博士論文

Approximate Submodularity in Machine Learning

（機械学習における近似的劣モジュラ性）

48-177208　藤井 海斗

指導教員　岩田 覚 教授

2020 年 3 月

東京大学大学院情報理工学系研究科数理情報学専攻

Abstract

Submodularity is a property that represents diminishing returns and appears ubiquitously in machine learning problems. By utilizing submodularity, many efficient algorithms with theoretical guarantees have been developed. However, there are still many problems that cannot be dealt with under the framework of submodularity. One promising approach to these problems is to extract properties close to submodularity, which we call *approximate submodularity*, and devise algorithms by extending existing results on submodularity. In this dissertation, we propose two new notions of approximate submodularity: *adaptive submodularity ratio* and *approximate submodularity for local search*. By utilizing these two notions, we develop efficient algorithms for various machine learning problems.

The first notion we propose is the adaptive submodularity ratio, which represents approximate submodularity in *adaptive optimization*. We are often confronted with a decision-making problem where the objective function is uncertain. In adaptive optimization, a decision-maker aims at gradually constructing a solution by repeating small decisions while gathering information on the objective function. To make a better solution, it is vital to perform *adaptively*, that is, to change the next decision according to the information obtained so far. It is known that if the objective function satisfies adaptive submodularity, which is an adaptive analog of submodularity, an adaptive greedy algorithm is guaranteed to work well. However, there are still many adaptive optimization problems that do not have adaptive submodularity. To analyze these problems, we propose the notion of adaptive submodularity ratio, which measures how close the objective function is to adaptive submodularity and provide a theoretical guarantee for the adaptive greedy algorithm in terms of this notion. We also apply a similar approach to the *batch-mode setting* of adaptive optimization, in which the decision-maker obtains information all at once after making multiple decisions. By extending the framework of adaptive submodularity ratio to the batch-mode setting, we provide theoretical guarantees for greedy-based algorithms.

The second notion we propose is approximate submodularity for *local search*. Local search is a well-known algorithm design technique for combinatorial optimization problems. Local search algorithms start with an initial solution and gradually increase the objective value by repeatedly moving the solution to a nearby point. First, we analyze local search algorithms for *feature selection*. Feature selection is the problem of selecting a significant subset out of a large number of features and a vital component of sparse regression, compressed sensing, and structure learning of graphical models. By utilizing approximate submodularity for local search, we analyze and accelerate local search algorithms for feature selection. Next, we tackle *dictionary selection*, which can be regarded as a two-stage version of feature selection. A dictionary is a collection of patterns that make up signals in the real world. Dictionary selection is the problem of learning a dictionary suitable for the given dataset by selecting a subset of the union of ready-made dictionaries. Based on approximate submodularity for local search, we develop an efficient greedy algorithm Replacement OMP with theoretical guarantees.

Acknowledgments

First and foremost, I want to dedicate my greatest gratitude to my supervisor, Satoru Iwata. He always inspired me through his deep knowledge and constantly supported me in the past three years. I would like to thank him for giving me the freedom to choose my own research topics, providing wonderful ideas, and guiding me to complete this dissertation with encouragement.

I am deeply thankful to my collaborators. I would like to thank Tasuku Soma for his vast knowledge and friendship. It was my great pleasure to have a desk next to him during my PhD years. I am deeply grateful to Shinsaku Sakaue for his effort. Without his constant encouragement, we would not be able to complete our collaboration. I would like to thank Yuichi Yoshida for his kind instructions. I learned the right mind-set toward research from his overwhelming productivity and vigorous passion.

I am sincerely grateful to all the members of Mathematical Informatics 7th Laboratory in the University of Tokyo. In particular, I would like to thank Shin-ichi Tanigawa for always providing constructive advices to my research. Also, I thank past and current PhD students and postdocs in the lab, Katie Clinch, Ayumi Igarashi, Kota Ishihara, Naoki Ito, Shinji Ito, Tatsuya Matsuoka, Taihei Oki, Nobutaka Shimizu, Yutaro Yamaguchi, and Yu Yokoi. I was happy to have technical discussions and comfortable chats with them. I am also grateful to Shuichi Hirahara, Yuni Iwamasa, Shuichi Katsumata, Takeru Matsuda, Yuji Nakatasukasa, and Shun Sato for inspiring discussions. My special thanks go to Erika Hiruma for her immense administrative support.

My research stay at ETH Zürich for three months was so fruitful. I appreciate Andreas Krause for welcoming me kindly and sparing much time for making discussions with me. I thank all the members of the Learning and Adaptive Systems Group in ETH Zürich. Their warm welcome made my research stay extremely enjoyable.

I also thank the dissertation committee members, Hiroshi Hirai, Ken'ichiro Tanaka, and Kenji Yamanishi for providing me valuable feedback.

I would like to appreciate the financial support by JSPS Research Fellowship for Young Scientists, JST ERATO, and JST CREST. Finally I would like to express my sincere gratitude to my family and friends for their unstinting support and encouragement.

Contents

1 Introduction

Data are produced at tremendous speed in every field and machine learning techniques become increasingly popular for decision-making based on data. When trying to apply machine learning techniques to massive datasets, we are often faced with computational issues.

For example, suppose we want to apply regression analysis to a high-dimensional dataset. To obtain an interpretable model, we often try to reduce the dimension by selecting a subset of relevant features. However, there are exponentially many possible subsets of features and it takes considerable computational time to check all these subsets. To find a fairly good subset of features in realistic computational time, we need to devise an efficient algorithm.

Another example is active learning. Though a dataset of labeled data points is necessary for supervised classification, it often takes much cost to obtain labels for unlabeled data points. In such a scenario, we should select a small subset of unlabeled data points to be labeled and apply a supervised classification algorithm by using this labeled subset as the training dataset. This subset selection problem also has exponentially many possible solutions and requires a careful search method.

Due to their intrinsic combinatorial nature, these problems are fit into the framework of combinatorial optimization. In the rich history of combinatorial optimization research, numerous techniques have been developed for finding a good solution out of exponentially many feasible ones in reasonable computational time. Based on these techniques, many researchers have devised efficient algorithms for combinatorial optimization problems in machine learning. In this dissertation, we develop algorithmic frameworks for combinatorial optimization problems in machine learning based on concepts called *approximate submodularity*. In the following, we provide a background of approximate submodularity and summarize our key contributions.

1.1 Submodularity in Machine Learning

The starting point of this dissertation is *submodularity* [Fujishige, 2005]. Submodularity is a property of functions that is central to combinatorial optimization and has been studied extensively for developing efficient algorithms and modeling real phenomena. There are various interpretations of the formal definition of submodularity, but here, we often view submodularity as a property of *diminishing returns*. Intuitively speaking, this property means that the value of an item decreases as the already obtained items increase. This property appears ubiquitously in the real world and has been used for modeling practical problems such as combinatorial auctions with substitutable goods [Lehmann et al., 2006] and facility location [Cornuejols et al., 1977].

For the last 15 years, submodularity has attracted much attention as a strong tool to design algorithms for machine learning. An approach based on submodularity has advantages both in its flexibility for modeling real problems and its amenability for designing efficient algorithms.

Due to the versatility of submodularity, many machine learning problems have been formulated as a submodular optimization problem. For example, submodularity has been utilized for formalizing problems of extracting a small subset that has information as much as possible given a set of a large number of elements. This kind of problem appears in observation selection for Gaussian processes [Krause et al., 2008], document summarization [Lin and Bilmes, 2011], and making an interpretable summary of a dataset [Ribeiro et al., 2016]. In such a scenario, it is often the case that a part of information presented by

an element is also presented by another element. This implies that the value of adding a single element to the solution decreases as the elements already added to the solution become larger, which is exactly submodularity. Therefore, problems of selecting an informative set can be naturally formulated as a submodular optimization problem. Besides, submodularity has been utilized for viral marketing [Kempe et al., 2003] and analyzing the performance of algorithms for Bayesian optimization [Srinivas et al., 2010]. In this way, the application range of submodularity has been expanding.

On the other hand, rich theoretical techniques developed for submodularity enable us to design efficient algorithms. One of the most celebrated results is an approximation guarantee for the greedy algorithm. The greedy algorithm is a simple procedure that starts with the empty set and repeatedly adds the element that increases the objective value the most. It is widely used as a heuristic for obtaining a solution that is acceptable but not necessarily optimal in many real applications. Intuitively, if the objective function satisfies submodularity, the greedy algorithm returns a solution competitive with an optimal solution [Nemhauser et al., 1978]. Algorithms with such kinds of guarantees are called *approximation algorithms* and have been studied as a powerful approach to NP-hard problems. Starting with the result for the greedy algorithm for the simplest setting, a line of work on *submodular maximization* has provided approximation algorithms for various settings [Călinescu et al., 2011, Buchbinder et al., 2015]. Motivated by the growing size of datasets, the framework of submodular maximization has been extended to more practical computational settings such as the streaming setting [Badanidiyuru et al., 2014] and the distributed setting [Mirzasoleiman et al., 2016].

1.2 From Submodularity to Approximate Submodularity

As described in the last section, various combinatorial optimization problems in machine learning can be regarded as a submodular maximization problem. However, there are still many problems that deviate from the formulation of submodular maximization. Even if the problems do not have submodularity, they may have a property close to submodularity. In this dissertation, we call such a property *approximate submodularity*. Since submodularity appears in various fields of machine learning, we can expect approximate submodularity to appear frequently as well. Also, to design algorithms for problems with approximate submodularity, we can inherit a part of rich theoretical insights developed for submodular maximization.

The most prominent existing result on approximate submodularity is the work on feature selection for sparse linear regression by Das and Kempe [2011]. The problem of selecting a subset of a large number of features and applying linear regression by using the selected features is a combinatorial optimization problem fundamental in machine learning, but its mean squared error does not satisfy submodularity. To view this problem through the lens of approximate submodularity, Das and Kempe [2011] defined the *submodularity ratio*, which measures how close the objective function is to submodular functions and related the submodularity ratio to a spectral parameter of the design matrix. By utilizing the submodularity ratio, they provided theoretical guarantees for greedy algorithms.

The submodularity ratio is cleverly defined so that the greedy algorithm can be analyzed well. In the analysis of the greedy algorithm for submodular maximization, a certain property that is equivalent to submodularity is utilized. The submodularity ratio represents how much this property must be relaxed to be satisfied by the objective function. Hence the submodularity ratio is suitable for analyzing the greedy algorithm, but its scope is limited. To apply a similar approach to broader applications, we need other notions of approximate submodularity. For each application, by relaxing a property of submodular functions that is important in the analysis, we can define a notion of approximate submodularity suitable for the application.

The goal of this dissertation is to propose new concepts of approximate submodularity and apply

them to various applications in machine learning. In particular, we focus on two aspects of approximate submodularity. One is approximate submodularity in *adaptive optimization*. Adaptive optimization can model several essential problems in machine learning such as active learning. However, adaptive submodularity, which is a counterpart of submodularity in adaptive optimization, holds only in specific applications. To deal with broader applications, we need a notion of approximate submodularity for adaptive optimization. The other is approximate submodularity for *local search*. Local search is a well-known algorithm design technique for combinatorial optimization problems as well as greedy methods and also amenable to analyses based on submodularity. A new concept of approximate submodularity is expected to make it possible to analyze local search for wider applications.

1.3 Approximate Submodularity in Adaptive Optimization

Adaptive optimization is a framework for optimization problems where we must gather information and make decisions in parallel. Adaptive optimization was first proposed in theoretical computer science [Dean et al., 2008] and has been applied to problems in algorithmic game theory [Chen et al., 2009, Gupta and Nagarajan, 2013], but it also has significant applications in machine learning.

Due to its intrinsic suitability to adaptive optimization, active learning is one of the most important applications of adaptive optimization. As mentioned above, active learning is a problem of selecting a subset of unlabeled data points to be labeled. In the basic setting of active learning, we alternately repeat selecting the next unlabeled data point to be labeled and obtaining its label. For achieving high accuracy with a small number of labels, it is vital to select data points *adaptively*, that is, select the next unlabeled data point depending on the labels of the already selected data points. The framework of adaptive optimization is useful to formulate such an adaptive decision-making problem as an optimization problem.

Golovin and Krause [2011a] developed the most influential framework of adaptive optimization for machine learning problems based on *adaptive submodularity*. Adaptive submodularity is a generalization of submodularity to the setting of adaptive optimization, and represents the property of diminishing returns in a stochastic sense. Golovin and Krause [2011a] proved that the adaptive greedy algorithm, which is a natural extension of the greedy algorithm to the adaptive setting, is competitive with an optimal adaptive policy. This result implies that the simple greedy policy performs well if the objective function satisfies adaptive submodularity. Golovin and Krause [2011a] also showed that adaptive submodularity is satisfied by several problems, including active learning, adaptive influence maximization in the independent cascade model, and adaptive sensor placement. Since then, the framework of adaptive submodularity has been utilized for devising greedy algorithms with theoretical guarantees for various machine learning problems, including adaptive experimental design [Golovin et al., 2010], recommendation [Gabillon et al., 2013], touch-based localization in robotics [Javdani et al., 2014], and active object detection [Chen et al., 2014].

However, the applicability of the framework of adaptive submodularity has limitations. There are still many adaptive optimization problems that do not satisfy adaptive submodularity, including adaptive influence maximization in the triggering model. To handle more various problems from the perspective of adaptive submodularity, we define an approximate version of adaptive submodularity and apply it to several applications.

In Chapter 3, we propose a notion of *adaptive submodularity ratio*, which is an analog of the submodularity ratio in the adaptive setting. The adaptive submodularity ratio measures how close the objective function is to adaptive submodular functions. We provide a theoretical guarantee on the performance of the adaptive greedy algorithm in terms of the adaptive submodularity ratio. Intuitively, this result implies that if the objective function is close to adaptive submodular functions, the adaptive greedy

algorithm performs well.

We also provide another usage of the adaptive submodularity ratio about the difference between the adaptive and non-adaptive settings. One of the most important questions in adaptive optimization is how different adaptive and non-adaptive policies are. In the adaptive setting, we can perform better than in the non-adaptive setting, but their gap is generally difficult to evaluate. *The adaptivity gap* [Dean et al., 2008] is the ratio of the objective value achieved by an optimal adaptive policy and an optimal non-adaptive policy. We show the adaptive submodularity ratio can be used for obtaining a lower bound of the adaptivity gap.

We provide lower bounds of the adaptive submodularity ratio in two applications: adaptive influence maximization and adaptive feature selection. We show the adaptive submodularity ratio in the triggering model can be bounded if the underlying graph is bipartite, but it can be very small even if the underlying graph is a simple arborescence.

In Chapter 4, we extend the framework of the adaptive submodularity ratio to the *batch-mode setting* of adaptive optimization, which is more realistic than the ordinary setting in various applications. In the ordinary setting of adaptive optimization, the decision-maker gathers information in a fully sequential manner, but it is not practical due to a constraint on time or cost. The batch-mode setting is a more efficient setting, in which the decision-maker gathers information in a parallel manner. Chen and Krause [2013] first formulated the batch-mode setting of adaptive optimization and proposed a greedy-like algorithm. We apply the framework of the adaptive submodularity ratio to the batch-mode setting and analyze the algorithm for problems that lack adaptive submodularity.

1.4 Approximate Submodularity for Local Search

Local search is a common algorithm design technique for combinatorial optimization problems. It starts with an initial solution and repeatedly improves the solution by changing it locally. Though local search practically works well in various fields, it does not have theoretical guarantees on its performance in many cases. To judge whether or not local search is suitable for each problem, we need to investigate theoretical properties that guarantee local search to work well.

Submodularity has been utilized not only for analyzing greedy algorithms but also for analyzing local search algorithms. Based on local search approaches, several important results for submodular maximization have been obtained. In particular, for structured constraints, Lee et al. [2010] and Feldman et al. [2011] provided approximation guarantees better than best-known guarantees achieved by greedy algorithms.

In this study, we analyze local search algorithms for problems without submodularity. As mentioned before, the submodularity ratio is defined by relaxing the property equivalent to submodularity that is essential to the analysis of the greedy algorithm. For the analysis of local search algorithms for submodular maximization, another property derived from submodularity is utilized. We define approximate submodularity for local search as a relaxed version of this property. By utilizing this concept, we develop algorithms for feature selection in Chapter 5 and dictionary selection in Chapter 6, respectively.

In Chapter 5, we consider a sparse optimization problem, in which we aim at finding a sparse solution that maximizes a continuous function. Feature selection is the combinatorial optimization problem of finding the best sparse support for this sparse optimization problem. First, we show that restricted strong concavity and restricted smoothness of the continuous function imply approximate submodularity for local search. These conditions naturally arise in various applications, including sparse regression and structure learning of graphical models. We bound the approximation ratio of a simple local search algorithm, and then develop its accelerated versions while keeping the approximation ratio guarantees. Based on Lee et al. [2010] and Feldman et al. [2011], we can extend our proposed local

search algorithms to several classes of structured constraints such as matroid constraints, p-matroid intersection constraints, or p-exchange system constraints.

Further developing the techniques for approximate submodularity for local search, we tackle dictionary selection in Chapter 6. Dictionaries are a collection of patterns that appear in real signals and have many applications in compressed sensing and machine learning. In dictionary selection, we make a dictionary suitable to given signals by selecting several patterns out of finite candidates. Dictionary selection is formulated as an optimization problem that can be viewed as a two-stage version of the feature selection, therefore we can apply the techniques developed for feature selection. By incorporating the local search procedure into the greedy algorithm, we propose Replacement OMP, which runs in practical time and returns a reasonable solution in practice. Also, we define a class of structured constraints called p-replacement sparsity families, for which Replacement OMP achieves a good approximation.

1.5 Relevant Applications

In this section, we introduce applications of our proposed frameworks.

Feature Selection for Sparse Regression Given data points in the high-dimensional space, we often want to reduce the dimensions to obtain a robust and interpretable model. Feature selection for sparse regression is the problem of selecting a subset of features. The first study from the perspective of approximate submodularity is Das and Kempe [2011], which analyzed greedy algorithms for sparse linear regression. Their results were extended to restricted strongly convex and smooth objectives by Elenberg et al. [2018]. We apply local search algorithms to this problem in Chapter 5. Also, we consider the adaptive version, in which we observe features one by one, and analyze the adaptive greedy algorithm by bounding its adaptive submodularity ratio in Chapter 3.

Adaptive Influence Maximization To advertise a product efficiently, we should consider the information spread on social networks. By providing a free sample to a small number of people, we can expect that the information about the product spreads by word-of-mouth communication from them. Influence maximization is a problem of selecting a small subset of nodes of the given social network to spread the information to as many people as possible. In this dissertation, we particularly work on the setting where we can observe the spread from each node just after selecting it. This setting is called *adaptive influence maximization*. We analyze the adaptive greedy algorithm for adaptive influence maximization in the triggering model on bipartite graphs by bounding its adaptive submodularity ratio in Chapter 3. We analyze greedy-based algorithms for the batch-mode setting of adaptive influence maximization in Chapter 4.

Active Learning In real-world scenarios for supervised classification, it is often the case that unlabeled data points are cheap but it takes much cost to label these data points. Suppose we can use a *labeling oracle*, that is, we can obtain a label of each data point by paying a price. Then active learning can be used to achieve high accuracy with small labeling cost. In active learning, we select a small subset of unlabeled data points, and have the labeling oracle label them. By selecting this subset cleverly, we can achieve high accuracy by learning with this labeled subset. We can usually alternately repeat selecting an unlabeled data point and obtain its label. In this dissertation, we treat the *batch-mode* setting of active learning, in which we alternately repeat selecting multiple data points and obtain their labels. Batch-mode active learning with structured queries is dealt with in Chapter 4.

Structure Learning of Graphical Models Graphical models are a graph whose nodes correspond to random variables and edges represent the relationships between them. The problem of inferring edges of the graphical model from data that are generated from the random variables is important in a variety of applications. This problem is called *structure learning of graphical models*, and has been studied in machine learning and statistics. We deal with this problem in the case when the graphical model is sparse, i.e., the number of edges is small. By regarding the sparsity constraint as a b-matching constraint, we propose a local search algorithm with theoretical guarantees and conduct experiments in Chapter 5.

Dictionary Selection A dictionary is a collection of patterns that often appear in real signals. Dictionary selection is an approach to learning a dictionary suitable for the given data points. In dictionary selection, we select a subset of atoms as a dictionary out of the union of existing ready-made dictionaries. Dictionary selection can be regarded as a two-stage combinatorial optimization problem. In Chapter 6, we propose greedy algorithms, Replacement Greedy, Replacement OMP, and Replacement Deletion-OMP, and a class of generalized sparsity constraints, p-replacement sparsity families.

Multi-task Feature Selection Suppose we are given multiple tasks similar but different and solve them simultaneously. For example, let us consider the problem of making spam filters personalized for each of multiple users. It is natural to assume that the set of words relevant for judging whether or not an email is spam is different depending on users but similar. For such a scenario, we tackle the problem of selecting a subset of features for each task while considering relationships between tasks. Here, we consider a constraint that imposes the number of features used for at least one task. We apply the algorithms for dictionary selection to this formulation of multi-task feature selection in Chapter 6.

1.6 Thesis Organization

Chapter 2: Background and Related Work

We introduce basic fasts and existing studies about the topics that this dissertation deals with. First, we provide important existing results on submodular maximization, mostly focusing on greedy algorithms and local search algorithms. Next, we move to the introduction of approximate submodularity with its applications. Then we introduce basic definitions for adaptive submodular maximization. We also give notations used throughout this dissertation.

Chapter 3: Approximation Guarantees of Greedy Policy with Adaptive Submodularity Ratio

In this chapter, we propose a concept of adaptive submodularity ratio and provide its applications. First, we give the formal definition of the adaptive submodularity ratio. Then we provide a theoretical guarantee of the adaptive greedy algorithm in terms of the adaptive submodularity ratio. We also show that the adaptivity gap can be bounded by the product of the adaptive submodularity ratio and another parameter called the supermodularity ratio. We introduce two applications of our framework: One is bipartite adaptive influence maximization in the triggering model and the other is adaptive feature selection. Finally, we conduct experiments on these two applications and show the adaptive greedy algorithm works well in practice.

Chapter 4: Batch-mode Adaptive Optimization with Structured Queries

This chapter deals with the batch-mode setting of adaptive stochastic optimization. We show that adaptive submodularity is not sufficient for guaranteeing the performance of the adaptive greedy algorithm in the batch-mode setting, and propose a stronger concept than adaptive submodularity, which we call *set-adaptive submodularity*. Under the assumption of set-adaptive submodularity, we provide a theoretical guarantee for the batch-mode adaptive greedy algorithm. We consider the setting where set-adaptive submodularity does not hold, and provide a theoretical guarantee by using the adaptive submodularity ratio and the supermodularity ratio. We show that the batch-mode adaptive greedy algorithm also works for other extensions including a matroid constraint on the union of the selected batches, an online setting, a setting where the feasible batches change at each round. By conducting experiments on the batch-mode setting of adaptive influence maximization, active learning, and adaptive feature selection, we empirically illustrate that the batch-mode adaptive greedy algorithm is competitive with the adaptive greedy algorithm in the ordinary adaptive setting.

Chapter 5: Local Search for Feature Selection with Structured Constraints

In this chapter, we analyze local search algorithms for feature selection by proposing a property of approximate submodularity for local search. First, we formally define this property and show that this property of feature selection is derived from the restricted strong concavity and restricted smoothness of the underlying continuous function. We describe two applications: sparse regression and structure learning of graphical models. For a matroid constraint, we propose local search algorithms and their accelerated versions obtained by the idea of a quadratic approximation. Similarly, for p-matroid intersection constraints and p-exchange system constraints, we proposed local search algorithms. We empirically compare the proposed algorithms with existing methods to show the efficiency of the proposed algorithms.

Chapter 6: Fast Greedy Algorithms for Dictionary Selection

In this chapter, we propose fast greedy algorithms for a generalized version of dictionary selection. We formulate a problem setting of dictionary selection that generalizes existing problem settings. We propose a new class of sparsity constraints, which we call *p-replacement sparsity families*, and show existing sparsity constraints are included in this class. First, we apply Replacement Greedy to dictionary selection and provide a bound on the approximation ratio. Then we propose our main proposed algorithm Replacement OMP by accelerating Replacement Greedy and its fast implementation for a general sparsity constraint. We also propose an intermediate variant of Replacement Greedy and Replacement OMP, which we call Replacement Deletion-OMP. For the setting where data points arrive in an online fashion, we propose the online versions of Replacement Greedy and Replacement OMP. By conducting extensive experiments on synthetic and real datasets, we show Replacement OMP performs well compared not only to existing dictionary selection algorithms but also for basic dictionary learning algorithms.

1.7 Bibliographic Notes

Some contents of this dissertation were already published in refereed conference proceedings. Chapter 3 is based on the joint work with Shinsaku Sakaue presented in ICML 2019 [Fujii and Sakaue, 2019]. Chapter 6 is based on the joint work with Tasuku Soma presented in NeurIPS 2018 [Fujii and Soma, 2018]. Chapter 4 and Chapter 5 are based on unpublished single-authored work.

1.8 Basic Notation

In this section, we define notation used throughout this dissertation.

Sets and set families are denoted by upper case letters of roman and calligraphic fonts, respectively. We use V to denotes the finite ground set, from which we select a set of elements. For $X \subseteq V$ and $v \in V$, we define $X + v \coloneqq X \cup \{v\}$. Similarly, for $X \subseteq V$ and $v \in V$, we define $X - v \coloneqq X \setminus \{v\}$.

The sets of reals and non-negative reals are denoted by $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$, respectively. Similarly, the sets of integers and non-negative integers are denoted by $\mathbb{Z}$ and $\mathbb{Z}_{\geq 0}$. For any positive integer $n \in \mathbb{Z}_{\geq 0}$, we define $[n] = \{1, 2, \ldots, n\}$, the set of all positive integers no more than n.

Vectors and matrices are denoted by lower and upper case letters in boldface, respectively: $\mathbf{a}, \mathbf{x}, \mathbf{y}$ for vectors and $\mathbf{A}, \mathbf{X}, \mathbf{Y}$ for matrices. The ith standard unit vector is denoted by $\mathbf{e}_i$; that is, $\mathbf{e}_i$ is the vector such that its ith entry is equal to one and all other entries are zero. Throughout the dissertation, $\|\cdot\|$ represents the ℓ^2 norm. For any vector $\mathbf{a} \in \mathbb{R}^n$, let $\mathrm{supp}(\mathbf{a}) = \{i \in [n] \mid \mathbf{a}_i \neq 0\}$ be the set of indices with non-zero values and $\|\mathbf{a}\|_0 = |\mathrm{supp}(\mathbf{a})|$ the number of non-zero elements. Note that $\|\cdot\|_0$ is conventionally called "ell-zero norm", but does not satisfy the properties of the norm. For a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ and $X \subseteq [n]$, $\mathbf{A}_X$ denotes the column submatrix of $\mathbf{A}$ with respect to X. The maximum and minimum eigenvalues of a square matrix $\mathbf{M}$ are denoted by $\lambda_{\max}(\mathbf{M})$ and $\lambda_{\max}(\mathbf{M})$, respectively. For a positive integer k, we define $\lambda_{\max}(\mathbf{M}, k) \coloneqq \max_{X \subseteq [n]\colon |X| \leq k} \lambda_{\max}(\mathbf{M}[X, X])$, where $\mathbf{M}[X, X]$ is the submatrix with both of row and column indices X. We define $\lambda_{\min}(\mathbf{M}, k)$ similarly as $\lambda_{\min}(\mathbf{M}, k) \coloneqq \min_{X \subseteq [n]\colon |X| \leq k} \lambda_{\min}(\mathbf{M}[X, X])$.

2 Background and Related Work

In this chapter, we review the existing studies related to this dissertation. In Section 2.1, we introduce the definition of submodularity and standard problem settings of submodular maximization. In particular, we focus on two types of algorithms: *greedy algorithms* and *local search algorithms*. In Section 2.2, we illustrate existing studies on approximate submodularity, with a strong focus on the *submodularity ratio*. Here, we also introduce feature selection for sparse linear regression and more general problems as its important applications. In Section 2.3, we provide the definition and applications of *adaptive submodularity*, an analog of submodularity in the adaptive setting.

2.1 Submodular Maximization

In this section, we introduce basic facts of submodular maximization. We illustrate the general problem statement, and describe major greedy and local search algorithms for constrained monotone submodular maximization.

In machine learning, we are often faced with the problem of finding a good subset given a large number of elements such as features or data points. Such a problem can be formulated as an optimization problem of a *set function*. Let V denote the set that contains all the elements, which is called the *ground set*. A set function $f\colon 2^V \to \mathbb{R}$ is a function that assigns a real value to each subset of the ground set V. An optimization problem of a set function under some constraint can be written as

$$\begin{aligned} \text{Maximize} \quad & f(X) \\ \text{subject to} \quad & X \in \mathcal{I}, \end{aligned}$$

where $\mathcal{I} \subseteq 2^V$ is the set family of all feasible subsets. We assume $(V, \mathcal{I})$ is an *independence system*, i.e., $\varnothing \in \mathcal{I}$ and if $A \subseteq B \in \mathcal{I}$ then $A \in \mathcal{I}$ for any $A, B \subseteq V$.

Submodularity is a property of set functions that is important both in modeling and optimization, which is defined as follows.

Definition 1 (Submodularity)**.** Let V be an arbitrary finite set. A set function $f\colon 2^V \to \mathbb{R}$ is *submodular* if for any subset $S, T \subseteq V$, it holds that

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

It is widely known that submodularity is equivalent to *the property of diminishing returns* . To provide the formal definition of the property of diminishing returns, we need to define the *marginal gain* of a set function, which represents how the value of the function increases when an element is added to a subset. The marginal gain of element $v \in V$ with subset $S \subseteq V$ is defined as

$$f(v|S) \coloneqq f(S \cup \{v\}) - f(S).$$

By using this notation, we can state the equivalence of submodularity to the property of diminishing returns.

Proposition 2. *Let V be an arbitrary finite set. A set function $f\colon 2^V \to \mathbb{R}$ is submodular if and only if for any subset $S, T \subseteq V$ such that $S \subseteq T$ and any element $v \in V \setminus T$, it holds that*

$$f(v|S) \geq f(v|T).$$

Similarly, we define the marginal gain of subset $T \subseteq V$ with subset $S \subseteq V$ as $f(T|S) \coloneqq f(S \cup T) - f(S)$. Another important property of set functions is *monotonicity* defined as follows.

