances	# Labels	# Selected Features	# Machines	Distributed Algorithm Objective Value	Centralized Algorithm Objective Value	Distributed Algorithm Runtime	Centralized Algorithm Runtime	Speed-up
RCV1V2	[56]	47,236	6000	101	10	69	22.7	22.6	2.8m	1h 33m	33.2
					50	31	618.7	616.4	10.8m	2h 30.0m	15.1
					100	22	2468.2	2490.7	20.3m	3h 39m	10.8
					200	16	9338.7	10,016.0	47.0m	6h 16.8m	8.0
TMC2007	[75]	49,060	28,596	22	10	71	22.8	22.6	4.6m	2h 32.5m	33.4
					50	32	620.0	615.6	24.2m	6h 24.7m	15.9
					100	23	2510.0	2487.7	59.5m	11h 6.2m	11.2
					200	16	10,104.3	10,001.4	2h 41.3m	20h 49.8m	7.7

other methods. The reason is that, for every number of features, we did the feature selection, including the random partitioning, from scratch. This caused more variation in its results but also showed that the method is relatively stable and does not produce poor quality results for different random partitionings.

As discussed, we compared our method to centralized feature selection methods because there is no distributed multi-label feature selection method prior to our work. We should note that this comparison is unfair to the distributed method because it uses much less of the data compared to centralized methods. For example, it does not use the relation (or the distance) between the features in different machines. The advantage of the distributed method is that it is much faster and scalable.

Comparison of Distributed and Centralized Algorithms

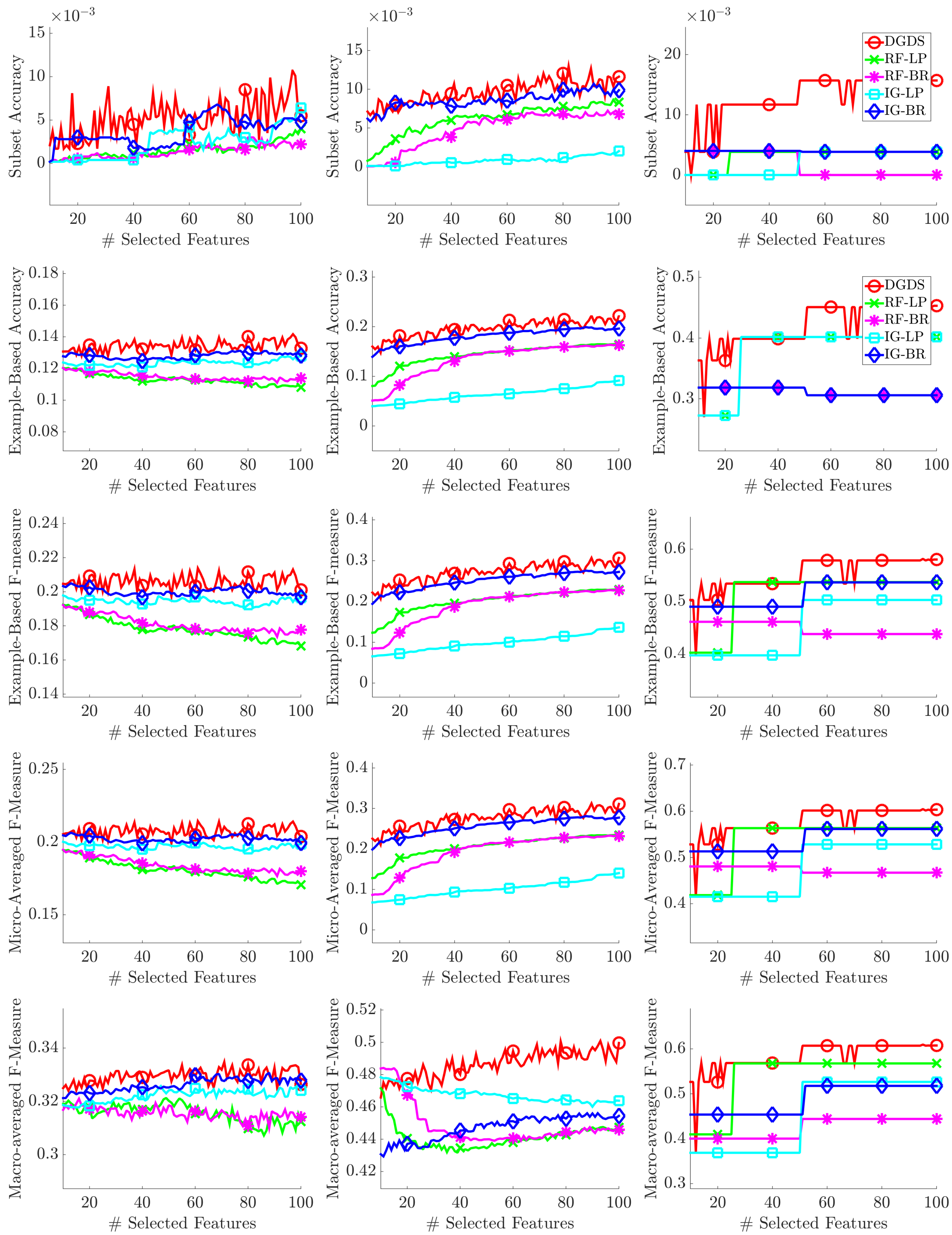