Definition 3 (Monotonicity)**.** Let V be an arbitrary finite set and $f\colon 2^V \to \mathbb{R}$ a set function. f is monotone if for any subset $S \subseteq V$ and any element $v \in V \setminus S$, it holds that

$$f(v|S) \geq 0.$$

In general, a set function may require a representation whose size is exponential in $|V|$. Hence, it is often assumed that we have access to a *value oracle*, which answers the function value in response to our queries. Formally, the value oracle of a set function $f\colon 2^V \to \mathbb{R}$ receives an input $X \subseteq V$ and returns the function value $f(X)$. Similarly, an independence system does not have a polynomial size representation in general. Therefore, we assume that we have access to an *independence oracle*, which answers the independence in response to our queries. Formally, the independence oracle of an independence system $(V, \mathcal{I})$ receives the input $X \subseteq V$ and returns the boolean value that represents $X \in \mathcal{I}$ or not.

In general, maximizing a monotone submodular function needs exponentially many oracle calls even in the case of cardinality constraints, i.e., $\mathcal{I} = \{X\colon |X| \leq k\}$ [Nemhauser and Wolsey, 1978]. Therefore, we consider *approximation algorithms* that run in polynomial time.

Definition 4 (Approximation ratio for maximization problems)**.** Assume the objective function $f\colon 2^V \to \mathbb{R}_{\geq 0}$ is non-negative. An algorithm is *α-approximation* if it returns a solution whose objective value is at least α times the optimal value for any problem instance, i.e., if X is the output of the algorithm, then

$$f(X) \geq \alpha \max_{X^* \in \mathcal{I}} f(X^*).$$

In this dissertation, we consider maximization problems, therefore approximation ratio α is always between 0 and 1.

Submodular minimization. As an opposite of maximization, *minimization* of submodular functions is also a profound research topic through its intense relationship with convex optimization [Fujishige, 2005] and has been applied to machine learning problems [Bach, 2013], but we focus on only submodular maximization in this dissertation.

2.1.1 Greedy Algorithms for Submodular Maximization

Here, we review several variants of greedy algorithms for submodular maximization. Existing studies have shown that submodularity is a key property that enables us to provide a theoretical guarantee for greedy algorithms.

Among many variants of greedy algorithms, the simplest one was proposed by Nemhauser et al. [1978], which we call the greedy algorithm in this dissertation. The greedy algorithm starts with the empty set and repeatedly adds the element with the largest marginal gain at each step. It stops when any element cannot be added due to the constraint. The detailed description of the greedy algorithm is given in Algorithm 1.

In the case of cardinality constraints, i.e., $\mathcal{I} = \{X\colon |X| \leq k\}$, Nemhauser et al. [1978] proved that the greedy algorithm returns the solution whose objective value is at least $(1 - 1/\mathrm{e})$ times the optimal value.

Algorithm 1 The greedy algorithm [Nemhauser et al., 1978]

Input The objective function $f\colon 2^V \to \mathbb{R}$ given by a value oracle, the independence system $(V, \mathcal{I})$ given by an independence oracle.

Output $X \in \mathcal{I}$.

1: $X \leftarrow \varnothing$.
2: $F \leftarrow V$.
3: **while** $F \neq \varnothing$ **do**
4: $\quad v \leftarrow \operatorname{argmax}_{v \in F} f(v|X)$.
5: $\quad X \leftarrow X \cup \{v\}$.
6: $\quad F \leftarrow \{v \in V \mid X \cup \{v\} \in \mathcal{I}\}$.
7: **return** X.

Theorem 5 ([Nemhauser et al., 1978]). *Suppose the objective function is monotone and submodular, and $\mathcal{I} = \{X\colon |X| \leq k\}$. If X is the output of Algorithm 1 and $X^* \in \operatorname{argmax}_{X\colon |X| \leq k} f(X)$ is an optimal solution, then it holds that*

$$f(X) \geq \left(1 - \frac{1}{\mathrm{e}}\right) f(X^*).$$

This approximation ratio is proved to be best possible under the assumption of $\mathrm{P} \neq \mathrm{NP}$ in the special case of max k-cover [Feige, 1998] or polynomially many oracle calls [Nemhauser and Wolsey, 1978].

Matroid constraints. The greedy algorithm can be applied to more general settings, including a *matroid constraint*. A matroid has been used for modeling discrete structures in the real world.

Definition 6 (Matroids). Let V be a finite set and $\mathcal{I} \subseteq 2^V$ be a set family. An independence system $\mathcal{M} = (V, \mathcal{I})$ is called a matroid if for any $S, T \in \mathcal{I}$ with $|S| < |T|$, there exists $v \in T \setminus S$ such that $S \cup \{v\} \in \mathcal{I}$.

The greedy algorithm is guaranteed to achieve $1/2$-approximation for a matroid constraint [Fisher et al., 1978].

Theorem 7 ([Fisher et al., 1978]). *Suppose the objective function is monotone and submodular, and $(V, \mathcal{I})$ is a matroid. If X is the output of Algorithm 1 and $X^* \in \operatorname{argmax}_{X \in \mathcal{I}} f(X)$ is an optimal solution, then it holds that*

$$f(X) \geq \frac{1}{2} f(X^*).$$

They also showed that this bound is tight by providing an example where the greedy algorithm returns a $1/2$-approximate solution. Călinescu et al. [2011] proposed the continuous greedy algorithm, which utilizes a continuous relaxation of the objective function and rounding techniques, and proved it to be $(1 - 1/\mathrm{e})$-approximation.

p-System constraints. A p-system is a general class of independence systems that include important classes such as p-matroid intersection constraints and p-exchange system constraints. Formally, p-systems are defined as follows.

Definition 8 (p-Systems [Jenkyns, 1976, Călinescu et al., 2011]). Let $\mathcal{M} = (V, \mathcal{I})$ be an independence system with a finite set V and set family $\mathcal{I} \subseteq 2^V$. We define a base of $X \subseteq V$ to be a maximal independent subset of X, that is, $Y \subseteq X$ such that $Y \in \mathcal{I}$ and for any $v \in X \setminus Y$, we have $Y \cup \{v\} \notin \mathcal{I}$. An independence system $\mathcal{M} = (V, \mathcal{I})$ is called a p-system if for any $X \subseteq V$, the cardinality of a largest size base of X is at most p times the cardinality of a smallest size base of X.

The greedy algorithm is guaranteed to achieve $1/(p+1)$-approximation for a p-system constraint, which was formally proved by [Călinescu et al., 2011].

Theorem 9 ([Călinescu et al., 2011]). *Suppose the objective function is monotone and submodular, and $(V, \mathcal{I})$ is a p-system. If X is the output of Algorithm 1 and $X^* \in \operatorname{argmax}_{X \in \mathcal{I}} f(X)$ is an optimal solution, then it holds that*

$$f(X) \geq \frac{1}{p+1} f(X^*).$$

2.1.2 Local Search for Submodular Maximization

While submodularity has provided theoretical bounds for the approximation ratios of greedy algorithms, it is also useful for analyzing local search. Several existing studies have designed algorithms based on the idea of local search and provided lower bounds on their approximation ratios under the assumption of submodularity. The first result of this approach is given by Nemhauser et al. [1978]. They showed that any local optimal solution is guaranteed to be $1/2$-approximation for monotone submodular maximization under a cardinality constraint. Formally, Nemhauser et al. [1978] defined that $X \in \mathcal{I}$ is a *q-interchange solution* if there is no $Y \in \mathcal{I}$ such that $|Y \setminus X| \leq q$, $|X \setminus Y| \leq q$, and $f(Y) > f(X)$.

Theorem 10 ([Nemhauser et al., 1978]). *Suppose $f \colon 2^V \to \mathbb{R}_{\geq 0}$ is monotone and submodular and the constraint is a cardinality constraint, i.e., $\mathcal{I} = \{X \subseteq V \colon |X| \leq k\}$. If $k = qs - r$ such that s is a positive integer and r is an integer such that $0 \leq r \leq q - 1$, then any q-interchange solution is $\frac{k-q+r}{2k-q+r}$-approximation.*

This bound is worse than that for the greedy algorithm in general. In other words, the local search procedure does not improve the approximation ratios of the greedy algorithm for cardinality constraints. It has been shown that local search algorithms derive the approximation bound better than the greedy algorithm for more complicated constraints. Lee et al. [2010] and Feldman et al. [2011] proposed local search algorithms whose approximation ratio bounds are better than those of the greedy algorithm for *p-matroid intersection constraints* and *p-exchange system constraints*, respectively. Here, we provide an overview of their results.

First, we introduce a local search algorithm proposed by Lee et al. [2010] for a p-matroid intersection constraint, which is a special case of p-system constraints. This is a constraint that can be expressed by the intersection of p matroids, which is formally defined as follows.

Definition 11 (p-Matroid intersection). Let V be a finite set and $\mathcal{I} \subseteq 2^V$ be a non-empty set family. An independence system $(V, \mathcal{I})$ is a p-matroid intersection if there exist p matroids $(V, \mathcal{I}_1), \cdots, (V, \mathcal{I}_p)$ such that $\mathcal{I} = \bigcap_{i=1}^{p} \mathcal{I}_i$.

To define a local search algorithm, we need to specify what is a local improvement from each feasible solution. Lee et al. [2010] defined the *q-reachability* among feasible solutions of a p-matroid intersection constraint, which represents the neighborhood of each solution in some sense, as follows.

Definition 12 (q-Reachability for p-matroid intersection [Lee et al., 2010]). Let $\mathcal{I} \subseteq 2^V$ be a p-matroid intersection. A feasible solution $T \in \mathcal{I}$ is q-reachable from $S \in \mathcal{I}$ if $|T \setminus S| \leq 2q$ and $|S \setminus T| \leq 2pq$.

A naive local search procedure repeatedly updates the solution to a better q-reachable solution and stops when there does not exist any better q-reachable solution. Since this naive algorithm does not guarantee the polynomial time complexity, they modify the algorithm so that it updates the solution to a solution with at least $(1 + \delta)$ times the current objective value for some constant $\delta > 0$. The detailed algorithmic description is provided in Algorithm 2. Lee et al. [2010] proved the polynomial time complexity and provided a bound on its approximation ratio.

Algorithm 2 Local search for a p-matroid intersection or p-exchange system constraint ($p \geq 2$)

Let $\epsilon = \frac{\delta}{n(k+1)}$.
Let $q = \lceil \frac{1}{\delta - \epsilon} \rceil$.
Let $X \leftarrow \operatorname{argmax}\{f(v) \mid v \in V\}$.
loop
  Search for X' that is q-reachable from X such that $f(X') > (1+\epsilon)f(X)$
  if $\exists\, X'$ satisfying the above condition **then**
    Let $X \leftarrow X'$.
  else
    return X.

Theorem 13 ([Lee et al., 2010]). *Suppose the objective function is monotone and submodular. If $\mathcal{I}$ is a p-matroid intersection for $p \geq 2$, Algorithm 2 runs in time polynomial in n and achieves $1/(p+\delta)$-approximation.*

A similar result was published by Feldman et al. [2011] for a p-exchange system constraint, which is also a special case of p-system constraints.

Definition 14 (p-Exchange systems [Feldman et al., 2011]). Let V be a finite set and $\mathcal{I} \subseteq 2^V$ be a non-empty set family. An independence system $(V, \mathcal{I})$ is a p-exchange system if for any $S, T \in \mathcal{I}$, there exist a map $\varphi\colon (T \setminus S) \to 2^{S \setminus T}$ such that .

1. For any $v \in T \setminus S$, it holds that $|\varphi(v)| \leq p$.
2. Each $v \in S \setminus T$ appears in at most p sets of $(\varphi(v))_{v \in T \setminus S}$.
3. For any $X \subseteq T \setminus S$, it holds that $(S \setminus \bigcup_{v \in X} \varphi(v)) \cup X \in \mathcal{I}$.

This class includes many important independence systems as special cases. For example, the family of all b-matchings in a general graph is a 2-exchange system. Feldman et al. [2011] defined the q-reachability for p-exchange systems, which is similar to the one for p-matroid intersection but different.

Definition 15 (q-Reachability for p-exchange systems [Feldman et al., 2011]). Let $\mathcal{I} \subseteq 2^V$ be a p-matroid intersection or p-exchange system. A feasible solution $T \in \mathcal{I}$ is q-reachable from $S \in \mathcal{I}$ if $|T \setminus S| \leq q$ and $|S \setminus T| \leq pq - q + 1$.

By using the different definition of q-reachability, we can reuse the same algorithmic description as that for p-matroid intersection given in Algorithm 2. The resulting approximation ratio is the same as that for p-matroid intersection.

Theorem 16 ([Feldman et al., 2011]). *Suppose the objective function is monotone and submodular. If $\mathcal{I}$ is a p-exchange system for $p \geq 2$, Algorithm 2 returns an output in time polynomial in n and achieves $1/(p+\delta)$-approximation.*

Other results on local search for submodular maximization. For unconstrained maximization of non-monotone non-negative submodular functions, Feige et al. [2011] developed deterministic $1/3$-approximation and randomized $2/5$-approximation algorithms based on local search. Feige et al. [2011] also showed that a deterministic local search algorithm achieves $1/2$-approximation if the objective function is symmetric. In the case of $p \geq 3$, an improved $\frac{2}{p+3}$-approximation local search algorithm for

monotone submodular maximization under a p-exchange system constraint was proposed by Ward [2012]. Another optimal approximation algorithm for monotone submodular maximization under a matroid constraint was devised based on the idea of local search [Filmus and Ward, 2014]. This algorithm repeatedly improves the surrogate function constructed from the original objective function and it achieves $(1 - 1/\mathrm{e})$-approximation in polynomial time.

2.2 Approximate Submodularity

In this section, we review existing studies on approximate submodularity. First, we introduce the submodularity ratio, which was defined for analyzing the approximation ratio of greedy algorithms. Then we summarize its applications to feature selection for sparse linear regression and sparse optimization for restricted strong concave and restricted smooth objective functions. Finally, we provide a brief explanation of other existing notions of approximate submodularity.

2.2.1 Submodularity Ratio

One of the most prevalent concepts of approximate submodularity is the *submodularity ratio*. The starting point of this concept is the following property equivalent to submodularity.

Proposition 17. *$f\colon 2^V \to \mathbb{R}$ is submodular if and only if for all $S \subseteq V$ and $L \subseteq V$, we have $\sum_{v\in S} f(v|L) \geq f(S|L)$.*

Intuitively, this property represents the marginal gain of a set is no more than the sum of the marginal gains of single elements. This property plays an important role in the proof of the approximation ratio of the greedy algorithm. The submodularity ratio is defined as a parameter that represents how much we must relax this property as follows.

Definition 18 (Submodularity ratio [Das and Kempe, 2011])**.** Let $f\colon 2^V \to \mathbb{R}_{\geq 0}$ be a non-negative set function. The submodularity ratio of a monotone non-negative set function $f\colon 2^V \to \mathbb{R}_{\geq 0}$ with respect to set $U \subseteq V$ and parameter $k \geq 1$ is defined to be

$$\gamma_{U,k}(f) \coloneqq \min_{L\subseteq U,\ S\colon |S|\leq k} \frac{\sum_{v\in S} f(v|L)}{f(S|L)}, \tag{2.1}$$

where $f(v|L) \coloneqq f(L\cup\{v\}) - f(L)$ and $f(S|L) \coloneqq f(L\cup S) - f(L)$. If the numerator and denominator are both 0, the submodularity ratio is considered to be 1.

If the submodularity ratio is large, we can say the set function is close to submodular functions. In fact, as shown in the following proposition, the submodularity ratio is always between 0 and 1, and the submodularity ratio is equal to 1 if and only if the set function is submodular.

Proposition 19 ([Das and Kempe, 2011])**.** *We have $\gamma_{U,k} \in [0, 1]$, and a monotone set function f is submodular if and only if $\gamma_{U,k} = 1$ for every $U \subseteq V$ and $k \geq 1$.*

If the submodularity ratio is bounded away from 0, we can guarantee the approximation ratio of the classical greedy algorithm.

Theorem 20 ([Das and Kempe, 2011])**.** *If X is the output of the greedy algorithm and $X^* \in \operatorname{argmax}_{X\colon |X|\leq k} f(X)$ is an optimal solution, then*

$$f(X) \geq (1 - \exp(-\gamma_{X,k}(f)))f(X^*).$$

We can see that this theorem is a generalization of Theorem 5. This bound is illustrated in Section 2.2.1.

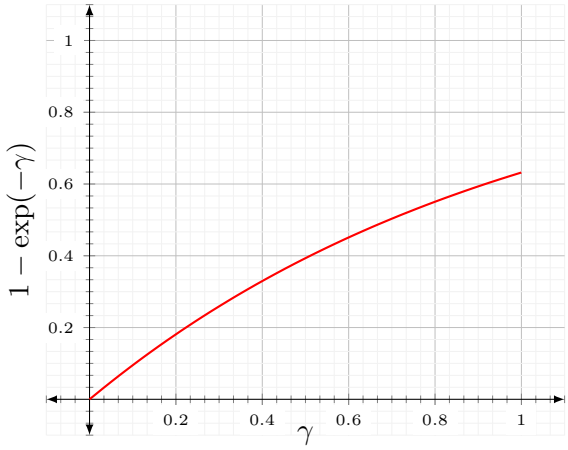


Figure 2.1: The approximation ratio bound of the greedy algorithm provided by Das and Kempe [2011]. We can see the bound is equal to 0 if the submodularity ratio is 0 and equal to $1 - 1/\mathrm{e}$ if the submodularity ratio is 1.

Supermodularity ratio. Supermodularity is an opposite concept of submodularity. A set function is called supermodular if its negative is submodular. As an opposite concept of the submodularity ratio, the *supermodularity ratio*, was considered in Bogunovic et al. [2018], which is defined as follows.

$$\beta_{U,k}(f) \coloneqq \min_{L \subseteq U,\ S\colon |S| \leq k} \frac{f(S|L)}{\sum_{v \in S} f(v|L)}, \tag{2.2}$$

where we regard $0/0 = 1$. We have $\beta_{U,k} \in [1/k, 1]$, and f is supermodular if and only if $\beta_{U,k} = 1$ for every $U \subseteq V$ and $k \geq 1$. We omit f from $\gamma_{U,k}(f)$ and $\beta_{U,k}(f)$ if it is clear from the context.

2.2.2 Feature Selection for Sparse Linear Regression

The first application of the concept of submodularity ratio was sparse linear regression, which is a fundamental problem in machine learning and compressed sensing. Das and Kempe [2011] proposed the concept of submodularity ratio and applied it to sparse linear regression. They focused on the optimization problem of maximizing the coefficient of determination, denoted by R^2, which represents the fraction of variance that is predicted by the trained predictor. In this problem, given a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ and a response vector $\mathbf{y} \in \mathbb{R}^d$, we aim at learning a sparse parameter $\mathbf{w} \in \mathbb{R}^d$ that fits the training dataset $\mathbf{A}$ and $\mathbf{y}$. A vector is called *sparse* when the number of non-zero elements is small. The optimization problem that we are interested in is

$$\begin{aligned} &\text{Maximize} && R^2 \coloneqq 1 - \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 / \|\mathbf{y}\|_2^2 \\ &\text{subject to} && \|\mathbf{w}\|_0 \leq s, \end{aligned}$$

where $\|\cdot\|_0$ represents the number of non-zero elements of a vector and s is the upper bound on the number of non-zero elements of $\mathbf{w}$. This problem can be formulated as a combinatorial optimization problem of selecting the set of non-zero elements of $\mathbf{w}$, that is,

$$\begin{aligned} &\text{Maximize} && f_{R^2}(X) \coloneqq 1 - \min_{\mathrm{supp}(\mathbf{w}) \subseteq X} \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 / \|\mathbf{y}\|_2^2 \\ &\text{subject to} && |X| \leq s. \end{aligned} \tag{2.3}$$

If we can obtain the solution X to (2.3), we can obtain the solution to the original problem by solving the ordinary linear regression problem, i.e.,

$$\mathbf{w}_i = \begin{cases} ((\mathbf{A}_X^\top \mathbf{A}_X)^+ (\mathbf{A}_X^\top \mathbf{y}))_i & \text{if } i \in X \\ 0 & \text{if } i \notin X, \end{cases}$$

where $\mathbf{A}_X$ represents the column submatrix of $\mathbf{A}$ with column indices X. Unfortunately, the optimization problem (2.3) was proved to be NP-hard by Natarajan [1995]. Hence a natural next research direction would be to consider approximation algorithms. Das and Kempe [2011] showed that, though the objective function f_{R^2} does not satisfy submodularity, the submodularity ratio of f_{R^2} can be bounded by a spectral parameter of $\mathbf{A}$ as follows.

Theorem 21 ([Das and Kempe, 2011, Lemma 2.4]). *Assume each column of $\mathbf{A}$ is normalized. Then*

$$\gamma_{U,s} \geq \min_{S \subseteq V \colon |S| \leq s + |U|} \lambda_{\min}(\mathbf{A}_S^\top \mathbf{A}_S).$$

The lower bound of the submodularity ratio is related to the restricted isometry constants defined as follows.

Definition 22 (Restricted isometry constants [Candes and Tao, 2005]). The sth restricted isometry constant $\delta_s = \delta_s(\mathbf{A})$ of a matrix $\mathbf{A} \in \mathbb{C}^{d \times n}$ is the smallest $\delta \geq 0$ such that

$$(1 - \delta)\|\mathbf{w}\|_2^2 \leq \|\mathbf{A}\mathbf{w}\|_2^2 \leq (1 + \delta)\|\mathbf{w}\|_2^2$$

for all $\mathbf{w} \in \mathbb{C}^n$ such that $\|\mathbf{w}\|_0 \leq s$.

Since the minimum eigenvalue is equal to the minimum Rayleigh quotient, the submodularity ratio is no less than $1 - \delta$. It is known that the restricted isometry constants can be bounded in various situations such as the case where each element of $\mathbf{A}$ is generated from some independent Gaussian distribution [Candes and Tao, 2005].

Forward regression. The greedy algorithm applied to feature selection for linear regression is called *forward regression* and has been widely used. At each step, this algorithm adds a new column of $\mathbf{A}$ that reduces the loss function value the most. This algorithm needs to solve the ordinary linear regression for calculating each reduction.

Theorem 23 ([Das and Kempe, 2011]). *If X is the output returned by forward regression and X^* is an optimal solution, then it holds that*

$$f_{R^2}(X) \geq \left(1 - \exp\left(-\min_{S \colon |S| \leq 2s} \lambda_{\min}(\mathbf{A}_S^\top \mathbf{A}_S)\right)\right) f_{R^2}(X^*).$$

The detailed description is provided in Algorithm 3.

Orthogonal matching pursuit. Orthogonal matching pursuit (OMP) is another greedy method for feature selection. At each step, it solves the ordinary linear regression with the current support X and computes the residual $\hat{\mathbf{y}} = \mathbf{y} - \mathbf{A}\mathbf{w}^{(X)}$, where $\mathbf{w}^{(X)}$ is an optimal parameter vector for support X. Orthogonal matching pursuit selects a column with the largest absolute value of correlation with the residual, i.e., $\operatorname{argmax}_{x \in V \setminus X} |\langle \hat{\mathbf{y}}, \mathbf{A}_x \rangle|$. First, Das and Kempe [2011] proved that OMP achieves $(1 - (\min_{S \colon |S| \leq 2s} \lambda_{\min}(\mathbf{A}_S^\top \mathbf{A}_S))^2)$-approximation. Elenberg et al. [2018] provided an improved result.

Algorithm 3 Forward regression for sparse linear regression

Let $X \leftarrow \varnothing$.
for $i = 1, \cdots, s$ **do**
  $x \leftarrow \operatorname{argmin}_{x \in V \setminus X} \min_{\mathbf{w}\colon \operatorname{supp}(\mathbf{w}) \subseteq X + x} \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2$.
  $X \leftarrow X + x$.
return X.

Algorithm 4 Orthogonal matching pursuit for sparse linear regression

Let $X \leftarrow \varnothing$.
for $i = 1, \cdots, s$ **do**
  Update parameter vector $\mathbf{w}^{(X)} = \operatorname{argmin}_{\mathbf{w}\colon \operatorname{supp}(\mathbf{w}) \subseteq X} \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2$.
  Update residual $\hat{\mathbf{y}} = \mathbf{y} - \mathbf{A}\mathbf{w}^{(X)}$.
  $x \leftarrow \operatorname{argmax}_{x \in V \setminus X} |\langle \hat{\mathbf{y}}, \mathbf{A}_x \rangle|$.
  $X \leftarrow X + x$.
return X.

Theorem 24 ([Elenberg et al., 2018]). *If X is the output returned by orthogonal matching pursuit and X^* is an optimal solution, then it holds that*

$$f_{R^2}(X) \geq \left(1 - \exp\left(-\min_{S\colon |S| \leq 2s} \lambda_{\min}(\mathbf{A}_S^\top \mathbf{A}_S)\right)\right) f_{R^2}(X^*).$$

Actually, Elenberg et al. [2018] dealt with a more general setting than sparse linear regression, which is described in the next subsection. The algorithmic description of OMP for sparse linear regression is given in Algorithm 4.

Other approaches to sparse regression. In machine learning and compressed sensing, many studies have been devoted to sparse regression. Various algorithms have been developed for sparse regression such as lasso [Tibshirani, 1996] and forward-backward greedy methods [Zhang, 2011, Jalali et al., 2011] as well as forward regression and orthogonal matching pursuit. A popular goal of theoretical analyses of sparse regression is *sparse recovery guarantees* [Foucart and Rauhut, 2013], which theoretically ensure that the output of an algorithm coincides with the true sparse parameter under some reasonable conditions. In this dissertation, we focus on bounds on approximation ratios, which cannot be directly compared with sparse recovery guarantees.

2.2.3 Restricted Strong Concavity and Restricted Smoothness

Elenberg et al. [2018] extended the framework on linear regression by Das and Kempe [2011] to more general sparse optimization problems including maximum likelihood estimation for generalized linear models, the problem of learning the structure of a graphical model [Jalali et al., 2011], and M-estimators [Negahban et al., 2012]. They considered the following optimization problem.

$$\begin{aligned} &\text{Maximize} && u(\mathbf{w}) \\ &\text{subject to} && \|\mathbf{w}\|_0 \leq s, \end{aligned}$$

where $u\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$. Linear regression is a special case of this optimization problem where $u(\mathbf{w}) = 1 - \|\mathbf{y} - \mathbf{A}\mathbf{w}\|_2^2 / \|\mathbf{y}\|_2^2$. In the same way as feature selection for linear regression, this problem can be

formulated as a combinatorial optimization problem of selecting the set of non-zero elements of $\mathbf{w}$, that is,

$$\begin{aligned} &\text{Maximize} \quad f_u(X) \coloneqq \max_{\mathbf{w}\colon \operatorname{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w}) \\ &\text{subject to} \quad |X| \leq s. \end{aligned} \tag{2.4}$$

Elenberg et al. [2018] argued that the *restricted strong concavity* and *restricted smoothness* are a sufficient condition for bounding the submodularity ratio of f_u. These conditions are defined as follows.

Definition 25 (Restricted strong concavity and restricted smoothness [Negahban et al., 2012, Elenberg et al., 2018])**.** Let Ω be a subset of $\mathbb{R}^d \times \mathbb{R}^d$ and $u\colon \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function. We say that u is *restricted strongly concave* with parameter m_Ω and *restricted smooth* with parameter M_Ω on domain Ω if,

$$-\frac{m_\Omega}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 \geq u(\mathbf{y}) - u(\mathbf{x}) - \langle \nabla u(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle \geq -\frac{M_\Omega}{2}\|\mathbf{y}-\mathbf{x}\|_2^2$$

for all $(\mathbf{x}, \mathbf{y}) \in \Omega$.

We define $\Omega_{s,p} \coloneqq \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^d\colon \|\mathbf{x}\|_0, \|\mathbf{y}\|_0 \leq s, \|\mathbf{x}-\mathbf{y}\|_0 \leq p\}$ and $\Omega_s \coloneqq \Omega_{s,s}$ for positive integers s and p. We often abbreviate M_{Ω_s}, $M_{\Omega_{s,p}}$, and m_{Ω_s} as M_s, $M_{s,p}$, and m_s, respectively. Elenberg et al. [2018] provided a lower bound on the submodularity ratio by using these constants as follows.

Theorem 26 ([Elenberg et al., 2018])**.** *Suppose $u\colon 2^V \to \mathbb{R}$ is a continuous function and $f\colon 2^V \to \mathbb{R}$ is a set function defined as $f(X) = \max_{\mathbf{w}\colon \operatorname{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w})$. We have*

$$\gamma_{U,s}(f_u) \geq \frac{m_{|U|+s}}{M_{|U|+1,1}}.$$

The greedy algorithm, which is called forward regression in this context, can be applied to this problem. In the same way as that for feature selection for sparse linear regression, this algorithm adds an element that increases the objective function the most at each step. Orthogonal matching pursuit is also generalized to this setting. At each step, it adds the element with the largest absolute value of derivative of u. Their algorithmic descriptions are given in Algorithm 5. The approximation ratios of these algorithms are bounded as follows.

Theorem 27 ([Elenberg et al., 2018])**.** *If X is the output returned by forward regression or orthogonal matching pursuit and X^* is an optimal solution for (2.4), then it holds that*

$$f_u(X) \geq \left(1 - \exp\left(-\frac{m_{2s}}{M_{s,1}}\right)\right) f_u(X^*).$$

This approach based on restricted strong concavity and restricted smoothness was extended to low rank optimization of matrices [Khanna et al., 2017].

2.2.4 Other Concepts for Approximate Submodularity

In this subsection, we review concepts of approximate submodularity other than the submodularity ratio.

Algorithm 5 Forward regression and orthogonal matching pursuit for general feature selection

1: Let $X \leftarrow \varnothing$.
2: **for** $i = 1, \cdots, s$ **do**
3: Search for $x \in X$ and $x' \in V \setminus X$ such that $X - x + x' \in \mathcal{I}$ and

$$x \leftarrow \begin{cases} \operatorname*{argmax}_{x \in V \setminus X} \max_{\operatorname{supp}(\mathbf{w}) \subseteq X + x} u(\mathbf{w}) & \text{(FR)} \\ \operatorname*{argmax}_{x \in V \setminus X} \left| \left(\nabla u(\mathbf{w}^{(X)}) \right)_x \right| \quad \text{where} \quad \mathbf{w}^{(X)} \in \operatorname*{argmax}_{\mathbf{w}\colon \operatorname{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w}) & \text{(OMP)} \end{cases}$$

4: **return** X.

Supermodular degree. Supermodular degree was first introduced by Feige and Izsak [2013] for measuring the degree of deviation from submodularity. In contrast to the submodularity ratio, the supermodular degree measures the deviation from the property of diminishing returns.

Definition 28 (Supermodular degree [Feige and Izsak, 2013])**.** The supermodular degree of an element $u \in V$ by f is defined as the cardinality of the set $\mathcal{D}_f^+(u) = \{v \in V | \exists S \subseteq V,\ f(u|S+v) > f(u|S)\}$, containing all elements whose existence in a set might increase the marginal contribution of u. The supermodular degree of a function f, denoted by $\mathcal{D}_f^+$, is the maximum supermodular degree of any element $u \in V$. Formally, $\mathcal{D}_f^+ = \max_{u \in V} |\mathcal{D}_f^+(u)|$.

It is readily seen that $0 \leq \mathcal{D}_f^+ \leq n - 1$ for any set function f, and $\mathcal{D}_f^+ = 0$ if and only if f is submodular. Feldman and Izsak [2014] proposed an algorithm that runs in time exponential in $\mathcal{D}_f^+$ and achieves a constant factor approximation.

Theorem 29 ([Feldman and Izsak, 2014, Theorem 2.4])**.** *There exists a* $(1 - \mathrm{e}^{-1/(\mathcal{D}_f^+ + 1)})$*-approximation algorithm of* $\operatorname{poly}(|V|, 2^{\mathcal{D}_f^+})$*-time complexity for the problem of maximizing a non-negative monotone set function* f *subject to a uniform matroid constraint.*

Approximate submodularity with a multiplicative error. Horel and Singer [2016] considered a problem of maximizing a set function f that satisfies approximate submodularity with a multiplicative error, that is, there exists an unknown submodular function g such that

$$(1 - \epsilon) g(S) \geq f(S) \geq (1 + \epsilon) g(S).$$

They also assumed g is not only submodular, but also monotone. They proved that if $\epsilon = \mathrm{O}(\delta/k)$, the greedy algorithm achieves $(1 - 1/\mathrm{e} - O(\delta))$-approximation. On the other hand, they also proved that if $\epsilon \geq n^{-\frac{1}{2}+\beta}$ for some $0 < \beta < \frac{1}{2}$, there is no algorithm with query complexity smaller than $2^{\Omega(\beta^{\beta/2})}$ that achieves approximation ratio better than $2/n^{\beta/2}$ with probability at least $1 - \frac{1}{2^{\Omega(n^{\beta/2})}}$.

2.3 Adaptive Submodularity

In this section, we introduce adaptive submodularity, which is an analog of submodularity in the adaptive setting.

2.3.1 Adaptive Stochastic Optimization

Adaptive stochastic optimization is a general framework for handling problems of sequentially selecting elements, where we can observe the states of only the selected elements. Let V be the ground set

$\pi(\varnothing) = a$
$\pi(\{(a,+)\}) = b$
$\pi(\{(a,-)\}) = c$
$\pi(\{(a,+),(b,+)\}) = d$
$\pi(\{(a,+),(b,-)\}) = c$
$\pi(\{(a,+),(c,-)\}) = e$

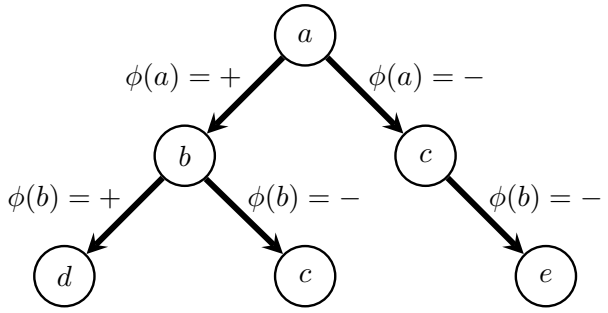


Figure 2.2: An example of a policy and its corresponding policy tree. The ground set is $V = \{a, b, c, d, e\}$ and the set of possible states is $\mathcal{Y} = \{+, -\}$.

consisting of a finite number of elements. Suppose every element $v \in V$ is assigned to some state in $\mathcal{Y}$, which is the set of all possible states. We let $\phi\colon V \to \mathcal{Y}$ be a map that associates each element, $v \in V$, with a state, $\phi(v) \in \mathcal{Y}$. We consider the Bayesian setting where ϕ is generated from a known prior distribution $p(\phi)$. Let Φ be a random variable representing the randomness of the realization ϕ.

A decision-maker can select one element $v \in V$ at each step. After selecting v, she can observe the state $\phi(v)$ of v. She repeatedly selects an element and then observes its state. The important point is that she can utilize the information about the states observed so far for selecting the next element. We denote by $\psi = \{(v_1, \phi(v_1)), \ldots, (v_\ell, \phi(v_\ell))\}$ the partial realization observed so far, where $\{v_1, \ldots, v_\ell\}$ is the set of selected elements.

Policies. The decision-maker's strategy for selecting elements can be encoded as a *policy*. Formally, a policy π is a partial map that returns an element $v \in V$ to be selected next given partial realization ψ observed so far. A policy can be described as a decision tree that determines the element to be selected next as illustrated in Figure 2.2.

Optimization formulation. The goal of the decision-maker is to maximize the expected value of the objective function $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$. The objective function value $f(S, \phi)$ depends on the set S of selected elements and the states ϕ of all elements. At the beginning, she does not know ϕ, but she can get partial information about ϕ by observing state $\phi(v)$ of selected v. In parallel, she must select elements to construct S that has high utility under the realization ϕ.

Let $E(\pi, \phi) \subseteq V$ be the set selected by policy π under realization ϕ. The expected value achieved by policy π is

$$f_{\text{avg}}(\pi) = \mathbb{E}_\Phi[f(E(\pi, \Phi), \Phi)], \tag{2.5}$$

where the expectation is taken with regard to the random variable Φ generated from p. The maximization version can be written as

$$\begin{aligned} \text{Maximize} \quad & f_{\text{avg}}(\pi) \\ \text{subject to} \quad & \pi \in \Pi_k, \end{aligned}$$

where $\Pi_k \coloneqq \{\pi \mid \forall \phi,\ |E(\pi, \phi)| \leq k\}$ is the set of all policies whose heights do not exceed k. Similarly, we can define the expected cost of policy π as

$$c_{\text{avg}}(\pi) = \mathbb{E}_\Phi[c(E(\pi, \Phi))],$$

where $c\colon 2^V \to \mathbb{R}_{\geq 0}$ be the cost function. The coverage version can be written as

$$\begin{aligned} &\text{Minimize} && c_{\text{avg}}(\pi) \\ &\text{subject to} && f(E(\pi, \phi), \phi) \geq Q \quad (\forall \phi), \end{aligned}$$

where $Q \in \mathbb{R}_{\geq 0}$ is a threshold. Here we assume $c(S) = |S|$ is the number of the elements in S.

Expected marginal gain. In contrast to the non-adaptive setting, the marginal gain of an element depends on realization ϕ in the adaptive setting. Therefore, we consider the expected value of the marginal gain with respect to the randomness of ϕ. The expected marginal gain of element $v \in V$ when partial realization ψ has been observed so far is defined as

$$\Delta_{f,p}(v|\psi) \coloneqq \mathbb{E}[f(\mathrm{dom}(\psi) \cup \{v\}, \Phi) - f(\mathrm{dom}(\psi), \Phi) | \Phi \sim \psi],$$

where $\mathrm{dom}(\psi) \coloneqq \{v \in V \mid \exists y \in \mathcal{Y},\ (v, y) \in \psi\}$. We omit the subscripts if they are clear from the context. We write $\Phi \sim \psi$ if Φ is generated from the posterior distribution $p(\phi|\psi)$. Given current realization ψ, the expected marginal gain $\Delta(v|\psi)$ represents the expected increase in the objective value yielded by selecting v.

We can extend this notation to a set, not a single element. Here we consider adding the elements in a set $S \subseteq V$ without observing the states of each element in S when having observed partial realization ψ so far. The expected marginal gain of set $S \subseteq V$ with partial realization ψ is defined as

$$\Delta_{f,p}(S|\psi) \coloneqq \mathbb{E}[f(\mathrm{dom}(\psi) \cup S, \Phi) - f(\mathrm{dom}(\psi), \Phi) | \Phi \sim \psi].$$

We can further extend the notation to the expected marginal gain of policies. Here we consider executing a policy π when having observed ψ so far. The expected marginal gain of policy π with partial realization ψ is defined as

$$\Delta_{f,p}(\pi|\psi) \coloneqq \mathbb{E}[f(\mathrm{dom}(\psi) \cup E(\pi, \Phi), \Phi) - f(\mathrm{dom}(\psi), \Phi) | \Phi \sim \psi].$$

2.3.2 Adaptive Submodularity and Adaptive Monotonicity

Adaptive submodularity, which is an adaptive extension of submodularity, is the property of diminishing returns of the expected marginal gain. Formally, adaptive submodularity is defined as follows.

Definition 30 (Adaptive submodularity [Golovin and Krause, 2011a])**.** Let $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ be a set function and p a distribution of ϕ. We say f is adaptive submodular with respect to p if for any partial realization $\psi \subseteq \psi'$ such that $p(\psi') > 0$ and any element $v \in V \setminus \mathrm{dom}(\psi')$, it holds that

$$\Delta(v|\psi) \geq \Delta(v|\psi').$$

Note that adaptive submodularity is defined relative to the distribution $p(\phi)$ over realizations, that is, f can be adaptive submodular with respect to one distribution, but not with respect to another. The monotonicity can also be extended to the adaptive setting as follows.

Definition 31 (Adaptive monotonicity [Golovin and Krause, 2011a])**.** Let $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ be a set function and p a distribution of ϕ. We say f is adaptive monotone with respect to p if for any partial realization ψ such that $p(\psi') > 0$ and any element $v \in V \setminus \mathrm{dom}(\psi)$, it holds that

$$\Delta(v|\psi) \geq 0.$$

Algorithm 6 Adaptive greedy algorithm with α-approximate greedy selection [Golovin and Krause, 2011a]

Input The objective function $f\colon 2^V \times \mathcal{Y}^V$ and the probability distribution $p \in \triangle^{\mathcal{Y}^V}$ given by a value oracle for the expected marginal gain $\Delta(\cdot|\cdot)$, the independence system $(V, \mathcal{I})$ given by an independence oracle.

$\psi_0 \leftarrow \varnothing$.
$F \leftarrow V$.
$i \leftarrow 0$.
while $F \neq \varnothing$ **do**
  $i \leftarrow i + 1$.
  Find $v \in V$ such that $\Delta(v|\psi_{i-1}) \geq \alpha \max_{v \in V} \Delta(v|\psi_{i-1})$.
  Observe $\phi(v)$ and let $\psi_i \leftarrow \psi_{i-1} \cup \{(v, \phi(v))\}$.
  Let $F \leftarrow \{v \in V \mid \mathrm{dom}(\psi_i) \cup \{v\} \in \mathcal{I}\}$.

Let $\pi'@\pi$ be a concatenated policy, i.e., a policy that executes π' as if from scratch after executing π. Adaptive monotonicity is known to be equivalent to the following condition:

Lemma 32 ([Golovin and Krause, 2011a, Lemma A.8]). *Fix $f : 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$. Then we have $\Delta(v|\psi) \geq 0$ for all ψ with $p(\psi) > 0$ and all $v \in V$ if and only if for all policies π and π', we have $f_{\mathrm{avg}}(\pi) \leq f_{\mathrm{avg}}(\pi'@\pi)$.*

The adaptive greedy algorithm [Golovin and Krause, 2011a] starts with the empty set and selects an element with the largest expected marginal gain at each step. Here we consider an approximate version that selects an element whose expected marginal gain is at least α times the maximum expected marginal gain. This algorithm can be regarded a counterpart of the greedy algorithm in the adaptive setting. The algorithmic description is given in Algorithm 6. Under the assumption of adaptive submodularity and adaptive monotonicity, the adaptive greedy algorithm is guaranteed to approximate the expected objective value achieved by an optimal policy.

Theorem 33 ([Golovin and Krause, 2011a]). *Suppose $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ is adaptive submodular and adaptive monotone with respect to p and $\mathcal{I}$ is a cardinality constraint. Let π be the policy obtained by executing the adaptive greedy with an α-approximate greedy selection k steps and π^* be any policy of height k. The objective value achieved by π is at least $(1 - \mathrm{e}^{-\alpha})$-approximation to the objective value achieved by π^*, i.e.,*

$$f_{\mathrm{avg}}(\pi) \geq (1 - \mathrm{e}^{-\alpha}) f_{\mathrm{avg}}(\pi^*).$$

Adaptive submodular coverage. The coverage version of adaptive submodular maximization is called *adaptive submodular coverage* [Golovin and Krause, 2011a]. Since the coverage version is more suitable for several applications including active learning, it has drawn much attention. The main question on this problem is what is the approximation ratio of the adaptive greedy algorithm that continues to select elements until the objective value achieves the threshold. To analyze this problem, Golovin and Krause [2011a] defined a stronger version of adaptive monotonicity called *strong adaptive monotonicity* as follows.

Definition 34 ([Golovin and Krause, 2011a]). Let $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ be a set function and p a distribution of ϕ. We say f is adaptive monotone with respect to p if for any partial realization ψ, any element $v \in V \setminus \mathrm{dom}(\psi)$, and any state $y \in \mathcal{Y}$ such that such that $\Pr(\Phi \sim \psi,\ \Phi(v) = y) > 0$, it holds that

$$\mathbb{E}[f(\mathrm{dom}(\psi), \Phi) | \Phi \sim \psi] \leq \mathbb{E}[f(\mathrm{dom}(\psi) \cup \{v\}, \Phi) | \Phi \sim \psi,\ \Phi(v) = y].$$

First, Golovin and Krause [2011a] argued that under the assumption of adaptive submodularity and strong adaptive monotonicity, the adaptive greedy algorithm is $(\ln \frac{Q}{\eta\delta} + 1)$-approximation in comparison to an optimal policy, where η is any value such that $f(S, \phi) > Q - \eta$ implies $f(S, \phi) = Q$ and $\delta = \min_\phi p(\phi)$. However, an error was found in the proof of this result by Nan and Saligrama [2017]. In response to this, the authors of Golovin and Krause [2011a] provided a slightly weaker result with a slightly stronger assumption in the updated arxiv version [Golovin and Krause, 2010]. The assumption newly added is *strong adaptive submodularity*, which is a stronger condition than adaptive submodularity defined as follows.

Definition 35 (Strong adaptive submodularity [Golovin and Krause, 2010])**.** We say f is strongly adaptive submodular with respect to p if f is adaptive submodular with respect to p and for any partial realization $\psi \subseteq \psi'$ and any policy $v \in V \setminus \mathrm{dom}(\psi')$, it holds that

$$\Delta(v|\psi; \psi') \geq \Delta(v|\psi'),$$

where $\Delta(v|\psi; \psi')$ is the extended expected marginal gain defined as

$$\Delta(v|\psi; \psi') = \mathbb{E}_\Phi[f(\mathrm{dom}(\psi) \cup \{v\}, \Phi) - f(\mathrm{dom}(\psi), \Phi)|\Phi \sim \psi'].$$

With the assumption of strong adaptive submodularity, the corrected result states $(\ln \frac{Q}{\eta\delta} + 1)^2$-approximation of the adaptive greedy algorithm. Here we consider the case where $f(V, \phi) = Q$ for all ϕ. If it does not hold, by truncating the function $\tilde{f}(S, \phi) = \min\{f(S, \phi), Q\}$ instead of f, we can reduce to such an instance.

Theorem 36 ([Golovin and Krause, 2010])**.** *Suppose $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ is strongly adaptive submodular and strongly adaptive monotone with respect to p and there exists Q such that $f(V, \phi) = Q$. Let η be any value such that $f(S, \phi) > Q - \eta$ implies $f(S, \phi) = Q$ for all S and ϕ. Let $\delta = \min_\phi p(\phi)$ be the minimum probability of any realization. Let π^* be an optimal policy minimizing the expected cost to guarantee $f(E(\pi^*, \phi), \phi) = Q$ for every realization ϕ. Let π be a policy that encodes α-approximate greedy algorithm. Then in general*

$$c_{\mathrm{avg}}(\pi) \leq \alpha c_{\mathrm{avg}}(\pi^*) \left(\ln\left(\frac{Q}{\delta\eta}\right) + 1\right)^2.$$

Remark 37. In the proof in Golovin and Krause [2010], the assumption that $f(S, \phi) = f(S, \phi')$ if $\phi(v) = \phi'(v)$ for any $v \in S$ seems to be implicitly used. This assumption holds in most applications, therefore it is not problematic.

p-System constraints. The adaptive greedy algorithm works for p-system constraints as well. Golovin and Krause [2011b] showed that the adaptive greedy algorithm achieves $1/(p+1)$-approximation for a p-system constraint.

Theorem 38 ([Golovin and Krause, 2011b])**.** *Suppose $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ is adaptive submodular and adaptive monotone with respect to p and $\mathcal{I}$ is the independence set family of a matroid. Let π be the policy obtained by executing the adaptive greedy with an α-approximate greedy selection k steps and π^* be any policy of height k. The objective value achieved by π is at least $(1 - \mathrm{e}^{-\alpha})$-approximation to the objective value achieved by π^*, i.e.,*

$$f_{\mathrm{avg}}(\pi) \geq \frac{\alpha}{p + \alpha} f_{\mathrm{avg}}(\pi^*).$$

Pointwise submodularity and pointwise monotonicity. Golovin and Krause [2011a] also defined submodularity for each realization ϕ. If each set function $f(\cdot, \phi)$ for ϕ satisfies submodularity, f is called *pointwise submodular*.

Definition 39 (Pointwise submodularity [Golovin and Krause, 2011a])**.** A set function $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ is *pointwise submodular* if for any realization ϕ, $f(\cdot, \phi)$ is submodular.

An interesting point is that pointwise submodularity does not imply adaptive submodularity, and vice versa. It is known that if adaptive submodularity and pointwise submodularity hold, then strong adaptive submodularity also holds [Golovin and Krause, 2010]. Similarly, *pointwise monotonicity* can be defined as follows.

Definition 40 (Pointwise monotonicity)**.** A set function $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ is *pointwise monotone* if for any realization ϕ, $f(\cdot, \phi)$ is monotone.

Though pointwise submodularity does not imply adaptive submodularity, pointwise monotonicity implies adaptive monotonicity. Moreover, pointwise monotonicity implies strong adaptive monotonicity.

3 Approximation Guarantees of Greedy Policy with Adaptive Submodularity Ratio

This chapter is organized as follows. Section 3.1 illustrates the background and overview of this chapter. In Section 3.2, we formally define the adaptive submodularity ratio, which is the key concept of this study. In Sections 3.3 and 3.4, we provide bounds on the approximation ratio of the adaptive greedy algorithm and adaptivity gaps, respectively, by using the adaptive submodularity ratio. In Sections 3.5 and 3.6, we apply the frameworks developed in Sections 3.3 and 3.4 to two applications: adaptive influence maximization and adaptive feature selection. In Section 3.7, we experimentally check the performance of the adaptive greedy algorithm in several applications. In Section 3.8 we review related work. Section 3.9 provides a summary and future work of this chapter.

3.1 Background and Overview

As illustrated in Section 2.3, the approach based on *adaptive submodularity* [Golovin and Krause, 2011a] is a well-established framework for analyzing greedy algorithms for adaptive optimization problems in machine learning. However, adaptive submodularity is not omnipotent. While the greedy policy works well for various sequential decision-making problems, many of these problems do not have adaptive submodularity. In fact, even if an objective function is submodular in the non-adaptive setting, its adaptive version does not always have adaptive submodularity.

Adaptive influence maximization is one such example. In this problem, a decision-maker aims at spreading information about a product by selecting several advertisements. She repeatedly alternates between selecting an advertisement and observing its effect. The objective function of this problem is known to have adaptive submodularity in the independent cascade model [Golovin and Krause, 2011a], but not in a more general diffusion model called the *triggering model* [Kempe et al., 2015], which is extensively studied as an important class of diffusion models [Leskovec et al., 2007b, Tang et al., 2014]. Note that this objective function satisfies submodularity in the non-adaptive setting, while it does not satisfy adaptive submodularity in the adaptive setting. Examples of other problems lacking adaptive submodularity appear in many applications such as feature selection and active learning. Therefore, we are waiting for an analysis framework that goes beyond adaptive submodularity.

To develop such a framework for the adaptive setting, we build on the the submodularity ratio [Das and Kempe, 2011] in the non-adaptive setting. As described in Section 2.2, the submodularity ratio is a prevalent tool for handling non-submodular functions. An adaptive variant of the submodularity ratio would be a promising approach to handling functions that lack adaptive submodularity, but how to define it is quite non-trivial since there is a large discrepancy between the non-adaptive and adaptive settings as exemplified above. In particular, success in defining an adaptive version of the submodularity ratio involves meeting the following two requirements: it must yield an approximation guarantee of the greedy policy, and it must be bounded in various important applications such as the adaptive influence maximization and adaptive feature selection. Previous studies [Kusner, 2014, Yong et al., 2017] tried to define similar notions, but none of them meet the requirements.

In this chapter, we invent a new notion of approximate submodularity for adaptive optimization, which we name adaptive submodularity ratio. We propose an analysis framework, *adaptive submodularity ratio*,

Table 3.1: Summary of our theoretical results about adaptive bipartite influence maximization and adaptive feature selection. We show lower bounds for the adaptive submodularity ratios, the approximation ratios of the adaptive greedy algorithm, and the adaptivity gaps. Let $\lambda_{\min,\ell} = \min_{\phi} \min_{S \subseteq V:\, |S| \leq \ell} \lambda_{\min}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)$ and $\lambda_{\max,\ell} = \max_{\phi} \max_{S \subseteq V:\, |S| \leq \ell} \lambda_{\max}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)$. Parameters q and d are determined by the diffusion model and the underlying graph structure. The results of Golovin and Krause [2011a] are indicated by †.

Problem	Adaptive submodularity ratio	Adaptive greedy	Adaptivity gaps
Linear threshold	$(k+1)/2k$	$1 - \exp(-(k+1)/2k)$	$(k+1)/2k$
Independent cascade	$1^\dagger$	$1 - 1/\mathrm{e}^\dagger$	$(1-q)^{\min\{d,k\}-1}$
Triggering	$(k+1)/2k$	$1 - \exp(-(k+1)/2k)$	
Feature selection	$\lambda_{\min,k+\ell}$	$1 - \exp(-\lambda_{\min,k+\ell})$	$\lambda_{\min,k}/\lambda_{\max,k}$

that meets the aforementioned requirements. An advantage of our proposal is that it has the potential to yield various theoretical results as in Table 3.1. Below we summarize our main contributions.

- We propose the definition of the adaptive submodularity ratio and, by using it, we prove an approximation guarantee of the adaptive greedy algorithm.
- We give a bound on the *adaptivity gap*, which represents the superiority of adaptive policies over non-adaptive policies, through the lens of the adaptive submodularity ratio.
- We provide lower bounds on the adaptive submodularity ratio for two important applications: adaptive influence maximization on bipartite graphs in the triggering model and adaptive feature selection. Regarding the former one, we show that our result is tight.
- Experiments confirm that the greedy policy performs well for the considered applications.

3.2 Adaptive Submodularity Ratio

In this section, we provide a precise definition of the adaptive submodularity ratio, which extends the submodularity ratio from the non-adaptive setting to the adaptive setting. We need to define it carefully so that it can yield an approximation guarantee of the greedy policy. An important point is to generalize subset S of size at most k, used to define the submodularity ratio, to policy π of height at most k.

Definition 41 (Adaptive submodularity ratio)**.** Suppose that $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ is adaptive monotone with respect to a distribution p. The adaptive submodularity ratio $\gamma_{\psi,k} \in [0,1]$ of f and p with respect to partial realization ψ and parameter $k \in \mathbb{Z}_{\geq 0}$ is defined to be

$$\gamma_{\psi,k}(f,p) \coloneqq \min_{\psi' \subseteq \psi,\ \pi \in \Pi_k} \frac{\sum_{v \in V} \Pr(v \in E(\pi,\Phi) | \Phi \sim \psi') \Delta(v|\psi')}{\Delta(\pi|\psi')}.$$

We omit f and p if they are clear from the context. We also define $\gamma_{\ell,k} \coloneqq \min_{\psi:|\psi| \leq \ell} \gamma_{\psi,k}$.

Intuitively, the adaptive submodularity ratio indicates the distance between (f,p) and the class of adaptive submodular functions. As with the non-adaptive setting, $\gamma_{\psi,k}(f,p) = 1$ implies the adaptive submodularity of f, which can formally be written as follows.

Proposition 42. *It holds that $\gamma_{\psi,k}(f,p)=1$ for any partial realization ψ and $k\in\mathbb{Z}_{\geq 0}$ if and only if f is adaptive submodular with respect to p.*

Proof. First, we deal with the "if" part. Let ψ_v be the partial realization just before v is selected in π. If there are multiple partial realizations ψ such that $\pi(\psi)=v$, we can duplicate v and take them to be different elements. From adaptive submodularity, for any partial realization ψ and policy π, we have

$$\begin{aligned}\Delta(\pi|\psi)&=\sum_{v\in V}\Pr(v\in E(\pi,\Phi)|\Phi\sim\psi)\Delta(v|\psi\cup\psi_v)\\&\leq\sum_{v\in V}\Pr(v\in E(\pi,\Phi)|\Phi\sim\psi)\Delta(v|\psi).\end{aligned}$$

Thus we can see $\gamma_{\psi,k}\geq 1$. Moreover, if π is a policy that selects a single element, the above inequality holds with equality. These two facts imply $\gamma_{\psi,k}=1$.

Next, we deal with the "only if" part. Let $\psi\subseteq\psi'$ be any partial realization such that $|\psi|+1=|\psi'|$ and $v\in V\setminus\mathrm{dom}(\psi')$ be any element. We define $u\in\mathrm{dom}(\psi')\setminus\mathrm{dom}(\psi)$ to be the additional element and y its state in ψ', i.e., $\psi'=\psi\cup\{(u,y)\}$. Let us consider a policy π that first selects u and, if $\phi(u)=y$, proceeds to select v. From the assumption, we have $\gamma_{\psi,2}=1$, and thus $\Delta(\pi|\psi)\leq\sum_{v\in V}\Pr(v\in E(\pi,\Phi))\Delta(v|\psi)$. We can calculate the left and right hand sides as follows.

$$\begin{aligned}\text{(LHS)}&=\Delta(u|\psi)+\Pr(\Phi(u)=y|\Phi\sim\psi)\Delta(v|\psi'),\\\text{(RHS)}&=\Delta(u|\psi)+\Pr(\Phi(u)=y|\Phi\sim\psi)\Delta(v|\psi).\end{aligned}$$

Therefore, we obtain $\Delta(v|\psi')\leq\Delta(v|\psi)$. By sequentially concatenating inequalities of this type, we can show that the statement holds for any $\psi\subseteq\psi'$. □

3.3 Adaptive Greedy Algorithm

In this section, we present a new approximation guarantee for the adaptive greedy algorithm based on the adaptive submodularity ratio. Thanks to this result, once the adaptive submodularity ratio is bounded, we can obtain approximation guarantees of the adaptive greedy algorithm for various applications. The adaptive greedy algorithm is an algorithm that starts with an empty set and repeatedly selects the element with the largest expected marginal gain. The detailed description is given in Algorithm 6. Golovin and Krause [2011a] have shown that this algorithm achieves $(1-1/\mathrm{e})$-approximation to the expected objective value of an optimal policy if f is adaptive submodular with respect to p. Here we extend their result and show that the adaptive greedy algorithm achieves $(1-\exp(-\gamma_{\ell,\ell}))$-approximation, where ℓ is the number of selected elements. More precisely, we can bound the approximation ratio relative to any policy π^* of height k as follows.

Theorem 43. *Suppose $f\colon 2^V\times\mathcal{Y}^V\to\mathbb{R}_{\geq 0}$ is adaptive monotone with respect to p. Let π be a policy representing the adaptive greedy algorithm until ℓ step. Then, for any policy $\pi^*\in\Pi_k$, it holds that*

$$f_{\mathrm{avg}}(\pi)\geq\left(1-\exp\left(-\frac{\gamma_{\ell,k}\ell}{k}\right)\right)f_{\mathrm{avg}}(\pi^*),$$

where $\gamma_{\ell,k}$ is the adaptive submodularity ratio of f with respect to p.

Proof. Let ψ be any possible partial realization that can appear while executing the adaptive greedy policy π. Since π stops after ℓ steps, we have $|\psi|\leq\ell$. According to the definition of the adaptive submodularity ratio, we have

$$\gamma_{\ell,k}\Delta(\pi^*|\psi)\leq\sum_{v\in V}\Pr(v\in E(\pi^*,\Phi)|\Phi\sim\psi)\Delta(v|\psi)\leq k\max_{v\in V}\Delta(v|\psi)$$

since $\sum_{v \in V} \Pr(v \in E(\pi^*, \Phi) | \Phi \sim \psi) = \mathbb{E}[|E(\pi^*, \Phi)|] \leq k$. Let Ψ be a random partial realization observed by executing $\pi_{[i]}$, where $\pi_{[i]}$ is a policy obtained by running π until it terminates or it selects i elements. Formally, Ψ conforms to the distribution $p_\Psi(\psi) \coloneqq \Pr(\Psi = \psi \mid \exists \phi,\ \psi = \{(v, \phi(v)) \mid v \in E(\pi_{[i]}, \phi)\})$. Then we can obtain a lower bound on the expected single step gain as follows.

$$
\begin{aligned}
f_{\mathrm{avg}}(\pi_{[i+1]}) - f_{\mathrm{avg}}(\pi_{[i]}) &= \mathbb{E}\left[\max_{v \in V} \Delta(v | \Psi)\right] \\
&\qquad \text{(due to the property of the adaptive greedy algorithm)} \\
&\geq \mathbb{E}\left[\frac{\gamma_{\ell,k}}{k} \Delta(\pi^* | \Psi)\right] \qquad \text{(due to (3.3))} \\
&= \frac{\gamma_{\ell,k}}{k} \left(f_{\mathrm{avg}}(\pi_{[i]} @ \pi^*) - f_{\mathrm{avg}}(\pi_{[i]})\right) \\
&\geq \frac{\gamma_{\ell,k}}{k} \left(f_{\mathrm{avg}}(\pi^*) - f_{\mathrm{avg}}(\pi_{[i]})\right). \\
&\qquad \text{(due to adaptive monotonicity and Lemma 32)}
\end{aligned}
$$

Let $\Delta_i \coloneqq f_{\mathrm{avg}}(\pi^*) - f_{\mathrm{avg}}(\pi_{[i]})$. The above inequality can be rewritten as $\Delta_i - \Delta_{i+1} \geq \gamma_{\ell,k} \Delta_i / k$, which implies $\Delta_{i+1} \leq (1 - \gamma_{\ell,k}/k) \Delta_i$. By repeatedly using this inequality, we obtain $\Delta_\ell \leq (1 - \gamma_{\ell,k}/k)^\ell \Delta_0 \leq \exp(-\gamma_{\ell,k} \ell / k) f_{\mathrm{avg}}(\pi^*)$. Consequently, we have $f_{\mathrm{avg}}(\pi) \geq (1 - \exp(-\gamma_{\ell,k} \ell / k)) f_{\mathrm{avg}}(\pi^*)$. □

3.4 Non-adaptive Policies and Adaptivity Gaps

We show that the adaptive submodularity ratio is also useful for theoretically comparing the performances of adaptive and non-adaptive policies. More precisely, we present a lower bound on the *adaptivity gap*, which represents the performance gap between adaptive and non-adaptive polices, by using the adaptive submodularity ratio. The adaptivity gap is defined as follows.