Here, we compare the performance of our proposed algorithm (Algorithm 5.3) with the centralized algorithm introduced in Borodin et al. [14] (Algorithm 5.2) on the optimization task. We compare the runtime and the value of the objective function the algorithms achieve. We select 10, 50, 100, and 200 features on two large datasets. If there are d' features in a machine, and we want to select k of them then the runtime of the machine is $\mathcal{O}(d'k)$. Therefore, if we have $\lceil\sqrt{d/k}\rceil$ slave machines then each of them has $\mathcal{O}(\sqrt{dk})$ features and its runtime is equal to $\mathcal{O}(k\sqrt{dk})$, where d is the total number of features. Also, the master machine will have $\mathcal{O}(\sqrt{dk})$ features, and its runtime is $\mathcal{O}(k\sqrt{dk})$ which means the runtime complexity of the master machine and the slave machines are equal. If we increase or decrease the number of slave machines, then the running time of the master machine or the slave machines will increase which results in a lower speed-up. Hence, we set the number of slave machines equal to $\lceil\sqrt{d/k}\rceil$. The results show that in practice our proposed distributed algorithm achieves an approximate solution as good as the centralized algorithm in a much shorter time. The results are summarized in Table 5.2. Moreover, we compared the distributed and the centralized algorithms on the classification task. Results of this experiment are shown in Figure 5.4.