Definition 44 (Adaptivity gaps)**.** The adaptivity gap $\mathsf{GAP}_k(f, p)$ of an objective function $f \colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ and a probability distribution p of $\phi \colon V \to \mathcal{Y}$ is defined as the ratio between an optimal adaptive policy and an optimal non-adaptive policy, i.e.,

$$
\mathsf{GAP}_k(f, p) = \frac{\max_{M \colon |M| \leq k} \mathbb{E}_\Phi[f(M, \Phi)]}{\max_{\pi^* \in \Pi_k} f_{\mathrm{avg}}(\pi^*)},
$$

where k is the height of adaptive and non-adaptive policies.

Theorem 45. *Let $f \colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ be an objective function and p a probability distribution of $\phi \colon V \to \mathcal{Y}$. Let $\gamma_{\varnothing,k}$ be the adaptive submodularity ratio of f with respect to p. Let $\beta_{\varnothing,k}$ be the supermodularity ratio of the set function $\mathbb{E}_\Phi[f(\cdot, \Phi)]$ of non-adaptive policies. We have*

$$
\mathsf{GAP}_k(f, p) \geq \beta_{\varnothing,k} \gamma_{\varnothing,k}.
$$

Proof of Theorem 45. Let π^*_{non} be an optimal non-adaptive policy and π^* be an optimal adaptive policy. Since π^*_{non} is a non-adaptive policy, it selects the same subset for all ϕ, i.e., $E(\pi^*_{\mathrm{non}}, \phi) = E(\pi^*_{\mathrm{non}}, \phi')$ for all ϕ and ϕ'. Let $M \in \operatorname{argmax} \sum_{v \in M \colon |M| \leq k} \Delta(v | \varnothing)$ and π^M_{non} the non-adaptive policy that selects M. From the optimality of π^*_{non}, we have

$$
f_{\mathrm{avg}}(\pi^*_{\mathrm{non}}) \geq f_{\mathrm{avg}}(\pi^M_{\mathrm{non}}).
$$

By the definition of the supermodularity ratio, we have

$$\Delta(\pi^M_{\mathrm{non}}|\varnothing) \geq \beta_{\varnothing,k} \sum_{v\in M} \Delta(v|\varnothing).$$

Note that $\sum_{v\in V} \Pr(v \in E(\pi^*, \Phi)) \leq k$ and $\Pr(v \in E(\pi^*, \Phi)) \leq 1$ for each $v \in V$. Due to the definition of M, we have

$$\sum_{v\in M} \Delta(v|\varnothing) \geq \sum_{v\in V} \Pr(v \in E(\pi^*, \Phi))\Delta(v|\varnothing).$$

From the definition of the adaptive submodularity ratio, we have

$$\sum_{v\in V} \Pr(v \in E(\pi^*, \Phi))\Delta(v|\varnothing) \geq \gamma_{\varnothing,k}\Delta(\pi^*|\varnothing).$$

Combining these inequalities, we have

$$\begin{aligned} f_{\mathrm{avg}}(\pi^*_{\mathrm{non}}) &\geq \mathbb{E}_\Phi[f(\varnothing, \Phi)] + \Delta(\pi^M_{\mathrm{non}}|\varnothing) \\ &\geq \beta_{\varnothing,k}\gamma_{\varnothing,k}(\mathbb{E}_\Phi[f(\varnothing, \Phi)] + \Delta(\pi^*|\varnothing)) \\ &= \beta_{\varnothing,k}\gamma_{\varnothing,k} f_{\mathrm{avg}}(\pi^*). \end{aligned}$$

□

Therefore, given any non-adaptive α-approximation algorithm, we can evaluate its performance relative to an optimal adaptive policy as follows.

Corollary 46. *Let $\pi_{\mathrm{non}} \in \Pi_k$ be a non-adaptive policy that achieves α-approximation to an optimal non-adaptive policy π^*_{non}. Let $\gamma_{\varnothing,k}$ be the adaptive submodularity ratio of f with respect to p. Let $\beta_{\varnothing,k}$ be the supermodularity ratio of the non-adaptive objective function $\mathbb{E}_\Phi[f(\cdot, \Phi)]$. Let π^* be an optimal adaptive policy. We have*

$$f_{\mathrm{avg}}(\pi_{\mathrm{non}}) \geq \alpha\beta_{\varnothing,k}\gamma_{\varnothing,k} f_{\mathrm{avg}}(\pi^*).$$

Proof of Corollary 46. From the approximation ratio, we have

$$f_{\mathrm{avg}}(\pi_{\mathrm{non}}) \geq \alpha f_{\mathrm{avg}}(\pi^*_{\mathrm{non}}).$$

From Theorem 45, we have

$$f_{\mathrm{avg}}(\pi^*_{\mathrm{non}}) \geq \beta_{\varnothing,k}\gamma_{\varnothing,k} f_{\mathrm{avg}}(\pi^*).$$

The above two inequalities imply the statement. □

From the following example, we can see that Theorem 45 is tight, i.e., for any rationals β and γ in $(0, 1]$, there exist f and p such that the equality holds.

Example 47. Let $V = \{u\} \cup \bigcup_{i=1}^M V_i$ be the ground set, where $V_i = \{v_i^1, \cdots, v_i^k\}$. Let $V_0 = \varnothing$. Let $\mathcal{Y} = \{0, 1, \cdots, M\}$. We define the probability distribution p such that $\phi(u) = y \in \mathcal{Y}$ with probability $p \in [0, 1/M]$ for each $y \neq 0$ and $\phi(u) = 0$ with probability $1 - pM$. Other elements always in state 0, i.e., $\phi(v) = 0$ with probability 1 for all $v \in V \setminus \{u\}$. We define the objective function f as

$$f(S, \phi) = \begin{cases} 1 + a|S \cap V_{\phi(u)}| & (u \in S) \\ 1 + ap(|S| - 1) & (u \notin S \text{ and } |S| \geq 1) \\ 0 & (S = \varnothing), \end{cases}$$

where $a \in \mathbb{R}_{\geq 0}$ is the parameter specified later. We have $\Delta(v|\varnothing) = 1$ for all $v \in V$. The supermodularity ratio $\beta_{\varnothing,k}$ of $\mathbb{E}[f(\cdot, \Phi)]$ is

$$\beta_{\varnothing,k} = \frac{1 + (k-1)ap}{k}.$$

The adaptive submodularity ratio $\gamma_{\varnothing,k}$ is

$$\gamma_{\varnothing,k} = \frac{k}{1 + (k-1)apM}.$$

The adaptivity gap is

$$\mathsf{GAP}_k(f, p) = \frac{1 + (k-1)ap}{1 + (k-1)apM}.$$

For any rationals $\beta \in (0, 1]$ and $\gamma \in (0, 1]$, there exist some k, a, M such that $\gamma_{\varnothing,k} = \gamma$ and $\beta_{\varnothing,k} = \beta$.

3.5 Adaptive Influence Maximization

In this section, we consider adaptive influence maximization on bipartite graphs. We provide a bound on the adaptive submodularity ratio in the case of the triggering model, and we show that this result is tight. We also present bounds on the adaptivity gaps in the case of the independent cascade and linear threshold models by using the adaptive submodularity ratio.

Let $G = (V \cup U, A)$ be a directed bipartite graph with source vertices V, sink vertices U, and directed edges $A \subseteq V \times U$. In the case of bipartite influence model [Alon et al., 2012], this graph represents the relationship between advertisements V and customers U. We consider the problem of selecting several advertisements $S \subseteq V$ to make as much influence as possible on the customers. Here, each edge is determined to be alive or dead according to a certain distribution, and influence can be spread only through live edges. Given vertex weights $w\colon U \to \mathbb{R}_{\geq 0}$, the objective function to be maximized is $f(X) = \sum_{u \in \bigcup_{v \in X} R(v)} w(u)$, where, for each $v \in V$, $R(v) \subseteq U$ represents a set of vertices that are reachable from v by going through only live edges. In the adaptive version of influence maximization, at each step, we select a vertex $v \in V$ and observe the states of all outgoing edges $(v, u) \in A$, while, in the non-adaptive setting, we select $S \subseteq V$ before observing the states of any edges.

We consider a general diffusion model called the *triggering model* [Kempe et al., 2015], which includes various important models such as the independent cascade model and the linear threshold model as special cases. In the triggering model, each vertex $v \in V$ is associated with some known probability distribution over the power set of incoming edges. According to this distribution, a subset of incoming live edges is determined. A vertex gets activated if and only if it is reachable from some selected vertex (or seed vertex) through only live edges. We aim to maximize the total weight of activated vertices by appropriately selecting seed vertices. Note that this objective function is submodular in the non-adaptive setting.

For later use, we explain the linear threshold model, a special case of the triggering model. In this model, the probability distribution on the incoming edges of each vertex is restricted so that each vertex has at most one live edge in any realization. In other words, there exists $b\colon A \to \mathbb{R}_{\geq 0}$ such that, for each $v \in V$, we have $\sum_{a \in \delta_-(v)} b(a) \leq 1$, where $\delta_-(v)$ is the full set of edges pointing to v, and $a \in A$ is alive with probability $b(a)$ exclusively over $\delta_-(v)$. In contrast to the linear threshold model, the triggering model accepts any distribution over the power set of $\delta_-(v)$.

3.5.1 Bound of Adaptive Submodularity Ratio

We first present the bound of the adaptive submodularity ratio. Here we provide a proof sketch, and the full proof is given in Section 3.5.3.

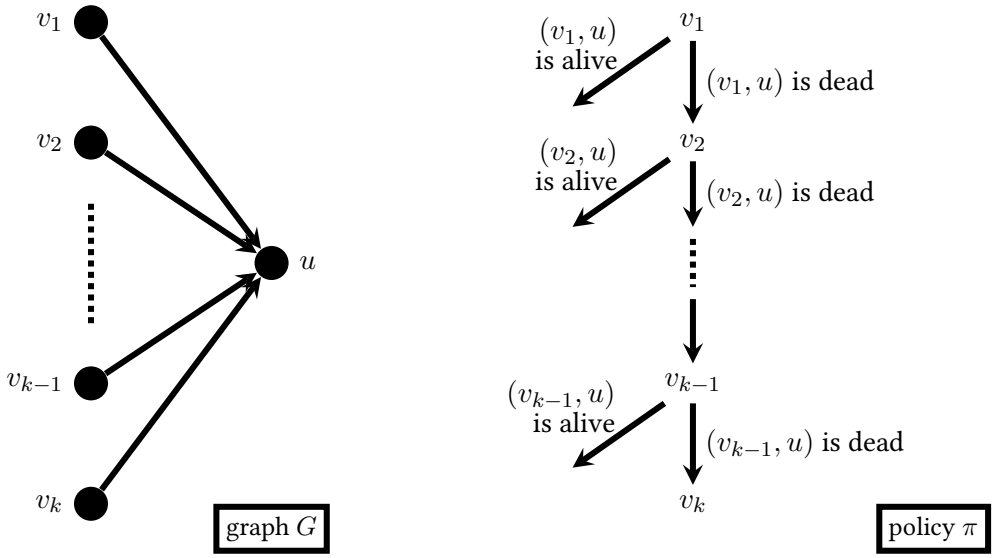


Figure 3.1: An example that implies the tightness of our bound.

Theorem 48. *Let G be an arbitrary directed bipartite graph and w be any weight function. For any $k \in \mathbb{Z}_{\geq 0}$ and partial realization ψ, the adaptive submodularity ratio $\gamma_{\psi,k}$ of the objective function and the distribution of the adaptive influence maximization in the triggering model is bounded as follows.*

$$\gamma_{\psi,k} \geq \frac{k+1}{2k}.$$

Proof sketch of Theorem 48. Since the objective function and the probability distribution of edge states can be decomposed into those defined for each vertex $u \in U$, it is sufficient to consider the case where $|U| = 1$.

Our goal is to prove

$$\begin{aligned}&\Delta(\pi|\psi') \\ &\leq \frac{2k}{k+1} \sum_{v \in V} \Pr(v \in E(\pi, \Phi) | \Phi \sim \psi') \Delta(v|\psi')\end{aligned}$$

for any observation ψ' and policy $\pi \in \Pi_k$. By duplicating $v \in V$ that appears multiple times in policy tree π, we can write the above inequality as

$$\sum_{v \in V} \Pr(v \in E(\pi, \Phi) | \Phi \sim \psi') \left(\frac{2k}{k+1} \Delta(v|\psi') - \Delta(v|\psi' \cup \psi_v) \right) \geq 0,$$

where ψ_v is the observation just before v is selected. We decompose the policy tree into the path wherein u remains inactive and the rest, and prove the inequality for each part separately. □

We can see that the above bound is tight even for the linear threshold model by considering the following example.

Example 49. Let G be a bipartite directed graph with $V = \{v_1, \ldots, v_k\}$, $U = \{u\}$, and $A = \{(v_i, u) \mid i \in [k]\}$. Let w be the vertex weight such that $w(u) = 1$. We consider the linear threshold model

in which an edge selected out of A uniformly at random is alive and the other edges are dead. We consider a simple policy π that selects all vertices one by one until u is activated. These graph and policy are illustrated in Figure 3.1. Since π finally activates u, the expected gain of π is $\Delta(\pi|\varnothing) = 1$. The probability that π selects each vertex is $\Pr(v_i \in E(\pi, \Phi)) = (k - i + 1)/k$. The expected marginal gain of v_i is $\Delta(v_i|\varnothing) = 1/k$. The adaptive submodularity ratio can be bounded as

$$\begin{aligned}\gamma_{\varnothing,k} &\leq \frac{\sum_{v \in V} \Pr(v \in E(\pi, \Phi))\Delta(v|\varnothing)}{\Delta(\pi|\varnothing)} \\ &\leq \sum_{i=1}^{k} \frac{k - i + 1}{k} \cdot \frac{1}{k} \\ &\leq \frac{k+1}{2k}.\end{aligned}$$

Hence the lower bound in Theorem 48 is tight.

The assumption that G is bipartite, considered in Theorem 48, may seem excessively strong, but it is actually a vital assumption. We show that, if G is not a bipartite graph, the adaptive submodularity ratio can be arbitrarily small; in fact, such an example can be constructed with the linear threshold model on a very simple graph G. We describe the details in Section 3.5.4.

3.5.2 Bound of Adaptivity Gap

Next, we provide a bound on the adaptivity gaps of bipartite influence maximization problems by using the adaptive submodularity ratio. First, we consider the independent cascade model. Since the adaptive submodularity holds for the independent cascade model [Golovin and Krause, 2011a], the adaptive submodularity ratio of its objective function is 1 by Proposition 42. In addition, by using a bound of the curvature [Maehara et al., 2017] and an inequality between the supermodularity ratio and the curvature [Bogunovic et al., 2018], we obtain $\beta_{\varnothing,k} \geq (1 - q)^{\min\{k,d\}-1}$, where q is an upper bound of the probability that each edge is alive and d is the largest degree of the vertex in V. From Theorem 45, we obtain the following result.

Proposition 50. *Let f be the objective function and p the probability distribution of bipartite influence maximization in the independent cascade model. We have*

$$\mathsf{GAP}_k(f, p) \geq (1 - q)^{\min\{k,d\}-1}.$$

We can derive a similar bound for the linear threshold model. Since the expected objective function is a linear function, its supermodularity ratio is 1. As a special case of Theorem 48, we have $\gamma_{\varnothing,k} \geq \frac{k+1}{2k}$. Combining these bounds with Theorem 45, we obtain the following result.

Proposition 51. *Let f be the objective function and p the probability distribution of bipartite influence maximization in the linear threshold model. We have*

$$\mathsf{GAP}_k(f, p) \geq \frac{k+1}{2k}.$$

3.5.3 Full Proofs for Adaptive Influence Maximization

In this subsection, we provide the full proof for Theorem 48. For the readability, we first give a proof for the case of the linear threshold model, which is a special case of the triggering model. After that, we give a proof for the case of the triggering model.

Proof for the Linear Threshold Model

Proof of Theorem 48 in the case of the linear threshold model. Let V be the source vertices, U the sink vertices, and $A \subseteq V \times U$ the directed edges. For notational simplicity, assume that $G = (V \cup U, A)$ is a complete bipartite graph, i.e., $A = V \times U$. By setting $b(a) = 0$ for all edges $a \in A$ that originally do not exist, we can assume this without loss of generality. Fix any $\psi' \subseteq \psi$ and $\pi \in \Pi_k$. It suffices to prove

$$\Delta(\pi|\psi') \leq \frac{2k}{k+1} \sum_{v \in V} \Pr(v \in E(\pi, \Phi)|\Phi \sim \psi')\Delta(v|\psi').$$

Let $\Delta_u(\cdot|\psi')$ be the expected marginal gain obtained by activating $u \in U$. Below we explain that the above inequality can be separated for each $u \in U$; i.e., it is enough to prove the above inequality for the case where $w(u) > 0$ for just one vertex $u \in U$ and 0 for the others. The objective function is the linear sum of the one for each $u \in U$: $\Delta(\cdot|\psi') = \sum_{u \in U} \Delta_u(\cdot|\psi')$. Therefore, the above inequality is decomposed into the sum of

$$\Delta_u(\pi|\psi') \leq \frac{2k}{k+1} \sum_{v \in V} \Pr(v \in E(\pi, \Phi)|\Phi \sim \psi')\Delta_u(v|\psi')$$

for each $u \in U$. Note that the states of any $(v, u) \in A$ and $(v', u') \in A$ are independent of each other if $u \neq u'$. Since the feedback about any $u' \in U$ such that $u' \neq u$ is never correlated with the states of edges pointing to u, we can regard the feedback about u' as an independent random factor when considering (3.5.3). Thus we can see that it is sufficient to consider the case of one sink vertex. Note that a randomized policy can be expressed as a linear sum of deterministic policies. Therefore, it is enough to consider the case where π is a deterministic policy. Below we fix $u \in U$ and use Δ instead of Δ_u for notational ease. We can assume $w(u) = 1$ without loss of generality. If u has been already activated in ψ', both sides of (3.5.3) are equal to zero; thus it holds trivially. We then consider the case where u is not activated in ψ'.

Let ψ_v be the partial realization just before v is selected in π. If there are multiple partial realizations ψ such that $\pi(\psi) = v$, we can duplicate v and consider them to be different elements. We can decompose $\Delta(\pi|\psi')$ as

$$\Delta(\pi|\psi') = \sum_{v \in V} \Pr(v \in E(\pi, \Phi)|\Phi \sim \psi')\Delta(v|\psi' \cup \psi_v).$$

The inequality that we aim to prove can be written as

$$\sum_{v \in V} \Pr(v \in E(\pi, \Phi)|\Phi \sim \psi') \left\{ \frac{2k}{k+1}\Delta(v|\psi') - \Delta(v|\psi' \cup \psi_v) \right\} \geq 0.$$

Since π is a deterministic policy that observes only states of edges pointing to u, there exists a path in policy tree π wherein u remains inactive; in Figure 3.2 such a path is colored in thin gray. Let $P = \{v_1, \cdots, v_m\} \subseteq V$ be the path, where $m \leq k$ and policy π selects the vertices $v_1, \cdots, v_m$ in this order. We consider proving the above inequality for P and $V \backslash P$ separately. We can easily see that $\Delta(v|\psi' \cup \psi_v) = 0$ holds for all $v \in V \setminus P$ since u is already activated there. Therefore, it is enough to prove

$$\sum_{v \in P} \Pr(v \in E(\pi, \Phi)|\Phi \sim \psi') \left\{ \frac{2k}{k+1}\Delta(v|\psi') - \Delta(v|\psi' \cup \psi_v) \right\} \geq 0.$$

Now we calculate the left hand side of this inequality, which we denote by C. Since u has not been activated yet in ψ', all edges (s, u) are dead for all $s \in \mathrm{dom}(\psi')$. In the linear threshold model, we can

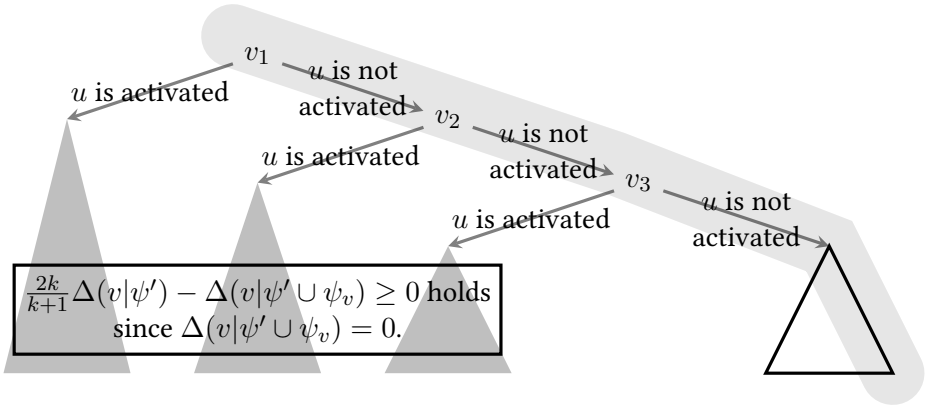


Figure 3.2: A description of our proof method. We can decompose the policy tree into the path wherein u is not activated and the rest.

define $p_i \coloneqq b(v_i u)/(1 - \sum_{t \in V \setminus \mathrm{dom}(\psi')} b(tu))$ to be the posterior probability that edge (v, u) is alive under observations ψ' for each $i = 1, \ldots, m$. Now we have $\Pr(v_i \in E(\pi, \Phi) | \Phi \sim \psi') = \Pr(\Phi \sim \psi' \cup \psi_{v_i} | \Phi \sim \psi') = 1 - \sum_{j=1}^{i-1} p_j$. In addition, we have $\Delta(v_i|\psi') = p_i$ and $\Delta(v_i|\psi' \cup \psi_{v_i}) = p_i/(1 - \sum_{j=1}^{i-1} p_j)$, and hence

$$C = \sum_{i=1}^{m} \left(1 - \sum_{j=1}^{i-1} p_j\right) \left\{ \frac{2k}{k+1} p_i - \frac{p_i}{1 - \sum_{j=1}^{i-1} p_j} \right\}.$$

In the case of $m = 1$, we have $C = (k-1)/(k+1) p_i \geq 0$. For $m \geq 2$, we obtain

$$\begin{aligned} C &= \sum_{i=1}^{m} \frac{2k}{k+1} p_i \left(1 - \sum_{j=1}^{i-1} p_j\right) - \sum_{i=1}^{m} p_i \\ &= \frac{k-1}{k+1} \sum_{i=1}^{m} p_i - \frac{2k}{k+1} \sum_{i=1}^{m} \left(p_i \sum_{j=1}^{i-1} p_j \right) \\ &= \frac{k-1}{k+1} \left\{ \sum_{i=1}^{m} p_i - \frac{2k}{k-1} \sum_{i=1}^{m} \left(p_i \sum_{j=1}^{i-1} p_j \right) \right\}. \end{aligned}$$

The right hand side can be bounded from below as

$$\begin{aligned} &\frac{k-1}{k+1} \left\{ \sum_{i=1}^{m} p_i - \frac{2k}{k-1} \sum_{i=1}^{m} \left(p_i \sum_{j=1}^{i-1} p_j \right) \right\} \\ &= \frac{k-1}{k+1} \left\{ \mathbf{1}^\top \mathbf{p} - \frac{k}{k-1} \mathbf{p}^\top (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \mathbf{p} \right\} \\ &\geq \frac{k-1}{k+1} \left\{ \mathbf{1}^\top \mathbf{p} - \frac{m}{m-1} \mathbf{p}^\top (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \mathbf{p} \right\}, \end{aligned}$$

where $\mathbf{p} = (p_1, \ldots, p_m)^\top$ and $\mathbf{I} \in \mathbb{R}^{m \times m}$ is the identity matrix. The inequality comes from $2 \leq m \leq k$ and $\mathbf{p}^\top (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \mathbf{p} \geq 0$. Since each entry of $\mathbf{p}$ represents a probability, we have $\mathbf{0} \leq \mathbf{p} \leq \mathbf{1}$ and $0 \leq \mathbf{1}^\top \mathbf{p} \leq 1$. From Lemma 52 proved below, we can see that this is non-negative. Therefore, we conclude that (3.5.3) holds. □

In the above proof, we used the following lemma.

Lemma 52. *Let $m \geq 2$ and $\mathbf{p} \in \mathbb{R}^m$ be an arbitrary vector such that $\mathbf{0} \leq \mathbf{p} \leq \mathbf{1}$ and $0 \leq \mathbf{1}^\top \mathbf{p} \leq 1$, then we have*

$$\mathbf{1}^\top \mathbf{p} - \frac{m}{m-1} \mathbf{p}^\top (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \mathbf{p} \geq 0.$$

Proof. Let $\mathbf{U} = (\mathbf{u}_1 \cdots \mathbf{u}_m) \in \mathbb{R}^{m \times m}$ be an orthonormal matrix whose first column is defined as $\mathbf{u}_1 = \mathbf{1}/\sqrt{m}$; we can write $\mathbf{p} = \mathbf{U}\mathbf{q}$ with some vector $\mathbf{q} = (q_1, \ldots, q_m)^\top$. Since $\mathbf{u}_1^\top \mathbf{u}_i = 0$ for all $i \neq 1$, we obtain $\mathbf{U}^\top \mathbf{1} = (\sqrt{m}, 0, \ldots, 0)^\top$. Hence the left hand side of the target inequality can be rewritten as

$$\begin{aligned}
\mathbf{1}^\top \mathbf{p} - \frac{m}{m-1} \mathbf{p}^\top (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \mathbf{p} &= \mathbf{1}^\top \mathbf{U}\mathbf{q} - \frac{m}{m-1} \mathbf{q}^\top \mathbf{U}^\top (\mathbf{1}\mathbf{1}^\top - \mathbf{I}) \mathbf{U}\mathbf{q} \\
&= \frac{m}{m-1} (\|\mathbf{q}\|_2^2 - m q_1^2) + \sqrt{m} q_1 \\
&= \sqrt{m} q_1 (1 - \sqrt{m} q_1) + \frac{m}{m-1} (q_2^2 + \cdots + q_m^2).
\end{aligned}$$

Since we have $0 \leq \mathbf{1}^\top \mathbf{p} = \sqrt{m} q_1 \leq 1$, this value is non-negative. □

Proof for the Triggering Model

Proof of Theorem 48. The outline of the proof for the triggering model is the same as the one for the linear threshold model. In the case of the triggering model, we can write C as follows.

$$C = \sum_{i=1}^{m} \Pr\left(\bigwedge_{j=1}^{i-1} X_j = 0 \middle| \Phi \sim \psi' \right) \left\{ \frac{2k}{k+1} \Pr\left(X_i = 1 | \Phi \sim \psi' \right) - \Pr\left(X_i = 1 \middle| \Phi \sim \psi', \bigwedge_{j=1}^{i-1} X_j = 0 \right) \right\},$$

where X_i is an event in which edge (v_i, u) is alive. Different from the linear threshold model, we cannot express C explicitly with parameters. Hence we define

$$\begin{aligned}
&p_i \coloneqq \Pr(X_i = 1 | \Phi \sim \psi') \quad \text{for } i = 1, \cdots, m, \\
&a_i \coloneqq \Pr\left(X_i = 1 \wedge \left\{ \textstyle\bigwedge_{j=1}^{i-1} X_i = 0 \right\} \middle| \Phi \sim \psi' \right) \quad \text{for } i = 1, \cdots, m, \\
\text{and} \quad &h_i \coloneqq \Pr\left(\left\{ \textstyle\bigwedge_{j=1}^{i} X_i = 0 \right\} \middle| \Phi \sim \psi' \right) \quad \text{for } i = 0, \cdots, m.
\end{aligned}$$

With these definitions, we can calculate C as

$$\begin{aligned}
C &= \sum_{i=1}^{m} h_{i-1} \left(\frac{2k}{k+1} p_i - \frac{a_i}{h_{i-1}} \right) \\
&= \frac{2k}{k+1} \sum_{i=1}^{m} p_i h_{i-1} - \sum_{i=1}^{m} a_i.
\end{aligned}$$

Our goal is to prove that

$$\frac{2k}{k+1} \sum_{i=1}^{m} p_i h_{i-1} - \sum_{i=1}^{m} a_i \geq 0.$$

Note that we have

$$0 \leq a_i \leq p_i \leq 1 \qquad \text{and} \qquad 0 \leq h_i \leq 1 \qquad \text{for } i = 1, \ldots, m,$$

where $a_1 = p_1$ and $h_0 = 1$. Therefore, if $m = 1$, we have

$$\begin{aligned}\frac{2k}{k+1}\sum_{i=1}^{m} p_i h_{i-1} - \sum_{i=1}^{m} a_i &= \frac{2k}{k+1} p_1 h_0 - a_1 \\ &= \frac{k-1}{k+1} p_1 \\ &\geq 0.\end{aligned}$$

Furthermore, it holds that

$$h_i + \textstyle\sum_{j=1}^{i} a_j = \Pr\left(\left\{\bigwedge_{j=1}^{i} X_i = 0\right\}\right) + \sum_{j=1}^{i} \Pr\left(X_i = 1 \wedge \left\{\bigwedge_{j=1}^{i-1} X_i = 0\right\}\right) = 1$$

for $i = 0, \ldots, m$, where $\sum_{j=1}^{0} a_j = 0$. By combining this equality with $0 \leq h_i \leq 1$, we obtain

$$0 \leq \sum_{j=1}^{i} a_j \leq 1$$

for $i = 0, \ldots, m$. With these inequalities, the light hand side of the target inequality can be bounded as

$$\begin{aligned}\frac{2k}{k+1}\sum_{i=1}^{m} p_i h_{i-1} - \sum_{i=1}^{m} a_i &= \frac{2k}{k+1}\sum_{i=1}^{m} p_i \left(1 - \sum_{j=1}^{i-1} a_j\right) - \sum_{i=1}^{m} a_i \\ &\geq \frac{2k}{k+1}\sum_{i=1}^{m} a_i \left(1 - \sum_{j=1}^{i-1} a_j\right) - \sum_{i=1}^{m} a_i \\ &= \frac{k-1}{k+1}\sum_{i=1}^{m} a_i - \frac{2k}{k+1}\sum_{i>j} a_i a_j \\ &= \frac{k-1}{k+1}\left(\mathbf{1}^\top \mathbf{a} - \frac{k}{k-1}\mathbf{a}^\top(\mathbf{1}\mathbf{1}^\top - \mathbf{I})\mathbf{a}\right) \\ &\geq \frac{k-1}{k+1}\left(\mathbf{1}^\top \mathbf{a} - \frac{m}{m-1}\mathbf{a}^\top(\mathbf{1}\mathbf{1}^\top - \mathbf{I})\mathbf{a}\right),\end{aligned}$$

which is non-negative from Lemma 52; this completes the proof as with the case of the linear threshold model. □

3.5.4 Example for the Case of General Graphs

In this subsection, we provide a problem instance of a general graph in which the adaptive submodularity ratio can be very small.

Before that, we briefly describe the problem setting of adaptive influence maximization in general graphs. Let $G = (V', A)$ be a general directed graph and $V \subseteq V'$ be a set of vertices that can be selected. At each step, the algorithm selects one vertex $v \in V$, then the influence spreads from v according to some stochastic diffusion process such as the independent cascade model or the linear threshold model. After that, the algorithm observes the diffusion from this vertex v under some feedback model. This problem includes the bipartite influence maximization as a special case where $G = (V \cup U, A)$ is a directed bipartite graph with $A \subseteq V \times U$ and $w(v) = 0$ for all $v \in V$.

There are two standard feedback models, both of which are proposed by Golovin and Krause [2011a]. Note that these two feedback models are equivalent in bipartite graphs. In the first feedback model

called the *myopic feedback model*, the algorithm observes the states of all edges outgoing from v. Golovin and Krause [2011a] proved that the adaptive submodularity does not hold in this case by giving a simple example. This analysis can be applied to both the independent cascade and linear threshold models. With this example instance, we can readily see that the adaptive submodularity ratio can be very small under the myopic feedback model. These facts imply that the myopic feedback model is typically too harsh to deal with.

In the second feedback model called the *full-adoption feedback model*, the algorithm observes the states of all edges outgoing from any vertex $u \in R(v)$ when selecting v, where $R(v)$ is the set of all vertices reachable from v only through live edges. Golovin and Krause [2011a] showed that, even if graphs are general (non-bipartite), the objective function satisfies adaptive submodularity under the independent cascade model with the full-adaption feedback.

Below we show that, even under the linear threshold model with the full-adoption feedback, the adaptive submodularity ratio can be arbitrarily small if the graph is non-bipartite. This fact implies that the assumption of bipartiteness, which we imposed to obtain the bound on the adaptive submodularity ratio, is almost inevitable.

Example 53. Let G be a directed graph with vertices $V = \{v_1, \dots, v_\ell\} \cup \{u_0, u_1, \dots, u_\ell\}$ and directed edges $A = \{(u_{i-1}, u_i) \mid i = 1, \dots, \ell\} \cup \{(v_i, u_i) \mid i = 1, \dots, \ell\}$. Let w be the vertex weight such that $w(v) = 1$ for all $v \in V$. We consider the following linear threshold model: for each $i \in [\ell]$, only one of the two edges, (v_i, u_i) and (u_{i-1}, u_i), entering u_i is alive with probability ϵ and $1 - \epsilon$, respectively.

Let π be a policy defined as follows. π first selects u_0. Then the realized states of some edges are revealed under the full-adoption feedback model and we can observe which vertices are activated. If u_ℓ is activated, π stops. Otherwise, there exists some $i \in [\ell]$ such that u_{i-1} is activated but u_i is not. Then π proceeds to select v_i. Repeat this procedure until u_ℓ is activated. The graph and policy are illustrated in Figure 3.3.

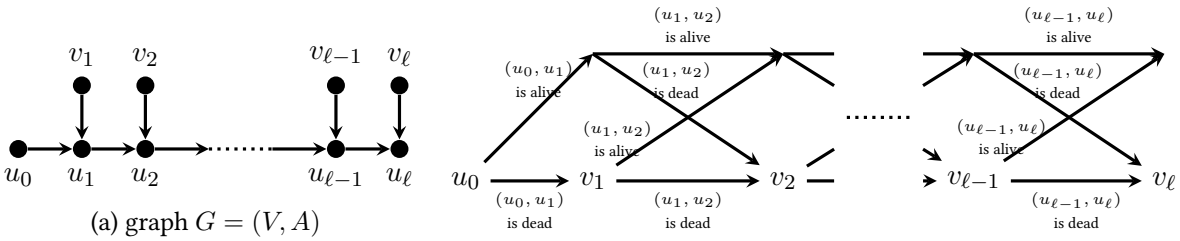
First, we consider the probability $\Pr(v_i \in E(\pi, \Phi))$ for each $i \in [\ell]$. We can see that π selects v_i if and only if the edge (u_{i-1}, u_i) is dead, which yields $\Pr(v_i \in E(\pi, \Phi)) = \epsilon$. We can easily confirm that π finally activates all $u_0, \dots, u_\ell$ for every realization and each v_i is selected with probability ϵ, therefore $\Delta(\pi|\varnothing) = \ell + 1 + \epsilon\ell$. On the other hand, the numerator of the definition of the adaptive submodularity ratio can be calculated as follows. The expected marginal gain of v_i is

$$\Delta(v_i|\varnothing) = 1 + \sum_{j=i}^{\ell} \epsilon(1-\epsilon)^{j-i} = 2 - (1-\epsilon)^{\ell-i+1}.$$

Similarly, we have $\Delta(u_0|\varnothing) = \frac{1}{\epsilon}\{1 - (1-\epsilon)^{\ell+1}\}$. Finally, we can compute the adaptive submodularity ratio as

$$\begin{aligned}
\gamma_{\varnothing,\ell} &\leq \frac{\sum_{v \in V} \Pr(v \in E(\pi, \Phi))\Delta(v|\varnothing)}{\Delta(\pi|\varnothing)} \\
&= \frac{\frac{1}{\epsilon}\{1 - (1-\epsilon)^{\ell+1}\} + \epsilon \sum_{i=1}^{\ell}(2 - (1-\epsilon)^{\ell-i+1})}{\ell + \epsilon\ell + 1} \\
&\leq \frac{\frac{1}{\epsilon} + 2\epsilon\ell}{\ell + \epsilon\ell + 1}
\end{aligned}$$

By setting $\epsilon = 1/\sqrt{\ell}$ and taking $\ell \to \infty$, we can see $\gamma_{\varnothing,\ell} \to 0$. To conclude, the adaptive submodularity ratio can become arbitrarily small if the graph is non-bipartite.



(a) graph $G = (V, A)$

(b) policy π

Figure 3.3: An instance with a non-bipartite graph such that the adaptive submodularity ratio can be arbitrarily small. Since the space is limited, nodes of π that have the same subtree are indicated by a single node.

3.6 Adaptive Feature Selection

In this section, we consider an adaptive variant of feature selection for sparse regression.

Let us consider the following scenario. A learner has all feature vectors in advance, but they are not accurate due to sensing noise. Here each sensor corresponds to a single feature vector. The learner can obtain accurate feature vectors by replacing inaccurate sensors with high-quality sensors, but the number of high-quality sensors is limited to k. The learner selects k features for observing their accurate feature vectors.

We formulate this scenario as the following problem. At the beginning, a learner knows a response vector $\mathbf{b} \in \mathbb{R}^m$ and a prior distribution over the features, but does not know the features themselves. Namely, we regard the inaccurate feature vectors obtained with noisy sensors as prior distributions on accurate feature vectors. A random variable Φ indicates the uncertainty over the observed feature vectors. From the noisy sensors, we can know only a prior distribution of Φ but not the true ϕ. Let $V = [n]$ be the set of features. At each step, the learner can query a feature $v \in V$ and observe its feature vector $\phi(v) \in \mathbb{R}^m$. We assume the noise of sensors are independent of each other; i.e., there exists a distribution $p_v(\phi(v))$ for each $v \in V$ and we can factorize p as $p(\phi) = \prod_{v \in V} p_v(\phi(v))$.

Let $\mathbf{A}(\phi) = (\phi(1) \cdots \phi(n))$ be the realized feature matrix under realization ϕ. The objective function to be maximized is defined as

$$f(S, \phi) = \|\mathbf{b}\|_2^2 - \min_{\mathbf{w} \in \mathbb{R}^S} \|\mathbf{b} - \mathbf{A}(\phi)_S \mathbf{w}\|_2^2.$$

3.6.1 Bound of Adaptive Submodularity Ratio

To bound the adaptive submodularity ratio of adaptive feature selection, we give a general lower bound of the adaptive submodularity ratio by using (non-adaptive) submodularity ratios of all realizations.

Theorem 54. *Let $f \colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ be adaptive monotone with respect to distribution $p(\phi)$. Assume the value of $f(S, \phi)$ depends only on $(\phi(v))_{v \in S}$ not on $(\phi(v))_{v \in V \setminus S}$, i.e., $f(S, \phi) = f(S, \phi')$ for all ϕ and ϕ' such that $\phi(v) = \phi(v)$ for all $v \in S$. We also assume $p(\phi)$ can be factorized to distributions $p_v(\phi(v))$ of states of each $v \in V$, i.e., $p(\phi) = \prod_{v \in V} p_v(\phi(v))$. Let $\gamma^{\phi}_{X,k}$ be the submodularity ratio of $f(\cdot, \phi)$ for each realization ϕ. For any distribution p_v of $\phi(v)$, the adaptive submodularity ratio $\gamma_{\psi,k}$ can be bounded as*

$$\gamma_{\psi,k} \geq \min_{\phi \sim \psi} \gamma^{\phi}_{\mathrm{dom}(\psi),k}.$$

Proof. Let ψ be any partial realization and $\pi \in \Pi_k$ be any policy of height at most k. Fix an arbitrary subset $\psi' \subseteq \psi$. Note that we have $f(\mathrm{dom}(\psi), \phi) = f(\mathrm{dom}(\psi), \phi')$ for any $\phi, \phi' \supseteq \psi$ due to the

assumption that $f(S,\phi)$ depends only on $(\phi(v))_{v\in S}$; considering this, we abuse the notation and define $f(\psi) \coloneqq f(\mathrm{dom}(\psi),\phi)$ for any $\phi \supseteq \psi$. Let ψ_v be the partial realization just before v is selected in π. If there are multiple partial realizations ψ such that $\pi(\psi) = v$, we can duplicate v and consider them to be different elements. Now we can transform the numerator of the adaptive submodularity ratio as

$$
\begin{aligned}
&\sum_{v\in V} \Pr(v \in E(\pi,\Phi)|\Phi\sim\psi')\Delta(v|\psi') \\
&= \sum_{v\in V} \Pr(v \in E(\pi,\Phi)|\Phi\sim\psi')\mathbb{E}\left[f(\psi'\cup\{(v,\Phi(v))\}) - f(\psi')\middle|\Phi\sim\psi'\right] \\
&= \sum_{v\in V} \Pr(v \in E(\pi,\Phi)|\Phi\sim\psi') \sum_{y\in\mathcal{Y}} \Pr(\Phi(v)=y|\Phi\sim\psi')\Big\{f(\psi'\cup\{(v,y)\}) - f(\psi')\Big\} \\
&= \sum_{v\in V} \Pr(v \in E(\pi,\Phi)|\Phi\sim\psi') \sum_{y\in\mathcal{Y}} \Pr(\Phi(v)=y|\Phi\sim\psi'\cup\psi_v)\Big\{f(\psi'\cup\{(v,y)\}) - f(\psi')\Big\} \\
&\qquad\qquad \text{(due to the independence of } \phi(v) \text{ from } (\phi(u))_{u\in\mathrm{dom}(\psi_v)}) \\
&= \sum_{v\in V} \Pr(v \in E(\pi,\Phi)|\Phi\sim\psi')\mathbb{E}\left[f(\psi'\cup\{(v,\Phi(v))\}) - f(\psi')\middle|\Phi\sim\psi'\cup\psi_v\right] \\
&= \sum_{v\in V} \Pr(\Phi\sim\psi'\cup\psi_v|\Phi\sim\psi')\mathbb{E}\left[f(\mathrm{dom}(\psi')\cup\{v\},\Phi\}) - f(\mathrm{dom}(\psi'),\Phi)\middle|\Phi\sim\psi'\cup\psi_v\right] \\
&= \mathbb{E}\left[\sum_{v\in E(\pi,\Phi)}\Big\{f(\mathrm{dom}(\psi')\cup\{v\},\Phi) - f(\mathrm{dom}(\psi'),\Phi)\Big\}\middle|\Phi\sim\psi'\right].
\end{aligned}
$$

From the above equality, we get

$$
\begin{aligned}
&\min_{\phi\sim\psi} \gamma^{\phi}_{\mathrm{dom}(\psi),k}\Delta(\pi|\psi') \\
&= \min_{\phi\sim\psi} \gamma^{\phi}_{\mathrm{dom}(\psi),k}\mathbb{E}\left[f(\mathrm{dom}(\psi')\cup E(\pi,\Phi),\Phi) - f(\mathrm{dom}(\psi'),\Phi)\middle|\Phi\sim\psi'\right] \\
&\le \mathbb{E}\left[\gamma^{\Phi}_{\mathrm{dom}(\psi),k}\Big\{f(\mathrm{dom}(\psi')\cup E(\pi,\Phi),\Phi) - f(\mathrm{dom}(\psi'),\Phi)\Big\}\middle|\Phi\sim\psi'\right] \\
&\le \mathbb{E}\left[\sum_{v\in E(\pi,\Phi)}\Big\{f(\mathrm{dom}(\psi')\cup\{v\},\Phi) - f(\mathrm{dom}(\psi'),\Phi)\Big\}\middle|\Phi\sim\psi'\right] \\
&\qquad\qquad \text{(From the definition of the submodularity ratio)} \\
&= \sum_{v\in V} \Pr(v \in E(\pi,\phi)|\Phi\sim\psi')\Delta(v|\psi').
\end{aligned}
$$

This inequality holds for any ψ and $\pi \in \Pi_k$. To conclude, we obtain $\gamma_{\psi,k} \ge \min_{\phi\sim\psi} \gamma^{\phi}_{\mathrm{dom}(\psi),k}$. □

By using Theorem 54 and Theorem 21 by Das and Kempe [2011], we obtain the following lower bound of the adaptive submodularity ratio.

Corollary 55. *Assume each column of $\mathbf{A}(\phi)$ is normalized. For any $\mathbf{b} \in \mathbb{R}^n$ and any distribution p_v of each $\phi(v)$, the adaptive submodularity ratio $\gamma_{\ell,k}$ can be bounded as*

$$
\gamma_{\ell,k} \ge \min_{\phi} \min_{S\subseteq V:|S|\le k+\ell} \lambda_{\min}(\mathbf{A}(\phi)_S^{\top}\mathbf{A}(\phi)_S),
$$

where $\lambda_{\min}(\cdot)$ represents the smallest eigenvalue.

Proof. From the definition, we can see $f(S, \phi)$ depends only on selected columns $(\phi(v))_{v \in S}$ and not on the other columns $(\phi(v))_{v \in V \setminus S}$.

We can show

$$\begin{aligned} f(S, \phi) &= \|\mathbf{b} - \mathbf{0}\|^2 - \min_{\mathrm{supp}(\mathbf{w}) \subseteq S} \|\mathbf{b} - \mathbf{A}(\phi)\mathbf{w}\|^2 \\ &\leq \|\mathbf{b} - \mathbf{0}\|^2 - \min_{\mathrm{supp}(\mathbf{w}) \subseteq T} \|\mathbf{b} - \mathbf{A}(\phi)\mathbf{w}\|^2 = f(T, \phi) \end{aligned}$$

for all $S \subseteq T$. From this property, called pointwise monotonicity, for any partial realization ψ and $v \in V \setminus \mathrm{dom}(\psi)$, we obtain

$$\begin{aligned} \Delta(v|\psi) &= \mathbb{E}[f(\mathrm{dom}(\psi) \cup \{v\}, \Phi) - f(\mathrm{dom}(\psi), \Phi) | \Phi \sim \psi] \\ &\geq 0, \end{aligned}$$

from which the adaptive monotonicity of f with respect to p follows.

By applying Theorem 54, we obtain

$$\gamma_{\psi,k} \geq \min_{\phi \sim \psi} \gamma^{\phi}_{\mathrm{dom}(\psi),k},$$

where $\gamma^{\phi}_{X,k}$ is the submodularity ratio of $f(\cdot, \phi)$ for realization ϕ. From Theorem 21, we obtain the following lower bound:

$$\gamma^{\phi}_{\mathrm{dom}(\psi),k} \geq \min_{S \subseteq V \colon |S| \leq k + |\psi|} \lambda_{\min}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S).$$

Finally, we have

$$\begin{aligned} \gamma_{\ell,k} &= \min_{\psi \colon |\psi| \leq \ell} \gamma_{\psi,k} \\ &\geq \min_{\psi \colon |\psi| \leq \ell} \min_{\phi} \gamma^{\phi}_{\mathrm{dom}(\psi),k} \\ &\geq \min_{\psi \colon |\psi| \leq \ell} \min_{\phi} \min_{S \subseteq V \colon |S| \leq k + |\psi|} \lambda_{\min}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S) \\ &= \min_{\phi} \min_{S \subseteq V \colon |S| \leq k + \ell} \lambda_{\min}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S). \end{aligned}$$

□

3.6.2 Bound of Adaptivity Gap

We can also obtain a bound on the adaptivity gap of adaptive feature selection as follows.

Proposition 56. *Let* $f(S, \phi) = \|\mathbf{b}\|_2^2 - \min_{\mathbf{w} \in \mathbb{R}^S} \|\mathbf{b} - \mathbf{A}(\phi)_S \mathbf{w}\|_2^2$ *and suppose that* $p(\phi)$ *can be factorized as* $p(\phi) = \prod_{v \in V} p_v(\phi(v))$. *We have*

$$\mathsf{GAP}_k \geq \frac{\min_{\phi} \min_{S \subseteq V \colon |S| \leq k} \lambda_{\min}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)}{\max_{\phi} \max_{S \subseteq V \colon |S| \leq k} \lambda_{\max}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)}.$$

Proof. We can readily confirm that the objective function can be rewritten as follows.

$$f(S, \phi) = (\mathbf{A}(\phi)_S^\top \mathbf{b})^\top (\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)^+ (\mathbf{A}(\phi)_S^\top \mathbf{b}).$$

For any $S \subseteq V$ such that $|S| \leq k$, we have

$$\begin{aligned}
\mathbb{E}[f(S,\Phi)] &= \mathbb{E}\left[(\mathbf{A}(\Phi)_S^\top \mathbf{b})^\top (\mathbf{A}(\Phi)_S^\top \mathbf{A}(\Phi)_S)^+ (\mathbf{A}(\Phi)_S^\top \mathbf{b})\right] \\
&\geq \mathbb{E}\left[\lambda_{\min}((\mathbf{A}(\Phi)_S^\top \mathbf{A}(\Phi)_S)^+) \|\mathbf{A}(\Phi)_S^\top \mathbf{b}\|_2^2\right] \\
&\geq \mathbb{E}\left[\frac{\|\mathbf{A}(\Phi)_S^\top \mathbf{b}\|_2^2}{\max_\phi \max_{T\subseteq V\colon |T|\leq k} \lambda_{\max}(\mathbf{A}(\phi)_T^\top \mathbf{A}(\phi)_T)}\right] \\
&= \frac{\mathbb{E}\left[\|\mathbf{A}(\Phi)_S^\top \mathbf{b}\|_2^2\right]}{\max_\phi \max_{T\subseteq V\colon |T|\leq k} \lambda_{\max}(\mathbf{A}(\phi)_T^\top \mathbf{A}(\phi)_T)} \\
&= \frac{\sum_{v\in S} \mathbb{E}\left[(\mathbf{A}(\Phi)_v^\top \mathbf{b})^2\right]}{\max_\phi \max_{T\subseteq V\colon |T|\leq k} \lambda_{\max}(\mathbf{A}(\phi)_T^\top \mathbf{A}(\phi)_T)} \\
&= \frac{\sum_{v\in S} \mathbb{E}[f(\{v\},\Phi)]}{\max_\phi \max_{T\subseteq V\colon |T|\leq k} \lambda_{\max}(\mathbf{A}(\phi)_T^\top \mathbf{A}(\phi)_T)}.
\end{aligned}$$

From this inequality, we can bound the supermodularity ratio $\beta_{\varnothing,k}$ of $\mathbb{E}_\Phi[f(\cdot,\Phi)]$ as

$$\beta_{\varnothing,k} \geq \frac{1}{\max_\phi \max_{S\subseteq V\colon |S|\leq k} \lambda_{\max}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)}.$$

Plugging it and the inequality of Corollary 55 into Theorem 45, we obtain

$$\mathsf{GAP}_k \geq \frac{\min_\phi \min_{S\subseteq V\colon |S|\leq k} \lambda_{\min}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)}{\max_\phi \max_{S\subseteq V\colon |S|\leq k} \lambda_{\max}(\mathbf{A}(\phi)_S^\top \mathbf{A}(\phi)_S)}.$$

□

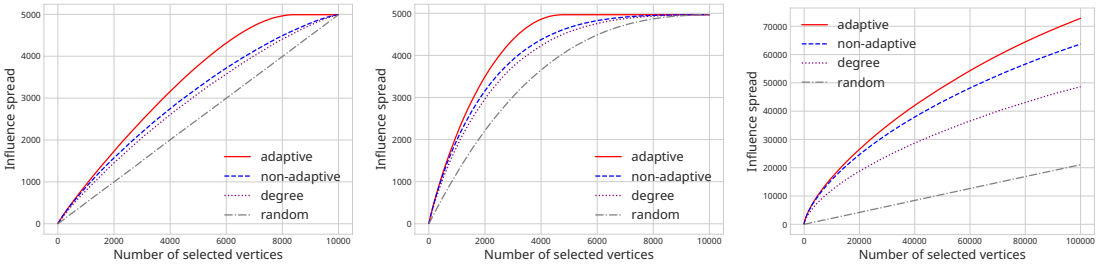
Remark 57. These results on the adaptive submodularity ratio and adaptivity gap can be extended to more general loss functions with restricted strong concavity and restricted smoothness as in Elenberg et al. [2018].

3.7 Experiments

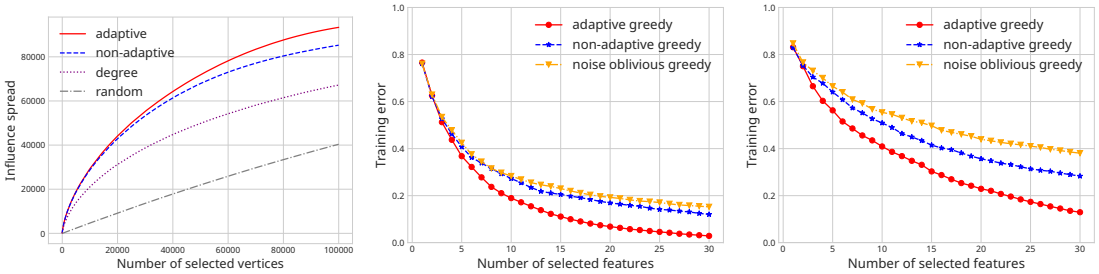
We conduct experiments on two applications: adaptive influence maximization and adaptive feature selection. For each setting, we conduct 20 trials and plot their mean values.

3.7.1 Adaptive Influence Maximization

Datasets. We conduct experiments on two datasets of adaptive influence maximization. The first dataset is a synthetic bipartite graph generated randomly according to Erdös–Renyi rule. We set the number of source and sink vertices to 10000, i.e., $|V| = |U| = 10000$. For each pair $(v,u) \in V \times U$, we add an edge between v and u with probability 0.001. The second dataset is Yahoo! Search Marketing Advertiser–Phrase Bipartite Graph [Yah], which is a bipartite graph representing relationships between advertisers and search phrases; we have $|V| = 459678$, $|U| = 193582$, and $|A| = 2278448$. For both datasets, the weight of each vertex in U is drawn from the uniform distribution on $[0,1]$.



(a) infmax, synth., linear threshold (b) infmax, synth., extended linear thre. (c) infmax, yahoo, linear threshold



(d) infmax, yahoo, extended linear thre. (e) feature, $\sigma = 0.1$ (f) feature, $\sigma = 0.2$

Figure 3.4: Experimental results on adaptive influence maximization (a)–(d) and adaptive feature selection (e)–(f). (a) and (b) are the results on synthetic datasets with the linear threshold model and extended linear threshold model, respectively. (c) and (d) are the results on Yahoo! dataset [Yah] with the linear threshold model and extended linear threshold model, respectively. (e) and (f) are the results on synthetic datasets with uniform noise distribution on $[-\sigma, \sigma]$ with $\sigma = 0.1, 0.2$, respectively.

Diffusion Model. We consider two diffusion models. The first one is the linear threshold model. The probability that each edge $(v, u) \in A$ is alive is set to the reciprocal of the degree of the sink vertex, that is, $1/|\delta_-(v)|$. As the second diffusion model, we consider an extended version of the linear threshold model, which is also a special case of the triggering model. In this model, for each sink vertex v, the subset of incoming live edges is determined as follows. We sample t edges with replacement from $\delta_-(v)$ uniformly at random, and an edge turns alive if it is sampled at least once. In our experiments, parameter t is set to 3.

Benchmarks. We compare the adaptive greedy algorithm with three non-adaptive benchmarks. The first benchmark is the non-adaptive greedy algorithm, called non-adaptive, which is a standard greedy algorithm [Nemhauser et al., 1978] for maximizing the expected value of the objective function $\mathbb{E}_\Phi[f(\cdot, \Phi)]$. The second benchmark is Degree, which selects the set of vertices with the top-k largest degree. The third benchmark is Random, which selects a random subset of size k.

Results. Objective values achieved by the algorithms are shown in Figures 3.4(a) to 3.4(d). In all settings, the adaptive greedy algorithm outperforms all the benchmarks.

3.7.2 Adaptive Feature Selection

Datasets. We use synthetic datasets generated randomly as follows. First, we determine the mean $\mathbb{E}_\Phi[\mathbf{A}(\Phi)] \in \mathbb{R}^{m \times n}$ according to the uniform distribution on $[0, 1]$. After that, each column is normalized so that its mean is 0 and its standard deviation is 1. We obtain $\mathbf{A}(\phi)$ by adding $\epsilon \in \mathbb{R}^{m \times n}$ to $\mathbb{E}_\Phi[\mathbf{A}(\Phi)]$, where each element of ϵ is drawn from the uniform distribution on $[-\sigma, \sigma]$. We consider two settings: $\sigma = 0.1$ and 0.2. We select a random sparse subset S^* of features such that $|S^*| = 30$, and we let $\mathbf{y} = \mathbf{A}(\phi)_{S^*}\mathbf{w}$ be the response vector, where each element of $\mathbf{w} \in \mathbb{R}^S$ is drawn from the standard normal distribution. In all settings, we set $n = 1000$ and $m = 100$.

Benchmarks. We compare the adaptive greedy algorithm with two benchmarks. The first benchmark is the non-adaptive greedy algorithm. Regarding the adaptive and non-adaptive greedy algorithms, it is hard to evaluate the exact values of the objective functions, and so we approximately evaluate them by sampling $\mathbf{A}(\Phi)$ randomly according to posterior distributions. The second benchmark is the noise-oblivious greedy algorithm, a non-adaptive algorithm that greedily selects a subset based on the mean, $\mathbb{E}_\Phi[\mathbf{A}(\Phi)]$.

Results. The results are shown in Figures 3.4(e) and 3.4(f). In both settings, the adaptive greedy algorithm outperforms the two benchmarks.

3.8 Related Work

Comparison with Kusner [2014]. To our knowledge, the first attempt to generalize the submodularity ratio to the adaptive setting is Kusner [2014]. They defined *approximate adaptive submodularity*, a notion that is similar to ours, as follows.

$$\gamma = \min_{S \subseteq V, \psi} \frac{\sum_{v \in S} \Delta(v|\psi)}{\Delta(S|\psi)}.$$

The key difference is that they did not replace subset S with policy π. In Section 3.8.1, we show that the approximate adaptive submodularity is not sufficient for providing an approximation guarantee of the adaptive greedy algorithm.

Comparison with Yong et al. [2017]. Another attempt to relax adaptive submodularity is presented in Yong et al. [2017]. They introduced ζ-*weakly adaptive submodular functions* as follows.

Definition 58 (ζ-weak adaptive submodularity)**.** Let $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ be a set function and p be a distribution of ϕ. For any $\zeta \geq 1$, we say f is adaptive submodular with respect to p if for any partial realization $\psi \subseteq \psi'$ and any element $v \in V \setminus \mathrm{dom}(\psi')$, it holds

$$\zeta \Delta(v|\psi) \geq \Delta(v|\psi').$$

Let ζ^* be the infimum of ζ satisfying the above inequality.

Analogous to our adaptive submodularity ratio, one can readily see that 1-weak adaptive submodularity is equivalent to the adaptive submodularity. In general, however, there is a difference between the two notions; the adaptive submodularity ratio can be bounded from below by $1/\zeta^*$, implying that it is more demanding to bound the value of ζ^* than that of the adaptive submodularity ratio.