Effect of λ hyper-parameter

To show the importance of both terms of the objective function, redundancy (diversity function) and relevance (submodular function), we compared the performance of the method for different λ value. We select 20, 30, 40, and 50 features on the scene dataset [15]. As shown in Figure 5.5, the best performance happens for some λ between 0 and 1. This shows that both terms are necessary and it is possible to get better results by choosing λ carefully.

5.5 Future Work

In this chapter, we presented a greedy algorithm for maximizing the sum of a sum-sum diversity function and a non-negative, monotone, submodular function subject to a cardinality constraint in distributed and streaming settings. We showed that this algorithm guarantees a provable theoretical approximation. Moreover, we formulated the multi-label feature selection problem as such an optimization problem and developed a multi-label feature selection method for distributed and streaming settings that can handle the redundancy of the features. Improving the theoretical approximation guarantee is appealing for future work. From the empirical standpoint, it would be nice to try other metric distances and other submodular functions for the multi-label feature selection problem.



(a) Corel5k (b) Eurlex-ev (c) Synthesized

Figure 5.2: Comparison of proposed distributed method with centralized methods in the literature (1).

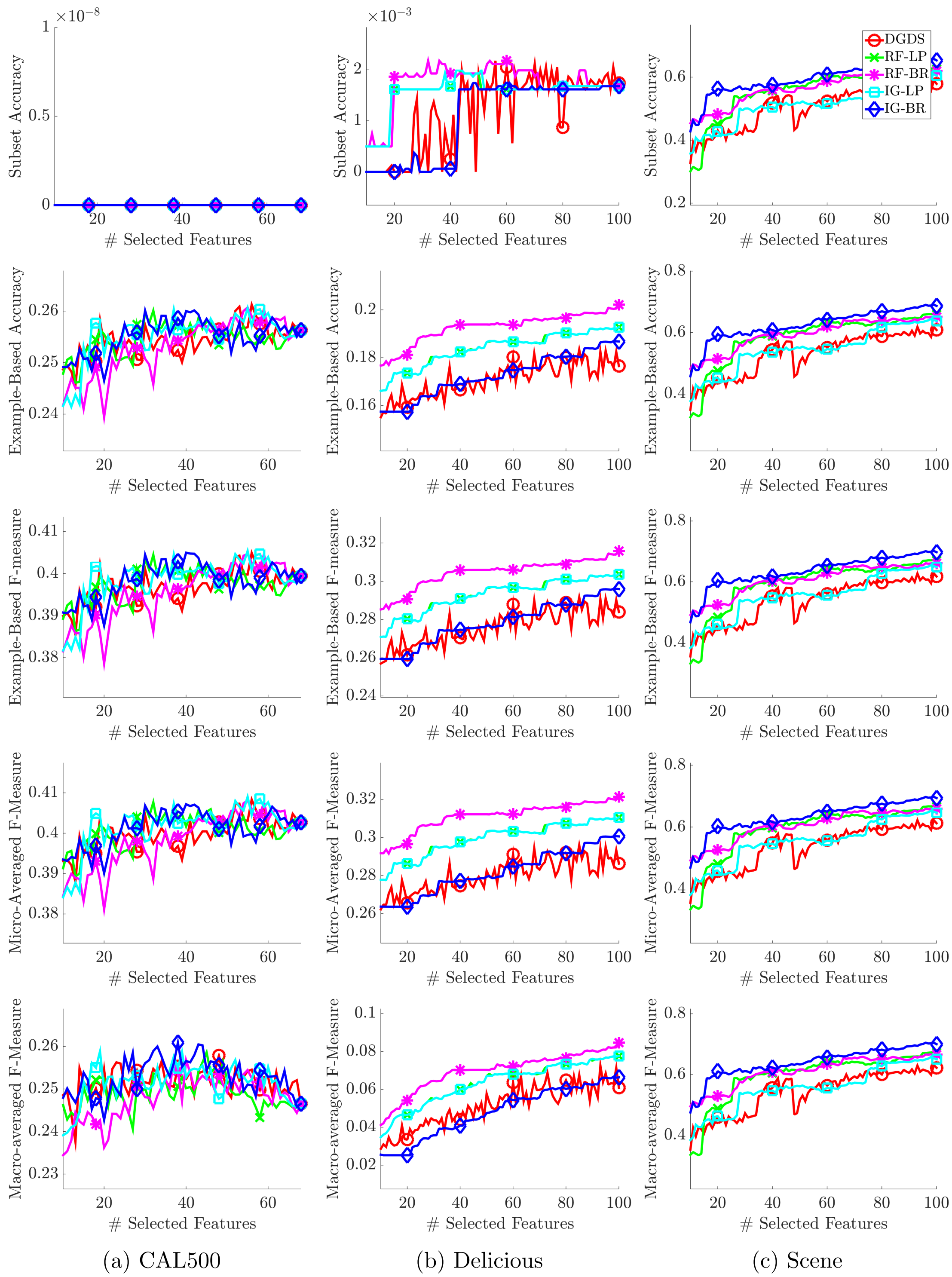
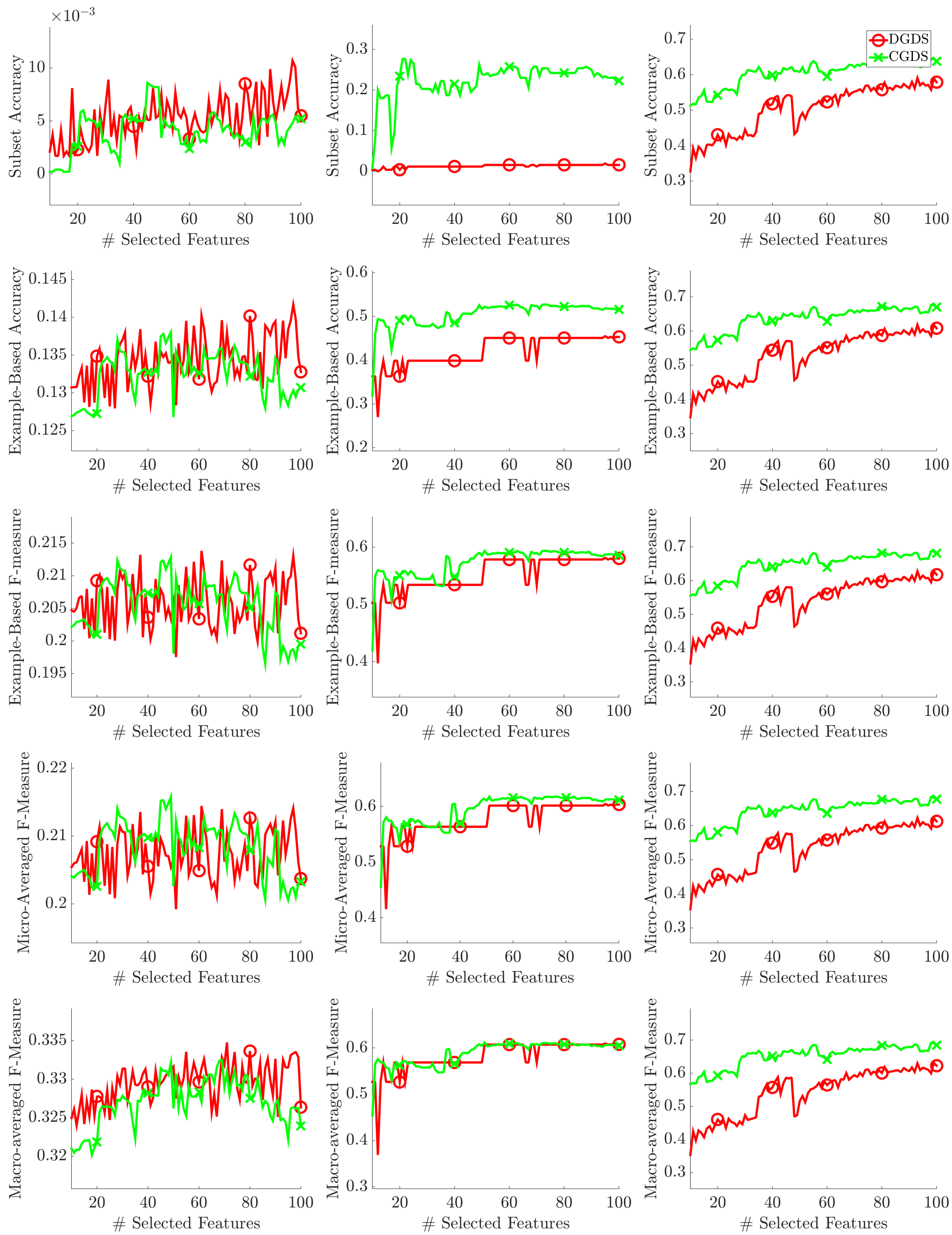


Figure 5.3: Comparison of proposed distributed method with centralized methods in the literature (2).



(a) Corel5k (b) Synthesized (c) Scene

Figure 5.4: Comparison of proposed distributed method (DGDS) with proposed centralized method (CGDS) on the classification task.

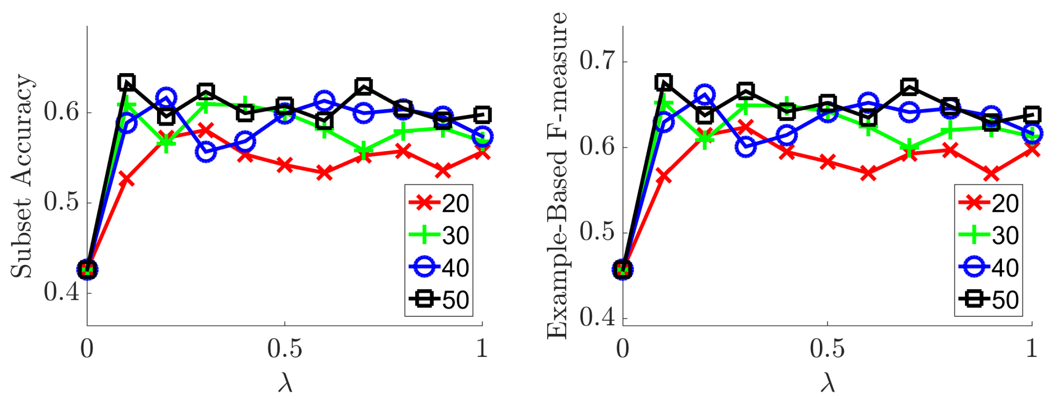


Figure 5.5: Effect of λ on the performance of the method.

Bibliography

[1] Zeinab Abbassi, Vahab S. Mirrokni, and Mayur Thakur. Diversity maximization under matroid constraints. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 32–40, 2013. doi: 10.1145/2487575.2487636. URL `http://doi.acm.org/10.1145/2487575.2487636`.

[2] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

[3] Alexander A. Ageev and Maxim I. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

[4] Sepideh Aghamolaei, Majid Farhadi, and Hamid Zarrabi-Zadeh. Diversity maximization via composable coresets. In *Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG 2015, Kingston, Ontario, Canada, August 10-12, 2015*, 2015. URL `http://research.cs.queensu.ca/cccg2015/CCCG15-papers/47.pdf`.

[5] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second International Conference on Web Search and Web Data Mining, WSDM 2009, Barcelona, Spain, February 9-11, 2009*, pages 5–14, 2009. doi: 10.1145/1498759.1498766. URL `http://doi.acm.org/10.1145/1498759.1498766`.

[6] Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. Inapproximability of densest κ-subgraph from average case hardness. *Unpublished manuscript*, 1, 2011.

[7] RafaeldaPonte Barbosa, Alina Ene, Huy L. Nguyen, and Justin Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 1236–1244, 2015. URL `http://jmlr.org/proceedings/papers/v37/barbosa15.html`.

[8] Aditya Bhaskara, Mehrdad Ghadiri, Vahab S. Mirrokni, and Ola Svensson. Linear relaxations for finding diverse elements in metric spaces. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4098–4106, 2016. URL `http://papers.nips.cc/paper/6500-linear-relaxations-for-finding-diverse-elements-in-metric-spaces`.

[9] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. Recent advances and emerging challenges of feature selection in the context of big data. *Knowl.-Based Syst.*, 86:33–45, 2015. doi: 10.1016/j.knosys.2015.05.014. URL `https://doi.org/10.1016/j.knosys.2015.05.014`.

[10] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. Recent advances and emerging challenges of feature selection in the context of big data. *Knowl.-Based Syst.*, 86:33–45, 2015. doi: 10.1016/j.knosys.2015.05.014. URL `http://dx.doi.org/10.1016/j.knosys.2015.05.014`.

[11] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2012, Scottsdale, AZ, USA, May 20-24, 2012*, pages 155–166, 2012. doi: 10.1145/2213556.2213580. URL `http://doi.acm.org/10.1145/2213556.2213580`.

[12] Allan Borodin, Dai Le, and Yuli Ye. Weakly submodular functions. *CoRR*, abs/1401.6697, 2014. URL `http://arxiv.org/abs/1401.6697`.

[13] Allan Borodin, Dai Le, and Yuli Ye. A proportionally submodular functions. *http://www.cs.toronto.edu/ bor/Papers/proportional-talg-submit.pdf*, 2015. URL `http://www.cs.toronto.edu/~bor/Papers/proportional-talg-submit.pdf`.

[14] Allan Borodin, Aadhar Jain, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Trans. Algorithms*, 13(3):41:1–41:25, 2017. doi: 10.1145/3086464. URL `http://doi.acm.org/10.1145/3086464`.

[15] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004. doi: 10.1016/j.patcog.2004.03.009. URL `https://doi.org/10.1016/j.patcog.2004.03.009`.

[16] Yuri Y. Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.

[17] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 182–196. Springer, 2007.

[18] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40 (6):1740–1766, 2011.