Proposition 59. *For any set function $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ and distribution p, we have*

$$\frac{1}{\zeta^*} \leq \min_{k \in \mathbb{Z}_{\geq 0}, \psi} \gamma_{\psi,k}.$$

We provide a proof in Section 3.8.2. Yong et al. [2017] studied a problem called *group-based active diagnosis* and gave a bound of ζ, but some vital assumptions seem to have been missed. In Section 3.8.2, we provide a problem instance in which their bound does not hold. We also present instances of adaptive influence maximization and adaptive feature selection for which our framework provides strictly better approximation ratios than those obtained with the weak adaptive submodularity in Section 3.8.2.

Adaptive Submodularity. Adaptive submodularity was proposed by Golovin and Krause [2011a]. There are several attempts to adaptively maximize set functions that do not satisfy adaptive submodularity (e.g., Kusner [2014], Yong et al. [2017]). Chen et al. [2015] analyzed the greedy policy focusing on the maximization of mutual information, which does not have adaptive submodularity.

Submodularity Ratio. Submodularity ratio was proposed by Das and Kempe [2011] for sparse regression with square loss. Recently, Elenberg et al. [2018] extended this result to more general loss functions with restricted strong convexity and restricted smoothness. Bogunovic et al. [2018] proposed the notion of *supermodularity ratio*. Bian et al. [2017] provided a guarantee of the non-adaptive greedy algorithm for the case where the total curvature and the submodularity ratio of objective functions are bounded.

Influence Maximization. Influence maximization was proposed by Kempe et al. [2003]. An adaptive version of influence maximization was first considered by Golovin and Krause [2011a]. They showed that this objective function satisfies adaptive submodularity under the independent cascade model in general graphs. Influence maximization on a bipartite graph has been studied for applications to advertisement selection [Alon et al., 2012, Soma et al., 2014]. This problem setting was extended to the adaptive setting by Hatano et al. [2016], but only the independent cascade model was considered. The curvature of its objective function was studied by Maehara et al. [2017].

Feature Selection. Kale et al. [2017] considered the problem called adaptive feature selection, but their problem setting is different from ours. In their setting, the learner solves feature selection problems multiple times. They studied the adaptivity among the multiple rounds, while we studied the adaptivity inside of a single round.

3.8.1 Counterexample to the Statement of Kusner [2014]

Kusner [2014] has defined *approximate adaptive submodularity* as follows.

Definition 60 ([Kusner, 2014, Definition 2]). A set function $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ and a distribution p on $\mathcal{Y}^V$ is *approximately adaptive submodular* if for any subrealization ψ such that $p(\psi) > 0$ and any $S \subseteq V \setminus \mathrm{range}(\psi)$, we have

$$\sum_{v \in S} \Delta(v|\psi) \geq \gamma \Delta(S|\psi),$$

where $\gamma \in [0, 1]$ represents the submodularity ratio of the non-adaptive function.

Below we present a counterexample to the statement of Kusner [2014], which says that a bounded γ yields a bounded approximation ratio of the adaptive greedy algorithm.

Let $\mathcal{Y} = \{0, 1, \ldots, M-1\}$ be the set of all possible states and $V = \{u\} \cup \{z_i \mid i \in [k]\} \cup \{v_i^y \mid i \in [k-1],\ y \in \mathcal{Y}\}$ be the ground set. We define $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}$ as follows.

$$f(S, \phi) = |S \cap \{u\}| + (1+\epsilon)|S \cap \{z_1, \ldots, z_k\}| + M \sum_{y \in \mathcal{Y}, i \in [k-1]} \mathbf{1}_{\{\phi(v_i^y) = 1 \text{ and } v_i^y \in S\}},$$

where $\epsilon > 0$ is any small constant. Note that this function is normalized and adaptive monotone. For each $y \in \mathcal{Y}$, we define ϕ_y as $\phi_y(u) = y$, $\phi_y(z_i) = 0$ for each $i \in [k]$, $\phi_y(v_i^y) = 1$ for each $i \in [k-1]$, and $\phi_y(v_i^{y'}) = 0$ for each $y' \in \mathcal{Y} \setminus \{y\}$ and $i \in [k-1]$. Let p be a distribution defined as

$$p(\phi) = \begin{cases} \frac{1}{|\mathcal{Y}|} & \text{if } \phi = \phi_y \text{ for some } y \in \mathcal{Y} \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to see that f is approximately adaptive submodular with $\gamma = 1$ with respect to p because $\Delta(\cdot|\psi)$ is a linear function for any subrealization ψ. Note that f is not adaptive submodular with respect to p because $\Delta(v_1^1|\varnothing) = 1 < M = \Delta(v_1^1|\{(u, 1)\})$.

Kusner [2014] stated that the adaptive greedy algorithm achieves $(1 - \mathrm{e}^{-\gamma})$-approximation for any normalized, adaptive monotone, and approximately adaptive submodular function. However, the adaptive greedy algorithm achieves only $(1+\epsilon)/M$-approximation for the above f and p as is explained below. The adaptive greedy algorithm selects $z_1, \ldots, z_k$ since their expected marginal gain is $1 + \epsilon$ and the expected marginal gain of other elements is 1. On the other hand, the optimal policy first selects u and proceeds to select $\{v_1^{\phi(u)}, \ldots, v_{k-1}^{\phi(u)}\}$ according to the observed $\phi(u)$. The adaptive greedy policy achieves $k(1+\epsilon)$ and the optimal policy achieves $1 + (k-1)M$. Thus the approximation ratio gets close to $(1+\epsilon)/M$ as k increases, and it can be arbitrarily small since the number of possible states, M, is not bounded. Namely, even if γ is bounded by a constant, the approximation guarantee of the adaptive greedy algorithm can become arbitrarily bad in general, which contradicts the statement of [Kusner, 2014].

3.8.2 About Comparison with Yong et al. [2017]

Proof for Comparison with Yong et al. [2017]

Proof of Proposition 59. From the definition of ζ^*, we have $\zeta^* \Delta(v|\psi) \geq \Delta(v|\psi')$ for any $\psi \subseteq \psi'$ and $v \in V \setminus \mathrm{dom}(\psi')$. It is enough to show $\frac{1}{\zeta^*}\Delta(\pi|\psi) \leq \sum_{v \in V} \Pr(v \in E(\pi, \phi))\Delta(v|\psi)$ for arbitrary $\psi \subseteq \psi'$ and π. Let ψ_v be the partial realization just before v is selected in π. If there are multiple partial realizations ψ such that $\pi(\psi) = v$, we can duplicate v and take them to be different elements. Then

we can write $\Delta(\pi|\psi) = \sum_{v\in V} \Pr(v \in E(\pi,\phi))\Delta(v|\psi_v)$. By applying the bound of weak adaptive submodularity, we have

$$\begin{aligned}\Delta(\pi|\psi) &= \sum_{v\in V} \Pr(v \in E(\pi,\phi))\Delta(v|\psi \cup \psi_v) \\ &\leq \zeta^* \sum_{v\in V} \Pr(v \in E(\pi,\phi))\Delta(v|\psi),\end{aligned}$$

which implies the statement. □

From this proposition, we can see that Theorem 43 is stronger than the result of Yong et al. [2017] as follows. They showed that the adaptive greedy algorithm is guaranteed to achieve $(1-\exp(-\ell/(\zeta^* k)))$-approximation in Yong et al. [2017, Theorem 1]. From Proposition 59, we always have $(1-\exp(-\ell/(\zeta^* k))) \leq (1-\exp(-\gamma_{\psi,k}\ell/k))$.

Counterexample to the Proposition of Yong et al. [2017]

In this subsection we provide an instance of group-based active diagnosis in which the weak adaptive submodularity cannot give a bound of the approximation ratio of the adaptive greedy algorithm.

The formal problem statement of group-based active diagnosis can be described as follows. We have set V of tests and set $\mathcal{Y}$ of their possible outcomes. There are two random variables that uniquely specify the outcome of each test: the state x and the mode q. Let $\mathcal{X}$ be the set of all possible states and $\mathcal{Q}$ the set of all possible modes. We know the prior joint distribution $p(x,q)$ of x and q, but does not know their true values. Let $\mu(v,x,q) \in \mathcal{Y}$ be the unique outcome of test v when the true state is x and the true mode is q. We aim to determine x by sequentially conducting several tests out of V.

Yong et al. [2017] formulated this problem as the problem of maximizing the following objective function:

$$f(S,(x,q)) = 1 - \sum_{x'\in\mathcal{X}:\ \exists q'\in\mathcal{Q},\ \forall v\in S,\ \mu(v,x',q')=\mu(v,x,q')} \ \sum_{q''\in\mathcal{Q}} p(x',q''),$$

where the first summation is about all possible $x' \in \mathcal{X}$ under the outcomes of tests S made so far. Proposition 2 of Yong et al. [2017] claims that this objective function is ζ-weakly adaptive submodular for

$$\zeta \leq \frac{|\mathcal{Q}|}{\min_{x\in\mathcal{X},q\in\mathcal{Q}} p(x,q)}.$$

However, it does not hold in the following example.

Example 61. Let $\mathcal{X} = \{x_1, x_2\}$ be the set of states and $\mathcal{Q} = \{q_1, q_2, q_3\}$ the set of modes. For each $x \in \mathcal{X}$ and $q \in \mathcal{Q}$, we assume $p(x,q) = \frac{1}{6}$. We consider two actions v_1 and v_2, which yield the unique outcome out of $\mathcal{Y} = \{+1, -1\}$ indicated in Table 3.2 for each state $x \in \mathcal{X}$ and mode $q \in \mathcal{Q}$.

We first consider the expected marginal gain obtained by performing action v_2 at the beginning. In this situation, performing v_2 yields outcome $+1$ or -1 with probability $1/2$. If the outcome is $+1$, we can reject neither x_1 nor x_2. This is the case for outcome -1. Thus we have $\Delta(v_2|\varnothing) = 0$.

Next, we assume the algorithm performs v_1 at the beginning and obtains the outcome of -1, i.e., $\psi = \{(v_1, -1)\}$. Now the possible pairs of the state and the mode are only (x_1, q_3) and (x_2, q_3). By performing action v_2, we obtain the outcome $+1$ or -1 with probability $\frac{1}{2}$ and reject x_2 or x_1, respectively. Thus the expected marginal gain is $\Delta(v_2|\psi) = \frac{1}{2}\sum_{q\in\mathcal{Q}} p(x_2,q) + \frac{1}{2}\sum_{q\in\mathcal{Q}} p(x_1,q) = \frac{1}{2}$.

From the definition of ζ, we must have $\Delta(v_2|\psi) \leq \zeta\Delta(v_2|\varnothing)$, but no finite ζ satisfies this inequality. This contradicts Proposition 2 of Yong et al. [2017], which claims ζ is finite.

Table 3.2: Outcome

(x, q)	$\mu(v_1, x, q)$	$\mu(v_2, x, q)$
(x_1, q_1)	$+1$	$+1$
(x_1, q_2)	$+1$	-1
(x_1, q_3)	-1	$+1$
(x_2, q_1)	$+1$	$+1$
(x_2, q_2)	$+1$	-1
(x_2, q_3)	-1	-1

Comparison in Adaptive Influence Maximization

We provide an instance of adaptive influence maximization such that the adaptive submodularity ratio yields an approximation ratio significantly better than that obtained with the weak adaptive submodularity [Yong et al., 2017].

Example 62. We use the same problem instance as Example 49. At the beginning, the expected marginal gain of v_k is $\Delta(v_k|\varnothing) = 1/k$. Let ψ be the observations obtained when $v_1, \dots, v_{k-1}$ are selected and all edges are turned out to be dead. In this case, since the edge (v_k, u) must be alive, the expected marginal gain is $\Delta(v_k|\psi) = 1$. The weak adaptive submodularity constant is bounded as $\zeta \geq \Delta(v_k|\psi)/\Delta(v_k|\varnothing) = k$. This implies that the weak adaptive submodularity constant cannot yield a lower bound of the approximation ratio better than $1 - \exp(-\frac{1}{k}) = \mathrm{O}(\frac{1}{k})$, while the adaptive submodularity ratio provides a lower bound $1 - \exp(-(k+1)/2k) = \Omega(1)$.

Comparison in Adaptive Feature Selection

Regarding adaptive feature selection, we describe an advantage of the adaptive submodularity ratio in comparison with the weak adaptive submodularity [Yong et al., 2017]. As detailed below, there exists an instance with the following condition: the approximation ratio obtained with the adaptive submodularity ratio is bounded, while that obtained with the weak adaptive submodularity is 0.

Example 63. We can make such an instance even if ϕ is deterministic. Let $\mathbf{A}(\phi) = (\phi(1) \cdots \phi(n))$ be the realized feature matrix under realization ϕ. The objective function is defined as

$$f(S, \phi) = \|\mathbf{b}\|_2^2 - \min_{\mathbf{w} \in \mathbb{R}^S} \|\mathbf{b} - \mathbf{A}(\phi)_S \mathbf{w}\|_2^2.$$

We here let

$$\mathbf{A}(\phi) = \begin{bmatrix} 1 & 1/\sqrt{2} & 0 & \cdots & 0 \\ 0 & 1/\sqrt{2} & 0 & \cdots & 0 \\ 0 & 0 & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 0 \\ a \\ a \\ \vdots \\ a \end{bmatrix},$$

where $a > 0$ is an any positive real value. Let $S = \{3, \dots, n\}$ and $T = \{2, \dots, n\}$, which satisfy $S \subseteq T$. Then, we have

$$\min_{\mathbf{w} \in \mathbb{R}^S} \|\mathbf{b} - \mathbf{A}(\phi)_S \mathbf{w}\|_2^2 = \min_{\mathbf{w} \in \mathbb{R}^{S \cup \{1\}}} \|\mathbf{b} - \mathbf{A}(\phi)_{S \cup \{1\}} \mathbf{w}\|_2^2 = a^2$$

and

$$\min_{\mathbf{w}\in\mathbb{R}^T} \|\mathbf{b} - \mathbf{A}(\phi)_S \mathbf{w}\|_2^2 = \frac{a^2}{2} > \min_{\mathbf{w}\in\mathbb{R}^{T\cup\{1\}}} \|\mathbf{b} - \mathbf{A}(\phi)_{T\cup\{1\}} \mathbf{w}\|_2^2 = 0.$$

Therefore, we obtain

$$f(S\cup\{1\},\phi) - f(S,\phi) = a^2 - a^2 = 0 \qquad \text{and} \qquad f(T\cup\{1\},\phi) - f(T,\phi) = \frac{a^2}{2} - 0 = \frac{a^2}{2},$$

which implies that ζ cannot be bounded from above in general. On the other hand, the largest and smallest eigenvalues of the Hessian, $\mathbf{A}(\phi)^\top \mathbf{A}(\phi)$, are $1 + 1/\sqrt{2}$ and $1 - 1/\sqrt{2}$, respectively. Therefore, the condition number is bounded from above by $3 + 2\sqrt{2}$, which means the adaptive submodularity ratio is bounded from below by $1/(3 + 2\sqrt{2})$.

3.9 Summary and Future Work

In this chapter, we have proposed the framework of the adaptive submodularity ratio. First we formally defined the adaptive submodularity ratio by extending the submodularity ratio to the adaptive setting. We showed that if the adaptive submodularity ratio is bounded, the approximation ratio of the adaptive greedy algorithm is bounded. We also showed that the adaptivity gap can be bounded by the product of the adaptive submodularity ratio and the supermodularity ratio. We provided two applications where the adaptive submodularity ratio is bounded. One is adaptive influence maximization on bipartite graphs in the triggering model and the other is adaptive feature selection. We experimentally illustrated that the adaptive greedy algorithm works well compared to non-adaptive algorithms.

An interesting direction for future work is to apply the proposed framework to active learning with unknown noise. The framework of adaptive submodularity has been applied to several settings of active learning, but its applications are limited to the cases where noise does not exist or the noise distribution is known in advance, such as a noiseless Bayesian active learning problem [Golovin and Krause, 2011a] or the equivalence class determination problem [Golovin et al., 2010]. It would be possible to apply the framework of the adaptive submodularity ratio to active learning with unknown noise.

4 Batch-mode Adaptive Optimization with Structured Queries

In this section, we handle the batch-mode setting of adaptive optimization. This chapter is organized as follows. Section 4.1 introduces the background and overview of this chapter. Section 4.2 formulates the problem setting of batch-mode adaptive optimization, which we tackle in this chapter. In Section 4.3, we introduce applications that can be dealt with under our framework. In Section 4.4, we define an important property called set-adaptive submodularity. Section 4.5 proposes greedy algorithms for the batch-mode setting. In Section 4.6, we consider the case where the set-adaptive submodularity does not hold. In Section 4.7, we describe extensions of the proposed framework for outer matroid constraints, the online setting, and the query-varying setting. In Section 4.8, we conduct experiments on batch-mode adaptive optimization. Section 4.9 reviews related work of this chapter. Section 4.10 provides a summary and future work of this chapter.

4.1 Background and Overview

As described in Section 2.3 and Chapter 3, various sequential decision-making problems can be handled as an adaptive optimization problem. However, the ordinary setting of adaptive optimization, in which a decision-maker alternately repeats the selection of the next element and the observation of its state, takes much time or cost due to its sequential manner. In such a situation, the batch-mode setting, in which the decision-maker selects multiple elements simultaneously, is more realistic.

For example, active learning [Settles, 2012] is a problem setting where we must choose the data points to be labeled while gathering labels of the data points that are unlabeled in the beginning. In the ordinary setting of active learning, we must alternately select the next element to be labeled and observe its label. To achieve high accuracy with a small number of labels, it is important to select the element adaptively according to the already obtained labels. In realistic scenarios of active learning, we are often required to parallelize the label gathering process. For example, it is common to delegate the labeling process to workers in a crowdsourcing platform. In such a scenario, it is not natural to ask each worker to label just one data point, therefore we are required to ask each worker to label multiple data points at the same time. Consequently, the process of active learning becomes the repetition of selecting a set of unlabeled data points given to a single worker and obtaining their labels simultaneously. In contrast to the original setting, we cannot observe the label of selected unlabeled data point immediately and must wait until selecting the set for a single worker. The set of unlabeled data points for each single worker is called *batch* or *query*. This setting of active learning is called *batch-mode active learning* and has been studied from both perspectives of theory and practice [Hoi et al., 2006].

Parallelizing the information gathering process is vital not only for active learning, but also for other sequential decision-making problems in machine learning. Chen and Krause [2013] developed a general framework of adaptive optimization for dealing with parallel information gathering problems including batch-mode active learning. Chen and Krause [2013] argued that if adaptive submodularity holds, then an algorithm that selects a batch at each round greedily, which we call *batch-mode greedy algorithm*, is guaranteed to be competitive with an optimal batch-mode policy. In this chapter, we extend their framework to more general settings including the setting where adaptive submodularity does not hold.

First, we show that the main result by Chen and Krause [2013] actually requires a stronger condition than adaptive submodularity, which we name *set-adaptive submodularity*, and this condition holds in many important applications. We generalize the framework by Chen and Krause [2013] to the setting with structured queries, in which the set of feasible queries is determined by a combinatorial constraint beyond cardinality constraints. We extend the framework to the setting where adaptive submodularity does not hold by using the adaptive submodularity ratio and the supermodularity ratio. In addition, we consider the online setting, in which the different ground set is given at each round, and the query-varying setting, in which the set of feasible queries change at each round. Finally, we conduct numerical experiments, and show our proposed methods outperform benchmarks.

In summary, our contributions in this chapter are as follows.

- We define set-adaptive submodularity, which is a stronger condition than adaptive submodularity, and this condition is satisfied by several applications.
- For providing a guarantee for the batch-mode greedy algorithm, we show that adaptive submodularity is not sufficient and set-adaptive submodularity is sufficient.
- We extend this result to the batch-mode adaptive optimization with structured queries.
- We provide guarantees on greedy-like algorithms for the online setting and query-varying setting.
- We provide guarantees on the batch-mode greedy algorithm in the setting where set-adaptive submodularity does not hold in terms of the adaptive submodularity ratio.
- We empirically confirm that the proposed algorithms outperform benchmarks.

4.2 Batch-mode Adaptive Optimization

In this section, we provide a formal statement of *batch-mode adaptive optimization*, which we tackle in this chapter. This problem setting is an extension of the ordinary adaptive optimization, and we follow the basic notations of adaptive optimization given in Section 2.3. For clarity, we call the setting of the ordinary adaptive optimization the *fully adaptive setting*.

Let V be the ground set and $\phi\colon V \to \mathcal{Y}$ the map that associates each element to its state, where $\mathcal{Y}$ is the set of all possible states. As explained in Section 2.3, in the fully adaptive setting, the decision-maker selects an element $v \in V$ at each round. In contrast, in batch-mode adaptive optimization, the decision-maker is given the set family $\mathcal{J} \subseteq 2^V$ that represents the set of feasible batches, and must select a batch $S \in \mathcal{J}$ at each round. While the decision-maker in the fully adaptive setting can observe the state of the selected element just after selecting it, the decision-maker in the batch-mode setting can observe the states of elements in S all at once. We consider a scenario in which the decision-maker repeats selecting a batch $S \in \mathcal{J}$ and observing their states $(\phi(v))_{v\in S}$ for k rounds.

For handling the batch-mode setting, we need to extend the definition of policies. In the original definition, a policy π is defined as a map from observations so far to an element to be selected next. A policy for the batch-mode setting can be defined as a map from observations so far to a batch $S \in \mathcal{J}$ to be selected next, not a single element. In the same way as the fully adaptive setting, we can denote by $E(\pi^b, \phi)$ the subset of V selected by batch-mode policy π^b under realization ϕ. The goal of this problem is to find a batch-mode policy $\pi^b \in \Pi^b_k$ that maximizes the objective value $\mathbb{E}_\Phi[f(E(\pi^b, \Phi), \Phi)]$. We consider the approximation ratio of batch-mode algorithms in comparison to an optimal batch-mode policy, not an optimal policy in the fully adaptive setting.

Chen and Krause [2013] considered a special case of our setting where the set of feasible batches consists of all subsets of V with size k, i.e., $\mathcal{J} = \{S \subseteq V \mid |S| \leq k\}$ for some $k \in \mathbb{Z}_{>0}$. We extend their problem setting to more general constraints on batches, such as the case where $\mathcal{J}$ is an independent set family of some matroid or $\mathcal{J}$ is the set of batches that satisfy a knapsack constraint.

As stated by Chen and Krause [2013], the batch-mode setting can be viewed as an intermediate setting between the fully adaptive setting and the non-adaptive setting. We can regard the fully adaptive setting as a special case of the batch-mode setting by considering the case where the set of feasible batches consists of all singletons, i.e., $\mathcal{J} = \{\{v\} \mid v \in V\}$. On the other hand, the non-adaptive setting can be identified with the batch-mode setting with only one round.

4.3 Applications

In this section, we describe applications of batch-mode adaptive optimization. The batch-mode versions of existing applications of adaptive optimization can be formulated as a batch-mode adaptive optimization problem.

4.3.1 Batch-mode Active Learning

Active learning is the problem of selecting a set of unlabeled data points to be labeled among the given set V of unlabeled data points. In active learning, the decision-maker is a learner who wants to construct a good training set that consists of a small number of labeled data points. The learner can utilize the labeling oracle who gives the labels of the selected unlabeled data points. The goal of active learning is to achieve high accuracy for predicting the labels of unknown data points with as few queries to the labeling oracle as possible.

Formally, active learning can be formulated as the problem of finding the true hypothesis h^* out of possible hypotheses $\mathcal{H}$. Each hypothesis $h \in \mathcal{H}$ represents some realization ϕ. Here we adopt the Bayesian setting where h^* is generated from a prior distribution p_h. By abuse of notation, we define $p_h(\mathcal{H}') = \sum_{h \in \mathcal{H}'} p_h(h)$ for any $\mathcal{H}' \subseteq \mathcal{H}$. The version space under partial realization ψ is defined to be $\mathcal{H}(\psi) = \{h \in \mathcal{H} \mid \forall (v, y) \in \psi,\ h(v) = y\}$.

There are several objective functions known to satisfy adaptive submodularity. The simplest one is generalized binary search in the Bayesian setting [Dasgupta, 2004], which is the adaptive greedy algorithm for the objective function $f_{\mathrm{GBS}}(S, \phi) = 1 - p_h(\mathcal{H}(\phi|_S))$, where $\phi|_S = \{(v, \phi(v)) \mid v \in S\}$ is the partial realization obtained by selecting S under realization ϕ.

If $\mathcal{H}$ is partitioned into $\mathcal{H}_1, \cdots, \mathcal{H}_m$ and the goal is to determine which partition contains the true hypothesis, EC^2 algorithm [Golovin et al., 2010] is the adaptive greedy algorithm for the objective function

$$f_{\mathrm{EC}^2}(S, \phi) = 1 - \sum_{i<j} \Big(p_h(\mathcal{H}_i(\phi|_S)) p_h(\mathcal{H}_j(\phi|_S)) \Big).$$

Gonen et al. [2013] considered the binary classification of a linearly separable dataset. By supposing the parameter $\mathbf{w} \in \mathbb{R}^d$ is uniformly distributed in the d-dimensional unit ball, ALuMA algorithm is defined to be the adaptive greedy algorithm for the objective function

$$f_{\mathrm{ALuMA}}(S, \phi) = 1 - \Pr\left[\left\{ \mathbf{w} \in \mathbb{B}_1^d \middle| \forall \mathbf{x} \in S,\ \phi(\mathbf{x}) = \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle) \right\} \right].$$

In the fully adaptive setting of active learning, the learner selects an unlabeled data point $v \in V$ and obtains its label $\phi(v) \in \mathcal{Y}$ at each round. In batch-mode active learning, the learner alternately repeats selecting a set $S \in \mathcal{J}$ of unlabeled data points and obtaining their labels $(\phi(v))_{v \in S}$ from the labeling oracle.

For real applications such as the case where we delegate the labeling oracle to workers in a crowdsourcing platform, the batch-mode setting is more efficient. If the expected working time $c_v \in \mathbb{R}_{\geq 0}$ for labeling is assigned to each data point $v \in V$, it is natural to construct each batch under a constraint on the total expected working time. This constraint boils down to a knapsack constraint, i.e.,

$\mathcal{J} = \{S \subseteq V \mid \sum_{v \in S} c_v \leq C\}$, where C is an upper bound on the total expected working time for each batch.

4.3.2 Batch-mode Influence Maximization

Influence maximization [Kempe et al., 2003] is the problem of selecting the set of nodes, called *seed nodes*, and spreading information through a social network from these nodes. This problem models a viral marketing scenario where a company wants to spread the information about a product through the social network by presenting free promotional samples to influential people. Each person who have obtained information about the product might tell the information about the product to their friends. The goal of influence maximization is to spread information to as many people as possible by utilizing this diffusion process. Here we focus on adaptive influence maximization, in which we can observe who are influenced by the selected seed node after selecting it.

Let $G = (V, E)$ be the graph representing the social network. We define $\mathbf{X} \in \{+1, -1\}^E$ to be a vector that represents the state of each edge, $+1$ for *alive* and -1 for *dead*. We assume that this vector $\mathbf{X}$ is determined according to a probability distribution specified by the *diffusion model* such as the independent cascade model, the linear threshold model, or the triggering model [Kempe et al., 2003]. Here we consider the *full-adaption feedback model* [Golovin and Krause, 2011a], in which the realization $\phi \colon V \to E \times \{+1, -1\}$ is defined as

$$\phi(v) = \{((s, t), \mathbf{X}_{st}) \mid (s, t) \in E \text{ such that } s \in R(v)\},$$

where $R(v)$ is the set of all nodes reachable from v only through live edges. The objective function to maximize is the number of influenced nodes, that is, $f(X, \phi) = |\cup_{v \in X} R(v)|$.

In the fully adaptive setting of adaptive influence maximization, the decision-maker alternately repeats selecting a node $v \in V$ and observing the spread $\phi(v)$. On the other hand, in batch-mode adaptive influence maximization, the decision-maker alternately repeats selecting a batch $S \in \mathcal{J}$ and observing the spread $\bigcup_{v \in S} \phi(v)$ all at once. In viral marketing scenarios, distributing free promotional samples to multiple people in parallel is more realistic than distributing them one by one. If we need to pay money for asking each influential people to spread information, we should make a advertising plan under a budget constraint at each round. This setting is naturally reduced to a knapsack constraint.

4.3.3 Batch-mode Adaptive Feature Selection

Adaptive feature selection for sparse regression is the problem of adaptively selecting a set of features that leads to a robust and interpretable model. As described in Section 3.6, we consider a scenario where each feature corresponds to a site for placing a sensor. A learner knows only a prior distribution of all sites and wants to place a limited number of high-quality sensors to important sites adaptively. Formally, the learner knows a response vector $\mathbf{b} \in \mathbb{R}^m$ and a prior distribution of the design matrix $\mathbf{A}(\phi) \in \mathbb{R}^{m \times n}$ in advance, where $V = [n]$. As shown in Section 3.6, the adaptive submodularity ratio of the objective function $f(S, \phi) = \|\mathbf{b}\|_2^2 - \min_{\mathbf{w} \in \mathbb{R}^S} \|\mathbf{b} - \mathbf{A}(\phi)_S \mathbf{w}\|_2^2$ can be bounded by a spectral parameter of the distribution of $\mathbf{A}$. In the fully adaptive setting of adaptive feature selection, the learner alternately selects a feature $v \in V$ and observes the accurate value $\phi(v) = \mathbf{A}(\phi)_v$ of the selected feature. In the batch-mode setting, at each round, the learner selects a batch $S \in \mathcal{J}$ and observes the accurate feature vectors $(\phi(v))_{v \in S} = \mathbf{A}(\phi)_S$ after selecting the batch.

4.4 Set-Adaptive Submodularity

For dealing with the batch-mode setting, Chen and Krause [2013] assumed adaptive submodularity, pointwise submodularity, and pointwise monotonicity of the objective function. Here we show that

these conditions are not sufficient to apply their analyses and there exists a counterexample to their key theorem. To address this issue, we propose a new concept called *set-adaptive submodularity*, which is a stronger condition than adaptive submodularity, but satisfied by objective functions of many applications.

Definition 64 (Set-adaptive submodularity)**.** We say f is set-adaptive submodular with respect to p if for any partial realization $\psi \subseteq \psi'$ and any subset $S \subseteq V \setminus \mathrm{dom}(\psi')$, it holds that

$$\Delta(S|\psi) \geq \Delta(S|\psi').$$

Since S can be regarded as a non-adaptive policy that selects subset S regardless of the observations, it can be easily seen that the following *policy-adaptive submodularity* is a stronger condition than set-adaptive submodularity.

Definition 65 (Policy-adaptive submodularity [Fujii and Kashima, 2016])**.** We say f is policy-adaptive submodular with respect to p if for any partial realization $\psi \subseteq \psi'$ and any policy π such that $\mathrm{range}(\pi) \subseteq V \setminus \mathrm{dom}(\psi')$, it holds that

$$\Delta(\pi|\psi) \geq \Delta(\pi|\psi').$$

Set-adaptive submodularity is strictly stronger than adaptive submodularity. In addition, the combination of strongly adaptive submodularity, pointwise submodularity, and pointwise monotonicity does not imply set-adaptive submodularity. Here we provide an example that satisfies strong adaptive submodularity, pointwise submodularity, and pointwise monotonicity, but does not satisfy set-adaptive submodularity.

Example 66. Let $V = \{a, b, c\}$ be the ground set and $\mathcal{Y} = \{+, -\}$ be the set of possible states. We define the distribution of realization ϕ as $p(\phi) = 0.5$ if $\phi(a) = \phi(b) = \phi(c)$ and $p(\phi) = 0$ otherwise. The objective function $f \colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ is defined as

$$f(S, \phi) = \begin{cases} 2 & \text{if } \phi(a) = \phi(b) = \phi(c) = + \text{ and } \{b, c\} \subseteq S \\ 0 & \text{if } \{b, c\} \cap S = \varnothing \\ 1 & \text{otherwise.} \end{cases}$$

First, we show that this function is adaptive submodular. Since adding a does not affect the objective value, the expected marginal gain of a is always 0 for any observations obtained so far. Hence it is sufficient to consider the expected marginal gain of b due to the symmetry of b and c. We can calculate the expected marginal gain of b as follows.

$$\begin{aligned} \Delta(b|\varnothing) &= 1 \\ \Delta(b|\{(a, +)\}) &= 1 \\ \Delta(b|\{(a, -)\}) &= 1 \\ \Delta(b|\{(c, +)\}) &= 1 \\ \Delta(b|\{(c, -)\}) &= 0 \\ \Delta(b|\{(a, +), (c, +)\}) &= 1 \\ \Delta(b|\{(a, -), (c, -)\}) &= 0, \end{aligned}$$

then we can see adaptive submodularity holds.

Similarly, we can show its strong adaptive submodularity. The extended expected marginal gain of a is 0 for any observations obtained so far, then it holds that $\Delta(a|\psi; \psi') \geq \Delta(a|\psi')$ for all $\psi \subseteq \psi'$. In the case

where $|\psi| = 0$, since the marginal gain of adding b to $\varnothing$ is always 1, it holds that $\Delta(b|\psi;\psi') = \Delta(b|\psi)$. In the case where $|\psi| = 1$, since the posterior distributions after observing ψ and ψ' are the same, $\Delta(b|\psi;\psi') = \Delta(b|\psi)$ holds. Therefore, we have $\Delta(b|\psi;\psi') = \Delta(b|\psi) \geq \Delta(b|\psi')$ from adaptive submodularity in these cases. In the other cases, $\Delta(b|\psi;\psi') \geq \Delta(b|\psi')$ holds trivially.

We can see that the objective function for each ϕ can be written as

$$f(S,\phi) = \begin{cases} |\{b,c\} \cap S| & \text{if } \phi(a) = \phi(b) = \phi(c) = +\\ \min\{1, |\{b,c\} \cap S|\} & \text{if } \phi(a) = \phi(b) = \phi(c) = -. \end{cases}$$

For each ϕ, the pointwise function $f(\cdot,\phi)$ is submodular, thus pointwise submodularity holds. Also, $f(\cdot,\phi)$ is monotone for each ϕ, thus pointwise monotonicity holds.

However, f is not set-adaptive submodular with respect to p. The expected marginal gain of $\{b,c\}$ is $\Delta(\{b,c\}|\varnothing) = 1.5$ at the beginning, but it increases after observing $\phi(a) = +$, that is, $\Delta(\{b,c\}|\{(a,+)\}) = 2$.

Based on Example 66, we can make an instance in which Lemma 3 of Chen and Krause [2013] does not hold.

Example 67. We consider adding dummy element d that does not affect the objective value to Example 66. Let $V = \{a,b,c,d\}$ be the ground set and $\mathcal{Y} = \{+,-\}$ the set of possible states. We define the objective function $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ in the same way as Example 66. We define the probability distribution of ϕ such that $\phi(d)$ is always $+$ and the same as Example 66 for the other elements. Let ψ be the partial realization obtained after selecting $\{a,d\}$ and observing $\phi_{\mathcal{J}}(\{a,d\})(a) = +$ and $\phi_{\mathcal{J}}(\{a,d\})(d) = +$. Then (f,p) satisfies adaptive submodularity, pointwise submodularity and pointwise monotonicity, but (g,q) does not satisfy adaptive submodularity as

$$\Delta(\{b,c\}|\psi) \geq \Delta(\{b,c\}|\varnothing),$$

which disproves Lemma 3 of Chen and Krause [2013].

Proposition 68. *The objective functions of generalized binary search, EC^2, and ALuMA are set-adaptive submodular. Also, if $\Phi(v)$ is independent for each $v \in V$, it is set-adaptive submodular.*

Proof. These objective functions satisfy policy-adaptive submodularity as proved in Fujii and Kashima [2016]. Since policy-adaptive submodularity implies set-adaptive submodularity, all of them are set-adaptive submodular with respect to p. □

Proposition 69. *The objective function of adaptive influence maximization with the independent cascade model and the full-adoption feedback model is set-adaptive submodular.*

Proof. Our proof is similar to the one for adaptive submodularity of the same function in Golovin and Krause [2011a]. Fix partial realizations $\psi \subseteq \psi'$ and a set of vertices $S \subseteq V \setminus \mathrm{dom}(\psi')$. Our goal is to prove $\Delta(v|\psi) \geq \Delta(v|\psi')$. We consider two random variables ϕ and ϕ', each of which conforms to the posterior distributions after ψ and ψ' are observed, respectively. Now we define $\mathbf{X} = (X_{uv})_{(u,v)\in E}$ to be the random variable that represents the states of all edges, that is, $X_{uv} = 1$ if (u,v) is alive and $X_{uv} = 0$ if (u,v) is dead. Note that if $\mathbf{X}$ is determined, ϕ is also determined based on $\mathbf{X}$. Similarly, we define $\mathbf{X}'$ that corresponds to ϕ'.

We make a coupling distribution $\hat{\mu}(\mathbf{X},\mathbf{X}')$ of $\mathbf{X}$ and $\mathbf{X}'$ as follows. For each edge $(u,v) \in E$ that has already been observed in ψ, the corresponding random variables $\mathbf{X}_{uv}$ and $\mathbf{X}'_{uv}$ are always equal to the observed state. For each edge $(u,v) \in E$ that is not observed in ψ', the corresponding random variable $\mathbf{X}_{uv}$ is determined according to the original distribution, i.e., $\Pr(\mathbf{X}_{uv} = 1) = p_{uv}$, and the

other one $\mathbf{X}'_{uv}$ is always equal to $\mathbf{X}_{uv}$, i.e., $\mathbf{X}_{uv} = \mathbf{X}'_{uv}$ always holds. For each edge $(u, v) \in E$ that has already been observed in ψ' but not in ψ, the corresponding random variable $\mathbf{X}'_{uv}$ is always equal to the observed state in ψ', and the other one $\mathbf{X}_{uv}$ conforms to the original distribution, i.e., $\Pr(\mathbf{X}_{uv} = 1) = p_{uv}$. As ϕ and ϕ' are determined by $\mathbf{X}$ and $\mathbf{X}'$, respectively, we can define the coupling distribution $\mu(\phi, \phi')$ corresponding to $\hat{\mu}$. The marginal distribution of μ coincides with each posterior distribution, i.e., $\sum_{\phi} \mu(\phi, \phi') = p(\phi|\psi)$ and $\sum_{\phi'} \mu(\phi, \phi') = p(\phi|\psi')$.

Here we show

$$f(\mathrm{dom}(\psi') \cup S, \phi') - f(\mathrm{dom}(\psi'), \phi') \geq f(\mathrm{dom}(\psi) \cup S, \phi) - f(\mathrm{dom}(\psi), \phi) \tag{4.1}$$

for each $(\phi, \phi') \in \mathrm{supp}(\mu)$. Let $B = \sigma(\mathrm{dom}(\psi), \phi)$ and $C = \sigma(\mathrm{dom}(\psi) \cup S, \phi)$ be the set of nodes influenced by $\mathrm{dom}(\psi)$ and $\mathrm{dom}(\psi) \cup S$ under realization ϕ. The left hand side of (4.1) is equal to $\hat{f}(B \cup D) - \hat{f}(B)$, where $D = C \setminus B$. Similarly, by defining B', C' and D' for ψ' and ϕ', we can write the right hand side of (4.1) is equal to $\hat{f}(B' \cup D') - \hat{f}(B')$.

We show $B \subseteq B'$ and $D' \subseteq D$. Let $v \in B$ be any vertex influenced in ψ. There exists a path consisting of only live edges from some vertex in $\mathrm{dom}(\psi)$ to v. Since all observed live edges in ψ are also observed and live in ψ', v is also activated in ψ' with the same path. Therefore we have $v \in B'$, which implies $B \subseteq B'$.

Let $v \in D'$ be any vertex such that there exists a path of live edges from some vertex $u \in S$ to v, but there does not exist a path of live edges from any vertex in $\mathrm{dom}(\psi')$ to v under realization ϕ'. Note that the state of each edge in this path from u to v is not observed in ψ', otherwise v also has already been observed in ψ' due to the full-adoption feedback model. Since the states of the edges unobserved in ψ' are the same under realizations ϕ and ϕ', we can see that there exists a path of live edges from u to v under realization ϕ as well. As ψ is a subset of ψ' and v is not activated in ψ', vertex v is not activated in ψ. Hence, we have $v \in D$.

Finally, since $\sum_{\phi'} \mu(\phi, \phi') = p(\phi|\psi)$ and $\sum_{\phi} \mu(\phi, \phi') = p(\phi|\psi')$, we have

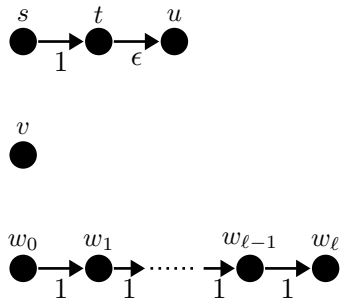
$$\begin{aligned}
\Delta(v|\psi) &= \sum_{\phi} \sum_{\phi'} \mu(\phi, \phi') \left\{ f(\mathrm{dom}(\psi) \cup S, \phi) - f(\mathrm{dom}(\psi), \phi) \right\} \\
&\geq \sum_{\phi'} \sum_{\phi} \mu(\phi, \phi') \left\{ f(\mathrm{dom}(\psi') \cup S, \phi') - f(\mathrm{dom}(\psi'), \phi') \right\} && \text{(due to (4.1))} \\
&= \Delta(v|\psi'),
\end{aligned}$$

which implies set-adaptive submodularity of the objective function. □

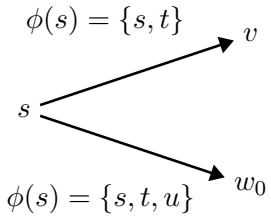
In the following, we provide an example where the objective function of an instance of adaptive influence maximization does not satisfy policy-adaptive submodularity. This example implies that policy-adaptive submodularity is a strictly stronger property than set-adaptive submodularity.

Example 70. Let $V = \{s, t, u, v, w_0, \cdots, w_\ell\}$ and $E = \{(s, t), (t, u)\} \cup \{(w_{i-1}, w_i) \mid i = 1, \cdots, \ell\}$ be the set of vertices and edges (See Figure 4.1). Assume the probabilities that each edge is alive are defined as $p_{(s,t)} = 1$, $p_{(t,u)} = \epsilon$, and $p_{(w_0,w_1)} = \cdots = p_{(w_{\ell-1},w_\ell)} = 1$. Proposition 69 implies this instance is set-adaptive submodular.

However, this is not policy-adaptive submodular as shown in the following. For simplicity, we write $\phi(v)$ as the set of all nodes reachable from v. Let us consider a policy π that first selects s, and if $\phi(s) = \{s, t\}$, proceeds to select v and if $\phi(s) = \{s, t, u\}$, proceeds to select w_0. The expected marginal gain of π when nothing is observed is $\Delta(\pi|\varnothing) = (1 - \epsilon)3 + \epsilon(4 + \ell) = 3 + \epsilon(\ell + 1)$, but after t is selected and $\phi(t) = \{t, u\}$ is observed, the expected marginal gain $\Delta(\pi|\{(t, \{t, u\})\}) = 2 + \ell$ is larger than $\Delta(\pi|\varnothing)$ for large ℓ. Therefore the objective function does not satisfy the policy-adaptive submodularity.



(a) Graph (V, E)



(b) policy π

Figure 4.1: An example that does not satisfy policy-adaptive submodularity under the independent cascade model and the full-adoption feedback model. The numbers below edges represent the probability that each edge is alive. The expected marginal gain of π is $\Delta(\pi|\varnothing) = 3+\epsilon(\ell+1)$ in the beginning, but after t is selected and the nodes influenced by t turned out to be $\phi(t) = \{t, u\}$, the expected marginal gain of π decreases $\Delta(\pi|(t, \{t, u\})) = 2 + \ell$.

Algorithm 7 Batch-mode adaptive greedy algorithm with α-approximate greedy selection

Input The objective function $f\colon 2^V \times \mathcal{Y}^V$ and the probability distribution $p \in \triangle^{\mathcal{Y}^V}$ given by a value oracle for the expected marginal gain $\Delta(\cdot|\cdot)$, the number of rounds $k \in \mathbb{Z}_{\geq 0}$, the family of feasible batches $\mathcal{J}$ given by an independence oracle, an α-approximation algorithm for monotone submodular maximization under constraint $S \in \mathcal{J}$ given by an oracle.

1: $\psi_0 \leftarrow \varnothing$.
2: **for** $i = 1, \cdots, k$ **do**
3: Apply an α-approximation algorithm to maximize $\Delta(S|\psi_{i-1})$ subject to $S \in \mathcal{J}$ and obtain an α-approximate solution S_i.
4: Query S_i and observe $\phi(v)$ for all $v \in S_i$.
5: $\psi_i \leftarrow \psi_{i-1} \cup \{(v, \phi(v)) \mid v \in S_i\}$.

These relationships among the properties are summarized in Figure 4.2.

4.5 Batch-mode Adaptive Greedy Algorithm

In this section, we describe a greedy algorithm for batch-mode adaptive optimization, which we call the *batch-mode adaptive greedy algorithm*. Starting with the empty set, this algorithm myopically selects a batch $S_i \in \mathcal{J}$ that approximately maximizes the marginal gain $\Delta(S_i|\psi_{i-1})$ at each round, where ψ_{i-1} is the partial observation obtained until the ith round. As shown later, the batch selection problem at each round can be reduced to constrained monotone submodular maximization, therefore we can use existing approximation algorithms. The detailed description of this algorithm is provided in Algorithm 7.

To bound the approximation ratio of the batch-mode adaptive greedy algorithm, our analysis takes the following two steps. The first step is to show that selecting a batch $S_i \in \mathcal{J}$ that approximately maximizes $\Delta(S_i|\psi_{i-1})$ is a constrained submodular maximization problem. The second step is to show that selecting batches that are approximately maximum at each round leads to an approximately optimal policy for the whole optimization problem.

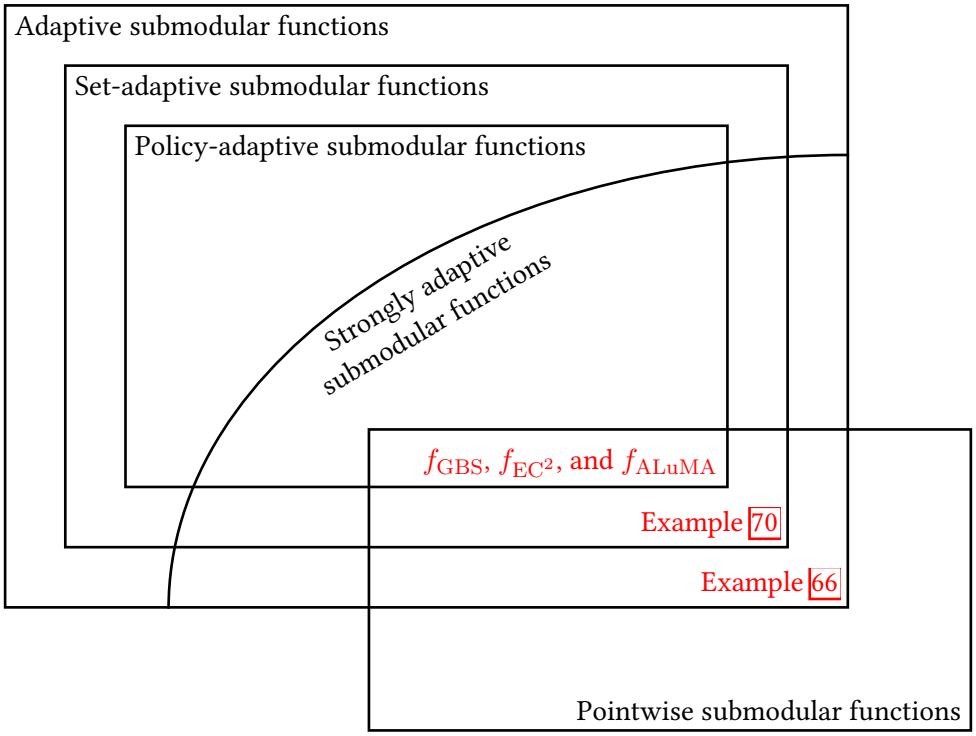


Figure 4.2: A diagram that indicates relationships between adaptive submodularity, set-adaptive submodularity, policy-adaptive submodularity, strong adaptive submodularity, and pointwise submodularity.

4.5.1 Greedy Selection

Here we show the submodularity and monotonicity of the expected marginal gain $\Delta(S|\psi)$ when we see it as a set function of S. Almost the same result is given in Chen and Krause [2013], but their assumptions on the objective function are different from ours. Here we provide the full proofs for completeness.

Proposition 71. *Suppose $f\colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ is adaptive submodular with respect to p. Then a set function $g\colon 2^V \to \mathbb{R}_{\geq 0}$ defined as $g(X) \coloneqq \Delta(X|\psi)$ for every $X \subseteq V$ is submodular for any partial realization ψ.*

Proof. It suffices to prove for any $A \subseteq B \subseteq V$ and $v \in V \setminus B$, it holds that $g(v|B) \leq g(v|A)$. Let $D = \mathrm{dom}(\psi)$. From the definition, we have

$$\begin{aligned} g(v|B) &= \mathbb{E}[f(D \cup B \cup \{v\}, \Phi) - f(D, \Phi) \mid \Phi \sim \psi] - \mathbb{E}[f(D \cup B, \Phi) - f(D, \Phi) \mid \Phi \sim \psi] \\ &= \mathbb{E}[f(D \cup B \cup \{v\}, \Phi) - f(D \cup B, \Phi) \mid \Phi \sim \psi]. \end{aligned}$$

Let $\mathcal{P}_A = \{\psi \mid \mathrm{dom}(\psi) = A\}$ and $\mathcal{P}_B = \{\psi \mid \mathrm{dom}(\psi) = B\}$. Then we have

$$\begin{aligned} g(v|B) &= \sum_{\psi_B \in \mathcal{P}_B} p(\psi_B|\psi)\mathbb{E}[f(D \cup B \cup \{v\}, \Phi) - f(D \cup B, \Phi) \mid \Phi \sim \psi \cup \psi_B] \\ &= \sum_{\psi_B \in \mathcal{P}_B} p(\psi_B|\psi)\Delta(v|\psi \cup \psi_B). \end{aligned}$$

In the same way, we have

$$g(v|A) = \sum_{\psi_A \in \mathcal{P}_A} p(\psi_A|\psi)\Delta(v|\psi \cup \psi_A).$$

Finally, by using the above equations, we obtain

$$\begin{aligned}
g(v|B) &= \sum_{\psi_B \in \mathcal{P}_B} p(\psi_B|\psi)\Delta(v|\psi \cup \psi_B) \\
&= \sum_{\psi_A \in \mathcal{P}_A} p(\psi_A|\psi) \sum_{\psi_B \in \mathcal{P}_B \colon \psi_B \sim \psi_A} p(\psi_B|\psi_A \cup \psi)\Delta(v|\psi \cup \psi_B) \\
&\leq \sum_{\psi_A \in \mathcal{P}_A} p(\psi_A|\psi) \sum_{\psi_B \in \mathcal{P}_B \colon \psi_B \sim \psi_A} p(\psi_B|\psi_A \cup \psi)\Delta(v|\psi \cup \psi_A) \\
&= \sum_{\psi_A \in \mathcal{P}_A} p(\psi_A|\psi)\Delta(v|\psi \cup \psi_A) \\
&= g(v|A),
\end{aligned}$$

where the inequality is due to adaptive submodularity of f with respect to p. □

Proposition 72. *Suppose $f \colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ is adaptive monotone with respect to p. Then a set function $g \colon 2^V \to \mathbb{R}_{\geq 0}$ defined as $g(X) \coloneqq \Delta(X|\psi)$ for every $X \subseteq V$ is monotone for any partial realization ψ.*

Proof. It suffices to prove for any $A \subseteq V$ and $v \in V \setminus A$, it holds that $g(v|A) \geq 0$. Let $D = \mathrm{dom}(\psi)$. Let $\mathcal{R}(A)$ be all possible partial realizations of the states of A. Then we have

$$\begin{aligned}
g(v|A) &= \mathbb{E}[f(D \cup A \cup \{v\}, \Phi) - f(D \cup A, \Phi) \mid \Phi \sim \psi] \\
&= \sum_{\psi_A \in \mathcal{R}(A)} p(\psi_A|\psi)\mathbb{E}[f(D \cup A \cup \{v\}, \Phi) - f(D \cup A, \Phi) \mid \Phi \sim \psi \cup \psi_A] \\
&= \sum_{\psi_A \in \mathcal{R}(A)} p(\psi_A|\psi)\Delta(v|\psi \cup \psi_A) \\
&\geq 0,
\end{aligned}$$

where the inequality is due to adaptive monotonicity of f with respect to p. □

From these two propositions, we can claim that the batch selection problem at each round can be reduced to constrained monotone submodular maximization. If there exists an approximation algorithm for maximizing a monotone submodular function subject to the batch constraint $S \in \mathcal{J}$, we can solve this problem approximately by using this algorithm.

4.5.2 Reduction from Batch-mode Setting to Fully Adaptive Setting

To bound the approximation ratio of the batch-mode adaptive greedy algorithm, we reduce the batch-mode setting to the fully adaptive setting and apply the existing result on the adaptive greedy algorithm. This idea was first used by Chen and Krause [2013] for the case of $\mathcal{J} = \{S \subseteq V \mid |S| \leq k\}$.

Definition 73 (Reduction from the batch-mode setting to the fully adaptive setting)**.** Given an instance of batch-mode adaptive optimization $(f, p, \mathcal{J})$, we can make an instance of fully adaptive optimization (g, q) as follows. Let $\mathcal{Z} = \bigcup_{S \in \mathcal{J}} \mathcal{Y}^S$ be the set of all possible outcomes for each batch $S \in \mathcal{J}$. We can define a map $\tau \colon \mathcal{Y}^V \to \mathcal{Z}^{\mathcal{J}}$ that returns realization $\tau(\phi)$ for instance (g, p) when given realization ϕ for instance (f, p) by setting $\tau(\phi)(S)(v) = \phi(v)$ for each $S \in \mathcal{J}$ and $v \in S$. Let $\phi_{\mathcal{J}} = \tau(\phi)$ and $\Phi_{\mathcal{J}}$ be

a random variable associated with it. We define $\phi_{\mathcal{J}}$ to be valid if there exists $\phi\colon V \to \mathcal{Y}$ such that for all $S \in \mathcal{J}$ and $v \in S$, it holds that $\phi_{\mathcal{J}}(S)(v) = \phi(v)$. For any valid $\phi_{\mathcal{J}}$, we can define the inverse map τ^{-1} such that $\tau^{-1}(\tau(\phi)) = \phi$. By using this notation, we can define a set function $g\colon 2^{\mathcal{J}} \times \mathcal{Z}^{\mathcal{J}} \to \mathbb{R}$ and a probability distribution $q\colon \mathcal{Z}^{\mathcal{J}} \to \mathbb{R}_{\geq 0}$. For each subset $\mathcal{S} \subseteq \mathcal{J}$ and realization $\phi_{\mathcal{J}}$, we define set function $g\colon 2^{\mathcal{J}} \times \mathcal{Z}^{\mathcal{J}} \to \mathbb{R}$ as

$$g(\mathcal{S}, \phi_{\mathcal{J}}) = f\left(\bigcup \mathcal{S}, \tau^{-1}(\phi_{\mathcal{J}})\right)$$

if $\phi_{\mathcal{J}}$ is valid and $g(\mathcal{S}, \phi_{\mathcal{J}}) = 0$ if $\phi_{\mathcal{J}}$ is invalid. Similarly, we define probability distribution q as

$$q(\phi_{\mathcal{J}}) = p(\tau^{-1}(\phi_{\mathcal{J}}))$$

if $\phi_{\mathcal{J}}$ is valid and $q(\phi_{\mathcal{J}}) = 0$ if $\phi_{\mathcal{J}}$ is invalid.

By using this reduction, we can bound the approximation ratio of the batch-mode adaptive greedy algorithm as follows.

Theorem 74. *Assume f is set-adaptive submodular and adaptive monotone with respect to p. Suppose an α-approximation algorithm for monotone submodular maximization subject to $X \in \mathcal{J}$ is used for the batch selection at each round. If π is a policy that encodes the batch-mode adaptive greedy algorithm with k rounds and π^* is an optimal batch-mode policy with k rounds, then we have*

$$f_{\mathrm{avg}}(\pi) \geq (1 - \mathrm{e}^{-\alpha}) f_{\mathrm{avg}}(\pi^*).$$

Proof. Given function f and probability distribution p, we obtain g and q by applying the reduction described in Definition 73. Here we prove g is adaptive submodular and adaptive monotone with respect to q by using the set adaptive submodularity and adaptive monotonicity of f and p. Let $S \in \mathcal{J}$ be any batch and $\psi_{\mathcal{J}}$ be any partial realization of $\phi_{\mathcal{J}}$. Let ψ be the corresponding partial realization for $\psi_{\mathcal{J}}$. The expected marginal gain can be written as

$$\Delta_{g,q}(S|\psi_{\mathcal{J}}) = \mathbb{E}_{\Phi}\left[f\left(\mathrm{dom}(\psi) \cup S, \Phi\right) - f\left(\mathrm{dom}(\psi), \Phi\right) | \Phi \sim \psi\right].$$

For any partial realizations $\psi_{\mathcal{J}} \subseteq \psi'_{\mathcal{J}}$, we have

$$\begin{aligned}
\Delta_{g,q}(S|\psi_{\mathcal{J}}) &= \Delta_{f,p}(S|\psi) \\
&\geq \Delta_{f,p}(S|\psi') && \text{(due to set-adaptive submodularity of } (f,p)) \\
&= \Delta_{g,q}(S|\psi'_{\mathcal{J}}),
\end{aligned}$$

which implies adaptive submodularity of (g, q). Next, we show adaptive monotonicity of (g, q). Let $S \setminus \mathrm{dom}(\psi) = \{s_1, \cdots, s_\ell\}$ by ordering elements in S arbitrarily and $S_i = \{s_1, \cdots, s_i\}$. If $\mathcal{P}_i = \{\psi \mid \mathrm{dom}(\psi) = S_i\}$ is the set of all partial realizations with domain S_i, we have

$$\begin{aligned}
&\Delta_{g,q}(S|\psi_{\mathcal{J}}) \\
&= \Delta_{f,p}(S|\psi) \\
&= \sum_{i=1}^{\ell} \mathbb{E}_{\Phi}[f(\mathrm{dom}(\psi) \cup S_{i-1} \cup \{s_i\}, \Phi) - f(\mathrm{dom}(\psi) \cup S_{i-1}, \Phi) | \Phi \sim \psi] \\
&= \sum_{i=1}^{\ell} \sum_{\psi_{i-1} \in \mathcal{P}_{i-1}} p(\psi_{i-1}|\psi) \mathbb{E}_{\Phi}[f(\mathrm{dom}(\psi \cup \psi_{i-1}) \cup \{s_i\}, \Phi) - f(\mathrm{dom}(\psi \cup \psi_{i-1}), \Phi) | \Phi \sim \psi \cup \psi_{i-1}]
\end{aligned}$$

$$= \sum_{i=1}^{\ell} \sum_{\psi_{i-1} \in \mathcal{P}_{i-1}} p(\psi_{i-1}|\psi) \Delta_{f,p}(s_i|\psi \cup \psi_{i-1})$$

$\geq 0,$ (due to adaptive monotonicity of (f, p))

which implies adaptive monotonicity of (g, q).

The batch-mode adaptive greedy algorithm for (f, p) can be regarded as an α-approximate adaptive greedy algorithm for (g, q). Therefore, from Theorem 33, it achieves at least $(1 - \exp(-\alpha))$ times the objective value achieved by an optimal fully adaptive policy for (g, q). Since any batch-mode policy for (f, p) is identified with a fully adaptive policy for (g, q), we conclude the statement. □

As direct consequences of this theorem and results on constrained monotone submodular maximization, we obtain bounds on the approximation ratios as follows.