[19] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. Max-sum diversity via convex programming. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, pages 26:1–26:14, 2016. doi: 10.4230/LIPIcs.SoCG.2016.26. URL `https://doi.org/10.4230/LIPIcs.SoCG.2016.26`.

[20] Alfonso Cevallos, Friedrich Eisenbrand, and Rico Zenklusen. Local search for max-sum diversification. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 130–142, 2017. doi: 10.1137/1.9781611974782.9. URL https://doi.org/10.1137/1.9781611974782.9.

[21] Barun Chandra and Magnús M. Halldórsson. Facility dispersion and remote subgraphs. In *Algorithm Theory - SWAT '96, 5th Scandinavian Workshop on Algorithm Theory, Reykjavík, Iceland, July 3-5, 1996, Proceedings*, pages 53–65, 1996. doi: 10.1007/3-540-61422-2_120. URL https://doi.org/10.1007/3-540-61422-2_120.

[22] 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. doi: 10.1109/FOCS.2010.60. URL https://doi.org/10.1109/FOCS.2010.60.

[23] Weizhu Chen, Jun Yan, Benyu Zhang, Zheng Chen, and Qiang Yang. Document transformation for multi-label feature selection in text categorization. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, pages 451–456, 2007. doi: 10.1109/ICDM.2007.18. URL https://doi.org/10.1109/ICDM.2007.18.

[24] Anirban Dasgupta, Ravi Kumar, and Sujith Ravi. Summarization through submodularity and dispersion. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1014–1022, 2013. URL http://aclweb.org/anthology/P/P13/P13-1100.pdf.

[25] Etienne De Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2):111–125, 2008.

[26] Elena Demidova, Peter Fankhauser, Xuan Zhou, and Wolfgang Nejdl. *DivQ*: diversification for keyword search over structured databases. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 331–338, 2010. doi: 10.1145/1835449.1835506. URL http://doi.acm.org/10.1145/1835449.1835506.

[27] Sareewan Dendamrongvit, Peerapon Vateekul, and Miroslav Kubat. Irrelevant attributes and imbalanced classes in multi-label text-categorization domains. *Intelligent Data Analysis*, 15(6):843–859, 2011.

[28] Debadeepta Dey, Tian Yu Liu, Martial Hebert, and J Andrew Bagnell. Contextual sequence prediction with application to control library optimization. 2012.

[29] Marina Drosou and Evaggelia Pitoura. Search result diversification. *SIGMOD Record*, 39(1):41–47, 2010. doi: 10.1145/1860702.1860709. URL http://doi.acm.org/10.1145/1860702.1860709.

[30] Pinar Duygulu, Kobus Barnard, João F. G. de Freitas, and David A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part IV*, pages 97–112, 2002. doi: 10.1007/3-540-47979-1_7. URL `https://doi.org/10.1007/3-540-47979-1_7`.

[31] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. *An analysis of approximations for maximizing submodular set functions-II*. Springer, 1978.

[32] Enrico Francesconi, Simonetta Montemagni, Wim Peters, and Daniela Tiscornia. *Semantic processing of legal texts: Where the language of law meets the law of language*, volume 6036. Springer, 2010.

[33] Mehrdad Ghadiri and Mark Schmidt. Distributed maximization of "submodular plus diversity" functions for multi-label feature selection on huge datasets. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, pages 2077–2086, 2019. URL `http://proceedings.mlr.press/v89/ghadiri19a.html`.

[34] Mehrdad Ghadiri, Richard Santiago, and F. Bruce Shepherd. Beyond submodular maximization. *CoRR*, abs/1904.09216, 2019. URL `http://arxiv.org/abs/1904.09216`.

[35] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. doi: 10.1007/BF02579273. URL `https://doi.org/10.1007/BF02579273`.

[36] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[37] Peter M Gruber and Jörg M Wills. *Handbook of convex geometry*, volume 2. North Holland, 1993.

[38] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. URL `http://www.jmlr.org/papers/v3/guyon03a.html`.

[39] Refael Hassin, Shlomi Rubinstein, and Arie Tamir. Notes on dispersion problems. *Unpublished manuscript*, 1994.

[40] Refael Hassin, Shlomi Rubinstein, and Arie Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 1997.

[41] Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 100–108, 2014. doi: 10.1145/2594538.2594560. URL `http://doi.acm.org/10.1145/2594538.2594560`.

[42] Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1897–1904. IEEE, 2011.

[43] Mark Jerrum. Large cliques elude the metropolis process. *Random Struct. Algorithms*, 3 (4):347–360, 1992. doi: 10.1002/rsa.3240030402. URL `https://doi.org/10.1002/rsa.3240030402`.

[44] Richard M. Karp. The probabilistic analysis of some combinatorial search problems, algorithms and complexity (proc. sympos., carnegie-mellon univ., pittsburgh, pa. 1976), 1-19, 1976.

[45] Shima Kashef, Hossein Nezamabadi-pour, and Bahareh Nikpour. Multilabel feature selection: A comprehensive review and guiding experiments. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 8(2), 2018. doi: 10.1002/widm.1240. URL `https://doi.org/10.1002/widm.1240`.

[46] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[47] Rajiv Khanna, Ethan R. Elenberg, Alexandros G. Dimakis, Sahand Negahban, and Joydeep Ghosh. Scalable greedy feature selection via weak submodularity. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 1560–1568, 2017. URL `http://proceedings.mlr.press/v54/khanna17b.html`.

[48] Jon M. Kleinberg, Christos H. Papadimitriou, and Prabhakar Raghavan. Segmentation problems. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 473–482, 1998. doi: 10.1145/276698.276860. URL `http://doi.acm.org/10.1145/276698.276860`.

[49] Pushmeet Kohli, M. Pawan Kumar, and Philip H. S. Torr. P^3 & beyond: Move making algorithms for solving higher order functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1645–1656, 2009.

[50] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *Machine Learning: ECML-94, European Conference on Machine Learning, Catania, Italy, April 6-8, 1994, Proceedings*, pages 171–182, 1994. doi: 10.1007/3-540-57868-4_57. URL `https://doi.org/10.1007/3-540-57868-4_57`.

[51] Guy Kortsarz and David Peleg. On choosing a dense subgraph (extended abstract). In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 692–701, 1993. doi: 10.1109/SFCS.1993.366818. URL `https://doi.org/10.1109/SFCS.1993.366818`.

[52] Nitish Korula, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 889–898, 2015. doi: 10.1145/2746539.2746626. URL `http://doi.acm.org/10.1145/2746539.2746626`.

[53] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*, pages 71–104. 2014. doi: 10.1017/CBO9781139177801.004. URL `https://doi.org/10.1017/CBO9781139177801.004`.

[54] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654, 2007.

[55] Andreas Krause and Carlos Guestrin. Beyond convexity: Submodularity in machine learning. *ICML Tutorials*, 2008.

[56] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004. URL `http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lewis04a.pdf`.

[57] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

[58] Yuzong Liu, Kai Wei, Katrin Kirchhoff, Yisong Song, and Jeff Bilmes. Submodular feature selection for high-dimensional acoustic score spaces. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 7184–7188. IEEE, 2013.

[59] Ziyang Liu, Peng Sun, and Yi Chen. Structured search result differentiation. *PVLDB*, 2 (1):313–324, 2009. URL `http://www.vldb.org/pvldb/2/vldb09-500.pdf`.

[60] László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art, XIth International Symposium on Mathematical Programming, Bonn, Germany, August 23-27, 1982.*, pages 235–257, 1982. doi: 10.1007/978-3-642-68874-4_10. URL `https://doi.org/10.1007/978-3-642-68874-4_10`.

[61] Vahab S. Mirrokni and Morteza Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 153–162, 2015. doi: 10.1145/2746539.2746624. URL `http://doi.acm.org/10.1145/2746539.2746624`.

[62] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. *Journal of Machine Learning Research*, 17:238:1–238:44, 2016. URL `http://jmlr.org/papers/v17/mirzasoleiman16a.html`.

[63] George L. Nemhauser and Leonard A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

[64] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 14 (1):265–294, 1978.

[65] Xuan Vinh Nguyen, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010. URL `http://portal.acm.org/citation.cfm?id=1953024`.

[66] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005. doi: 10.1109/TPAMI.2005.159. URL `http://dx.doi.org/10.1109/TPAMI.2005.159`.

[67] Adarsh Prasad, Stefanie Jegelka, and Dhruv Batra. Submodular meets structured: Finding diverse subsets in exponentially-large structured item sets. In *Advances in Neural Information Processing Systems*, pages 2645–2653, 2014.

[68] S. S. Ravi, Daniel J. Rosenkrantz, and Giri Kumar Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994. doi: 10.1287/opre.42.2.299. URL `https://doi.org/10.1287/opre.42.2.299`.

[69] Marko Robnik-Sikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53(1-2):23–69, 2003. doi: 10.1023/A:1025667309714. URL `https://doi.org/10.1023/A:1025667309714`.

[70] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[71] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

[72] Newton Spolaôr, Everton Alvares Cherman, and Maria Carolina Monard. Using relieff for multi-label feature selection. In *Conferencia Latinoamericana de Informática*, pages 960–975, 2011.

[73] Newton Spolaôr, Everton Alvares Cherman, Maria Carolina Monard, and Huei Diana Lee. Filter approach feature selection methods to support multi-label learning based on relieff and information gain. In *Advances in Artificial Intelligence - SBIA 2012 - 21th Brazilian Symposium on Artificial Intelligence, Curitiba, Brazil, October 20-25, 2012. Proceedings*, pages 72–81, 2012. doi: 10.1007/978-3-642-34459-6_8. URL `https://doi.org/10.1007/978-3-642-34459-6_8`.

[74] Newton Spolaôr, Everton Alvares Cherman, Maria Carolina Monard, and Huei Diana Lee. A comparison of multi-label feature selection methods using the problem transformation approach. *Electr. Notes Theor. Comput. Sci.*, 292:135–151, 2013. doi: 10.1016/j.entcs.2013.02.010. URL `https://doi.org/10.1016/j.entcs.2013.02.010`.

[75] Ashok N. Srivastava and Brett Zane-Ulman. Discovering recurring anomalies in text reports regarding complex space systems. In *Aerospace conference, 2005 IEEE*, pages 3853–3862. IEEE, 2005.

[76] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 1577–1584, 2009.

[77] Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

[78] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08)*, volume 21, pages 53–59. sn, 2008.

[79] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12: 2411–2414, 2011.

[80] Douglas Turnbull, Luke Barrington, David A. Torres, and Gert R. G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio, Speech & Language Processing*, 16(2):467–476, 2008. doi: 10.1109/TASL.2007.913750. URL `https://doi.org/10.1109/TASL.2007.913750`.

[81] Jan 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*, pages 67–74. ACM, 2008.

[82] Eleftherios Spyromitros Xioufis, Grigorios Tsoumakas, and Ioannis P. Vlahavas. An empirical study of lazy multilabel classification algorithms. In *Artificial Intelligence: Theories, Models and Applications, 5th Hellenic Conference on AI, SETN 2008, Syros, Greece, October 2-4, 2008. Proceedings*, pages 401–406, 2008. doi: 10.1007/978-3-540-87881-0_40. URL `https://doi.org/10.1007/978-3-540-87881-0_40`.

[83] Cong Yu, Laks V. S. Lakshmanan, and Sihem Amer-Yahia. It takes variety to make a world: diversification in recommender systems. In *EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, March 24-26, 2009, Proceedings*, pages 368–378, 2009. doi: 10.1145/1516360.1516404. URL `http://doi.acm.org/10.1145/1516360.1516404`.

[84] Sepehr Abbasi Zadeh and Mehrdad Ghadiri. Max-sum diversification, monotone submodular functions and semi-metric spaces. *CoRR*, abs/1511.02402, 2015. URL `http://arxiv.org/abs/1511.02402`.

[85] Sepehr Abbasi Zadeh, Mehrdad Ghadiri, Vahab S. Mirrokni, and Morteza Zadimoghaddam. Scalable feature selection via distributed diversity maximization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 2876–2883, 2017. URL `http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14914`.

[86] Zheng Zhao, Fred Morstatter, Shashvata Sharma, Salem Alelyani, Aneeth Anand, and Huan Liu. Advancing feature selection research. *ASU feature selection repository*, pages 1–28, 2010.