Corollary 75. *If $X \in \mathcal{J}$ is a matroid constraint, the batch-mode adaptive greedy with the continuous greedy algorithm [Călinescu et al., 2011] achieves $(1 - \exp(-(1 - 1/\mathrm{e})))$-approximation. If $X \in \mathcal{J}$ is a knapsack constraint, the batch-mode adaptive greedy with the greedy algorithm with partial enumeration [Sviridenko, 2004] achieves $(1 - \exp(-(1 - 1/\mathrm{e})))$-approximation. If $X \in \mathcal{J}$ is a p-system constraint, the batch-mode adaptive greedy with the greedy algorithm [Călinescu et al., 2011] achieves $(1 - \exp(-1/(p + 1)))$-approximation.*

4.6 Beyond Set-Adaptive Submodularity

In the last section, we have assumed set-adaptive submodularity of the objective function. However, as illustrated in Chapter 3, there are still many problems that cannot be dealt with under adaptive submodularity, such as adaptive influence maximization in the triggering model or adaptive feature selection. Since set-adaptive submodularity is a stronger condition than adaptive submodularity, these problems cannot be dealt with under set-adaptive submodularity as well. In this section, we tackle these problems by utilizing the notion of adaptive submodularity ratio developed in Chapter 3. We provide a lower bound on the approximation ratio of the batch-mode adaptive greedy algorithm in terms of the adaptive submodularity ratio and the supermodularity ratio of the objective function.

The starting point is the bound on the approximation ratio of the adaptive greedy algorithm, which is shown in Chapter 3. First, we need to extend Theorem 43 to the adaptive greedy algorithm with α-approximate greedy selection, which at each step selects element $v \in V$ such that $\Delta(v|\psi) \geq \alpha \max_{v \in V} \Delta(v|\psi)$, where ψ is the partial realization observed so far.

Theorem 76. *Suppose $f \colon 2^V \times \mathcal{Y}^V \to \mathbb{R}_{\geq 0}$ is adaptive monotone with respect to p. Let π be a policy representing the adaptive greedy algorithm with α-approximate greedy selection until ℓ step. Then, for any policy $\pi^* \in \Pi_k$, it holds that*

$$f_{\mathrm{avg}}(\pi) \geq \left(1 - \exp\left(-\alpha \gamma_{\ell,k} \ell / k\right)\right) f_{\mathrm{avg}}(\pi^*),$$

where $\gamma_{\ell,k}$ is the adaptive submodularity ratio of f with respect to p.

Proof. The outline of the proof is the same as the proof of Theorem 43. The only difference is that we use

$$f_{\mathrm{avg}}(\pi_{[i+1]}) - f_{\mathrm{avg}}(\pi_{[i]}) \geq \mathbb{E}\left[\alpha \max_{v \in V} \Delta(v|\Psi)\right]$$

instead of $f_{\mathrm{avg}}(\pi_{[i+1]}) - f_{\mathrm{avg}}(\pi_{[i]}) = \mathbb{E}\left[\max_{v \in V} \Delta(v|\Psi)\right]$. Then we obtain the bound in the statement. □

By using the reduction from the batch-mode setting to the fully adaptive setting, we obtain the following result.

Theorem 77. *Suppose the original objective function f is adaptive monotone with respect to p and the supermodularity ratio of its expected value $\mathbb{E}_\Phi[f(\cdot,\Phi)]$ is $\beta_{k,\psi}$.*

1. *The batch selection at each round can be reduced to finding $X \in \mathcal{J}$ that maximizes a monotone function whose submodularity ratio is bounded as $\gamma_{U,k} \geq \min_{\psi_U} \gamma_{\psi\cup\psi_U,k}(f,p)$.*
2. *The batch-mode adaptive greedy algorithm with α-approximate greedy selection that is executed for k rounds achieves the objective value at least $1-\exp\left(-\alpha \min_{\psi:\ |\psi|\leq(k-1)\ell} \beta_{\varnothing,\ell}(\Delta_{f,p}(\cdot|\psi))\gamma_{k\ell,k\ell}(f,p)\right)$ times the objective value achieved by an optimal batch-mode policy with k rounds, where $\ell = \max\{|S|\colon S\in\mathcal{J}\}$.*

Proof. First, we show the batch selection can be reduced to maximizing a monotone function with a bounded submodularity ratio. Suppose ψ is a partial realization obtained so far for the original ground set V. The batch selection is the problem of finding $X \in \mathcal{J}$ that approximately maximizes $\Delta_{f,p}(X|\psi)$. From Proposition 72, $\Delta_{f,p}(\cdot|\psi)$ is monotone. From the definition, the submodularity ratio of $\Delta_{f,p}(\cdot|\psi)$ is

$$\gamma_{U,k}(\Delta_{f,p}(\cdot|\psi)) = \min_{L\subseteq U,\ S\colon |S|\leq k} \frac{\sum_{v\in S}\{\Delta_{f,p}(L\cup\{v\}|\psi) - \Delta_{f,p}(L|\psi)\}}{\Delta_{f,p}(L\cup S|\psi) - \Delta_{f,p}(L|\psi)}.$$

Let $L\subseteq U$ and $S\subseteq V$ such that $|S|\leq k$. Let ψ_L be a partial realization such that $\mathrm{dom}(\psi_L) = L$ and Ψ_L the random variable associated with ψ_L. We can transform the numerator of the submodularity ratio as follows.

$$\begin{aligned}
&\sum_{v\in S}\{\Delta_{f,p}(L\cup\{v\}|\psi) - \Delta_{f,p}(L|\psi)\}\\
&= \sum_{v\in S}\mathbb{E}_\Phi[f(\mathrm{dom}(\psi)\cup L\cup\{v\}) - f(\mathrm{dom}(\psi)\cup L)|\Phi\sim\psi]\\
&= \sum_{v\in S}\mathbb{E}_{\Psi_L}[\mathbb{E}_\Phi[f(\mathrm{dom}(\psi)\cup\mathrm{dom}(\Psi_L)\cup\{v\}) - f(\mathrm{dom}(\psi)\cup\mathrm{dom}(\Psi_L))|\Phi\sim\psi\cup\Psi_L]]\\
&= \sum_{v\in S}\mathbb{E}_{\Psi_L}[\Delta_{f,p}(v|\psi\cup\Psi_L)]\\
&= \mathbb{E}_{\Psi_L}\left[\sum_{v\in S}\Delta_{f,p}(v|\psi\cup\Psi_L)\right]
\end{aligned}$$

By considering a non-adaptive policy that always selects S, from the definition of the adaptive submodularity ratio, we have

$$\sum_{v\in S}\Delta_{f,p}(v|\psi\cup\psi_L) \geq \gamma_{\psi\cup\psi_L,k}(f,p)\Delta_{f,p}(S|\psi\cup\psi_L)$$

for all ψ_L. Therefore, we obtain

$$\begin{aligned}
&\sum_{v\in S}\{\Delta_{f,p}(L\cup\{v\}|\psi) - \Delta_{f,p}(L|\psi)\}\\
&\geq \mathbb{E}_{\Psi_L}\left[\gamma_{\psi\cup\Psi_L,k}(f,p)\ \Delta_{f,p}(S|\psi\cup\Psi_L)\right]\\
&\geq \min_{\psi_L}\gamma_{\psi\cup\psi_L,k}(f,p)\ \mathbb{E}_{\Psi_L}\left[\Delta_{f,p}(S|\psi\cup\Psi_L)\right]
\end{aligned}$$

$$= \min_{\psi_L} \gamma_{\psi\cup\psi_L,k}(f,p)\ \mathbb{E}_{\Psi_L}\left[\mathbb{E}_{\Phi}[f(\mathrm{dom}(\psi)\cup\mathrm{dom}(\Psi_L)\cup S) - f(\mathrm{dom}(\psi)\cup\mathrm{dom}(\Psi_L))|\Phi\sim\psi\cup\Psi_L]\right]$$
$$= \min_{\psi_L} \gamma_{\psi\cup\psi_L,k}(f,p)\ \mathbb{E}_{\Phi}[f(\mathrm{dom}(\psi)\cup L\cup S) - f(\mathrm{dom}(\psi)\cup L)|\Phi\sim\psi]$$
$$= \min_{\psi_L} \gamma_{\psi\cup\psi_L,k}(f,p)\ \left\{\Delta_{f,p}(L\cup S|\psi) - \Delta_{f,p}(L|\psi)\right\}.$$

Now we can bound the submodularity ratio as follows.

$$\begin{aligned}\gamma_{U,k}(\Delta_{f,p}(\cdot|\psi)) &\geq \min_{L\subseteq U}\min_{\psi_L}\gamma_{\psi\cup\psi_L,k}(f,p)\\ &= \min_{\psi_U}\gamma_{\psi\cup\psi_U,k}(f,p),\end{aligned}$$

where we use the fact that the adaptive submodularity ratio is monotonically non-increasing with respect to the first subsript $\psi\cup\psi_U$.

Next, we bound the approximation ratio of the batch-mode adaptive greedy algorithm. By using the reduction from the batch-mode setting to the fully adaptive setting described in Definition 73, we obtain the problem instance (g,q) induced by batches $\mathcal{J}$. To distinguish the realizations for the original instance and the modified instance, we use Φ_V and $\Phi_{\mathcal{J}}$ for each of them. Similarly, ψ_V and π_V are for the original instance (f,p) and $\psi_{\mathcal{J}}$ and $\pi_{\mathcal{J}}$ are for the modified instance (g,q). Now we can write $\gamma_{k,k}(g,q)$ as

$$\gamma_{k,k}(g,q) = \min_{\psi_{\mathcal{J}}:\,|\psi_{\mathcal{J}}|\leq k,\ \pi_{\mathcal{J}}\in\Pi_{\mathcal{J},k}} \frac{\sum_{S\in\mathcal{J}}\Pr(S\in E(\pi_{\mathcal{J}},\Phi_{\mathcal{J}})|\Phi_{\mathcal{J}}\sim\psi_{\mathcal{J}})\Delta_{g,q}(S|\psi_{\mathcal{J}})}{\Delta_{g,q}(\pi_{\mathcal{J}}|\psi_{\mathcal{J}})}$$

Let $\psi^*_{\mathcal{J}}$ and $\pi^*_{\mathcal{J}}$ be a minimizer of the above definition. Note that a batch-mode policy $\pi^*_{\mathcal{J}}$ of height k can be regarded as a policy π^* of height at most $k\ell$ for instance (f,p). Similarly, partial realization $\psi^*_{\mathcal{J}}$ for (g,q) can be regarded as a partial realization $\tau(\psi^*_{\mathcal{J}})$ for instance (f,p). Then we can rewrite the above definition as follows.

$$\begin{aligned}\gamma_{k,k}(g,q) &= \frac{\sum_{S\in\mathcal{J}}\Pr(S\in E(\pi^*_{\mathcal{J}},\Phi_{\mathcal{J}})|\Phi_{\mathcal{J}}\sim\psi^*_{\mathcal{J}})\Delta_{g,q}(S|\psi^*_{\mathcal{J}})}{\Delta_{g,q}(\pi^*_{\mathcal{J}}|\psi^*_{\mathcal{J}})}\\ &\geq \frac{\sum_{S\in\mathcal{J}}\Pr(S\in E(\pi^*_{\mathcal{J}},\Phi_{\mathcal{J}})|\Phi_{\mathcal{J}}\sim\psi^*_{\mathcal{J}})\beta_{\varnothing,\ell}(\Delta(\cdot|\psi^*_{\mathcal{J}}))\sum_{v\in S}\Delta_{g,q}(v|\psi^*_{\mathcal{J}})}{\Delta_{g,q}(\pi^*_{\mathcal{J}}|\psi^*_{\mathcal{J}})}\\ &\geq \min_{\psi:\,|\psi|\leq(k-1)\ell}\beta_{\varnothing,\ell}(\Delta(\cdot|\psi))\frac{\sum_{v\in V}\Pr(v\in E(\pi^*,\Phi)|\Phi\sim\psi^*)\Delta_{g,q}(v|\psi^*)}{\Delta_{g,q}(\pi^*|\psi^*)}\\ &\geq \min_{\psi:\,|\psi|\leq(k-1)\ell}\beta_{\varnothing,\ell}(\Delta(\cdot|\psi))\min_{\psi:\,|\psi|\leq k\ell,\ \pi\in\Pi_{k\ell}}\frac{\sum_{v\in V}\Pr(v\in E(\pi,\Phi)|\Phi\sim\psi)\Delta_{f,p}(v|\psi)}{\Delta_{f,p}(\pi|\psi)}\\ &= \min_{\psi:\,|\psi|\leq(k-1)\ell}\beta_{\varnothing,\ell}(\Delta_{f,p}(\cdot|\psi))\gamma_{k\ell,k\ell}(f,p).\end{aligned}$$

Finally, we substitute this bound into Theorem 76 and obtain

$$f_{\mathrm{avg}}(\pi) \geq \left(1-\exp\left(-\alpha\min_{\psi:\,|\psi|\leq(k-1)\ell}\beta_{\varnothing,\ell}(\Delta_{f,p}(\cdot|\psi))\gamma_{k\ell,k\ell}(f,p)\right)\right)f_{\mathrm{avg}}(\pi^*),$$

where π is a batch-mode adaptive greedy with α-approximate greedy selection and π^* is an optimal batch-mode policy for instance (f,p). □

4.7 Other Extensions

In this section, we consider extensions and variants of batch-mode adaptive optimization under the assumption of set-adaptive submodularity.

4.7.1 Outer Matroid Constraints

Until now, we have considered batch-mode adaptive optimization with a constant number of rounds. In other words, the decision-maker can select k batches regardless of which batches are selected at each round. Here we consider a problem setting where a constraint is imposed not on the number of rounds, but the union of the selected batches.

In particular, we consider a matroid constraint on the union of the selected batches, which we call an *outer matroid constraint*. Let $\mathcal{M} = (V, \mathcal{I})$ be a matroid on the ground set V. An outer matroid constraint requires the selected batches $\mathcal{S} \subseteq \mathcal{J}$ to satisfy $\bigcup \mathcal{S} \in \mathcal{I}$.

Besides an outer matroid constraint, we impose a constraint that requires the selected batches to be disjoint. The reason why we need the disjointness is that if the batches may intersect, the problem would be virtually reduced to the fully adaptive setting. For example, we consider the case the batch constraint is the simplest one that allows all batches of size ℓ, i.e., $\mathcal{J} = \{S \subseteq V : |S| = \ell\}$. Suppose the decision-maker has selected any batch S_1 at the first round. Let S_1' be an arbitrary subset of S_1 of size $\ell - 1$. From the second round to the last round, the decision-maker can act as if in the fully adaptive setting by selecting a batch $S \in \mathcal{J}$ with $S = S_1' \cup \{v\} \in \mathcal{J}$ instead of selecting the element $v \in V$ that she actually wants in the fully adaptive setting. To avoid such a situation, we need the disjointness among the selected batches.

For this problem, the batch-mode adaptive greedy algorithm can be described as follows. At the ith round, the algorithm selects a batch $S_i \in \mathcal{J}$ that approximately maximizes $\Delta(S_i|\psi_{i-1})$ without violating either the outer matroid constraint or the disjointness constraint. In other words, the batch at the ith round must satisfy $S_i \in \mathcal{J}$, $(\bigcup_{j=1}^{i-1} S_j) \cup S_i \in \mathcal{I}$, and $S_i \cap \left(\bigcup_{j=1}^{i-1} S_j\right) = \varnothing$. In the case where $\mathcal{J}$ consists of all batches of size ℓ, we can solve the bach selection by an approximation algorithm for monotone submodular maximization under a matroid constraint. Since $\Delta(S_i|\psi_{i-1})$ is monotone and submodular as a set function of S_i from Propositions 71 and 72, it is sufficient to prove the constraint is a matroid constraint.

Proposition 78. *If the set of feasible batches consists of all batches of size ℓ, i.e., $\mathcal{J} = \{S \subseteq V : |S| = \ell\}$, the intersection of the outer matroid constraint, the batch constraint, and the disjointness constraint at each round is the base family of some matroid.*

Proof. The outer matroid constraint that must be satisfied by the batch at the ith round is

$$\mathcal{I}_i = \left\{ S \subseteq V \setminus \left(\bigcup_{j=1}^{i-1} S_j \right) \;\middle|\; \left(\bigcup_{j=1}^{i-1} S_j \right) \cup S \in \mathcal{I} \right\},$$

which is the independence set family of the independence system by contracting $\bigcup_{j=1}^{i-1} S_j$ of matroid $(V, \mathcal{I})$. Since matroids are closed under the contraction operation, $\mathcal{I}_i$ is the independence set family of a matroid. By taking the intersection with the batch constraint $\mathcal{J}$ and the disjointness constraint, the decision-maker can select any independent set in $\mathcal{I}_i$ of size ℓ. Since matroids are closed under the truncation operation, $\{X \in \mathcal{I}_i \mid |X| \leq \ell\}$ is still the independence set family of a matroid. The set of feasible batches is the base family of this matroid. □

Next, we bound the approximation ratio of the batch-mode adaptive greedy algorithm. Similarly to the proof of Theorem 74, we use the reduction from the batch-mode setting to the fully adaptive setting. But now, it is reduced to the fully adaptive setting with a *k-uniform matroid matching* constraint. A k-uniform matroid matching constraint is known to be a special case of k-system constraints, which was proved by Lee et al. [2013]. This analysis can be applied not only to the setting with $\mathcal{J} = \{S \subseteq V : |S| = \ell\}$, but also for any setting where the batch selection can be approximated at each round.

Algorithm 8 Batch-mode adaptive greedy for an outer matroid constraint

Input The objective function $f\colon 2^V \times \mathcal{Y}^V$ and the probability distribution $p \in \triangle^{\mathcal{Y}^V}$ given by a value oracle for the expected marginal gain $\Delta(\cdot|\cdot)$, the outer matroid constraint $\mathcal{I}$ given by an independence oracle, the size of each batch ℓ.

1: $\psi_0 \leftarrow \varnothing$.
2: Let r be the rank of matroid $(V, \mathcal{I})$.
3: **while** $i = 1, \cdots, \lfloor r/\ell \rfloor$ **do**
4: Let $\mathcal{I}_i = \left\{ S \subseteq V \setminus \left(\bigcup_{j=1}^{i-1} S_j\right) \;\middle|\; \left(\bigcup_{j=1}^{i-1} S_j\right) \cup S \in \mathcal{I} \right\}$ be the contraction of $\mathcal{I}$ to $V \setminus \bigcup_{j=1}^{i-1} S_j$.
5: Apply an α-approximation algorithm to maximize $\Delta(S|\psi_{i-1})$ subject to $S \in \mathcal{I}_i \cap \mathcal{J}$ and obtain an α-approximate solution S_i.
6: Query S_i and observe $\phi(v)$ for all $v \in S_i$.
7: $\psi_i \leftarrow \psi_{i-1} \cup \{(v, \phi(v)) \mid v \in S_i\}$.

Theorem 79. *Assume f is set-adaptive submodular and adaptive monotone with respect to p. Suppose the outer constraint is a matroid constraint and the selected batches must be disjoint, and the batch selection at each round is α-approximation. If π be the policy that encodes the batch-mode adaptive greedy algorithm and π^* an optimal policy, then we have*

$$f_{\mathrm{avg}}(\pi) \geq \frac{\alpha}{k_{\max} + \alpha} f_{\mathrm{avg}}(\pi^*),$$

where $k_{\max} = \max_{X \in \mathcal{J}} |X|$.

Proof. Similarly to the proof of Theorem 74, we obtain the instance (g, q) in the fully adaptive setting by using the reduction described in Definition 73. From set-adaptive submodularity and adaptive monotonicity, we can show adaptive submodularity and adaptive monotonicity of (g, q) in the same way as the proof of Theorem 74. By considering the constraint of the reduced instance, the set family of all feasible sets of batches is

$$\mathfrak{I} = \{\mathcal{S} \subseteq \mathcal{J} \mid \forall S, T \in \mathcal{S},\ S \cap T = \varnothing \text{ and } \bigcup \mathcal{S} \in \mathcal{I}\}. \tag{4.2}$$

This is already the independence set of a matroid matching, but not $k_{\max}$-uniform. To reduce it to a $k_{\max}$-uniform matroid matching constraint, we add dummy elements that do not give any effect to the value of f. We obtain V by adding $k_{\max} - |S|$ dummy elements to V for each batch $S \in \mathcal{J}$ and obtain S by augmenting S with these dummy elements so that the size of S is $k_{\max}$. Then we have $|S'| = k_{\max}$ for all $S' \in \mathcal{J}'$, where $\mathcal{J}'$ is the set of S' obtained by adding dummy elements to S for all $S \in \mathcal{J}$. We also define $\mathcal{I}' \subseteq 2^{V'}$ as the direct sum of $\mathcal{I}$ and the free matroid on the set of all dummy items. Now we can define $\mathfrak{I}' \subseteq 2^{\mathcal{J}'}$ by replacing S with S', $\mathcal{J}$ with $\mathcal{J}'$, and $\mathcal{I}$ with $\mathcal{I}'$ of Equation (4.2). Since $\mathfrak{I}'$ is the independence set family of a $k_{\max}$-uniform matroid matching, so is $\mathfrak{I}$. From Theorem 38, the adaptive greedy algorithm with α-approximate greedy selection for a $k_{\max}$-system constraint achieves $\alpha/(k_{\max} + \alpha)$-approximation, which implies the statement. □

4.7.2 Online Setting

Here we consider the setting where the ground set is different at each round, which we call the *online setting*. In the online setting, the decision-maker is given a ground set V_i and a batch constraint $\mathcal{J}_i \subseteq 2^{V_i}$ at the ith round, and must select a batch $S_i \in \mathcal{J}_i$. Just after selecting batch S_i, the decision-maker observes the states $(\phi(v))_{v \in S_i}$ of the elements in S_i. A difficult point of the online setting is that the

Algorithm 9 Batch-mode adaptive greedy algorithm for the online setting

$\psi_0 \leftarrow \varnothing$.
for $i = 1, \cdots, k$ **do**
  The ground set V_i for the ith round appears.
  Apply an α-approximation algorithm to maximize $\Delta(S|\psi_{i-1})$ subject to $S \in \mathcal{J}_i$ and obtain an α-approximate solution S_i.
  Query S_i and observe $\phi(v)$ for all $v \in S_i$.
  $\psi_i \leftarrow \psi_{i-1} \cup \{(v, \phi(v)) \mid v \in S_i\}$.

decision-maker must select S_i without knowing the information about the rounds in the future. Let $V = V_1 \cup \cdots \cup V_k$ be the total ground set. The objective function $f\colon 2^V \to \mathbb{R}_{\geq 0}$ and the probability distribution of $\phi\colon V \to \mathcal{Y}$ defined on the total ground set are assumed to be set-adaptive submodular and adaptive monotone. We assume that at the ith round, the access to an oracle for the expected marginal gain is limited to the part for the elements that have already appeared, that is, $V = V_1 \cup \cdots \cup V_T$.

For the online setting, we devise the batch-mode adaptive greedy algorithm as well. At the ith round, given V_i and $\mathcal{J}_i$, this algorithm selects a batch $S_i \in \mathcal{J}_i$ that satisfies

$$\Delta(S_i|\psi_{i-1}) \geq \alpha \max_{S \in \mathcal{J}_i} \Delta(S|\psi_{i-1}),$$

where ψ_{i-1} is the partial observation obtained just before the ith round. The algorithmic description is given in Algorithm 9. From Proposition 71 and Proposition 72, the batch selection problem can be reduced to constrained monotone submodular maximization problem.

Here we analyze the batch-mode adaptive greedy algorithm in the online setting by comparing it with an optimal batch-mode policy in the offline setting, not in the online setting. We can consider the corresponding offline setting where the decision-maker can select a batch from each $\mathcal{J}_i$ *in any order*. Note that in the online setting, the decision-maker must select a batch from $\mathcal{J}_i$ *in the fixed order*, that is, first from $\mathcal{J}_1$, and second from $\mathcal{J}_2$, and so on. In the corresponding offline setting, the decision-maker can select one batch from $\mathcal{J}_i$ for each i, but the order of $\mathcal{J}_i$s is arbitrary. We can prove that the batch-mode adaptive greedy algorithm in the online setting is competitive with an optimal batch-mode policy in the corresponding offline setting as follows.

Theorem 80. *Assume f is set-adaptive submodular and adaptive monotone with respect to p. Suppose an α-approximation algorithm for monotone submodular maximization subject to $X \in \mathcal{J}$ is used for the batch selection at each round. If π is a policy that encodes the batch-mode adaptive greedy algorithm in the online setting and π^* is an optimal batch-mode policy in the offline setting, then we have*

$$f_{\mathrm{avg}}(\pi_{\mathrm{g}}) \geq \frac{\alpha}{1+\alpha} f_{\mathrm{avg}}(\pi^*).$$

Proof. The expected objective value achieved by the adaptive greedy algorithm can be bounded as

$$\begin{aligned} f_{\mathrm{avg}}(\pi) =& \mathbb{E}[f(E(\pi, \Phi), \Phi)] \\ =& \sum_{i=1}^{k} \mathbb{E}[f(E(\pi_{[i]}, \Phi), \Phi) - f(E(\pi_{[i-1]}, \Phi), \Phi)] \\ =& \sum_{i=1}^{k} \sum_{\psi_{i-1} \in \mathcal{L}(\pi_{[i-1]})} p(\psi_{i-1}) \Delta(\pi(\psi_{i-1})|\psi_{i-1}), \end{aligned}$$

where $\pi_{[i]}$ is the policy obtained by executing π just before the $i+1$th rounds and $\mathcal{L}(\pi)$ be the set of all possible partial realizations of policy π.

We consider the concatenated policy $\pi@\pi^*$ of π and π^*, that is, a policy obtained by executing π^* as if from scratch after executing π. Since π selects a batch from $\mathcal{J}_i$ once and π^* also selects a batch from $\mathcal{J}_i$ once, the concatenated policy $\pi@\pi^*$ selects a batch from $\mathcal{J}_i$ twice for any i, first during π and second during π^*. We define $\mathcal{M}(\psi_i)$ to be the set of all partial realizations observed by $\pi@\pi^*$ since having ψ_i during executing π until just before selecting a batch from $\mathcal{J}_i$ during executing π^*. Since $\sum_{\psi\in\mathcal{M}(\psi_{i-1})} p(\psi_{i-1}\cup\psi) = p(\psi_{i-1})$, we have

$$
\begin{aligned}
p(\psi_{i-1})\Delta(\pi(\psi_{i-1})|\psi_{i-1}) &\geq \alpha p(\psi_{i-1}) \max_{S\in\mathcal{J}_i} \Delta(S|\psi_{i-1}) && \text{(From the property of the algorithm)}\\
&= \alpha \sum_{\psi\in\mathcal{M}(\psi_{i-1})} p(\psi_{i-1}\cup\psi) \max_{S\in\mathcal{J}_i} \Delta(S|\psi_{i-1})\\
&\geq \alpha \sum_{\psi\in\mathcal{M}(\psi_{i-1})} p(\psi_{i-1}\cup\psi)\Delta((\pi@\pi^*)(\psi_{i-1}\cup\psi)|\psi_{i-1})\\
& && \text{(since } (\pi@\pi^*)(\psi_{i-1}\cup\psi)\in\mathcal{J}_i\text{)}\\
&\geq \alpha \sum_{\psi\in\mathcal{M}(\psi_{i-1})} p(\psi_{i-1}\cup\psi)\Delta((\pi@\pi^*)(\psi_{i-1}\cup\psi)|\psi_{i-1}\cup\psi). \\
& && \text{(due to the adaptive submodularity)}
\end{aligned}
$$

Therefore, finally, we have

$$
\begin{aligned}
f_{\mathrm{avg}}(\pi) &= \sum_{i=1}^{k} \sum_{\psi_{i-1}\in\mathcal{L}(\pi_{[i-1]})} p(\psi_{i-1})\Delta(\pi(\psi_{i-1})|\psi_{i-1})\\
&\geq \alpha \sum_{i=1}^{k} \sum_{\psi_{i-1}\in\mathcal{L}(\pi_{[i-1]})} \sum_{\psi\in\mathcal{M}(\psi_{i-1})} p(\psi_{i-1}\cup\psi)\Delta((\pi@\pi^*)(\psi_{i-1}\cup\psi)|\psi_{i-1}\cup\psi)\\
&= \alpha(f_{\mathrm{avg}}(\pi@\pi^*) - f_{\mathrm{avg}}(\pi)),
\end{aligned}
$$

which concludes

$$
f_{\mathrm{avg}}(\pi) \geq \frac{\alpha}{1+\alpha} f_{\mathrm{avg}}(\pi@\pi^*) \geq \frac{\alpha}{1+\alpha} f_{\mathrm{avg}}(\pi^*),
$$

where the second inequality is due to Lemma 32. □

By using existing approximation algorithms for constrained monotone submodular maximization, we obtain the following results.

Corollary 81. *If $X\in\mathcal{J}$ is a matroid constraint, the batch-mode adaptive greedy with the continuous greedy algorithm [Călinescu et al., 2011] achieves $(\mathrm{e}-1)/(2\mathrm{e}-1)$-approximation. If $X\in\mathcal{J}$ is a knapsack constraint, the batch-mode adaptive greedy with the greedy algorithm with partial enumeration [Sviridenko, 2004] achieves $(\mathrm{e}-1)/(2\mathrm{e}-1)$-approximation. If $X\in\mathcal{J}$ is a p-system constraint, the batch-mode adaptive greedy with the greedy algorithm [Călinescu et al., 2011] achieves $(p+1)/(p+2)$-approximation.*

4.7.3 Query-Varying Setting

Here we consider the batch-mode adaptive optimization problem where the batch constraint changes at each round, which we call the *query-varying setting*. In this setting, the ground set V is the same through all rounds but the batch constraints are different at each round. Let $\mathcal{J}_i$ be the constraints at the ith round

Algorithm 10 Batch-mode adaptive greedy algorithm for the query-varying setting

Input The objective function $f\colon 2^V \times \mathcal{Y}^V$ and the probability distribution $p \in \triangle^{\mathcal{Y}^V}$ given by a value oracle for the expected marginal gain $\Delta(\cdot|\cdot)$, the number of rounds $k \in \mathbb{Z}_{\geq 0}$.

$\psi_0 \leftarrow \varnothing$.
for $i = 1, \cdots, k$ **do**
  Given an independence oracle for $\mathcal{J}_i$,
  Apply an α-approximation algorithm to maximize $\Delta(S|\psi_{i-1})$ subject to $S \in \mathcal{J}_i$ and obtain an α-approximate solution S_i.
  Query S_i and observe $\phi(v)$ for all $v \in S_i$.
  $\psi_i \leftarrow \psi_{i-1} \cup \{(v, \phi(v)) \mid v \in S_i\}$.

for each i. At the ith round, given $\mathcal{J}_i$, the decision-maker must select a batch $S_i \in \mathcal{J}_i$ without knowing the batch constraints for the future rounds $\mathcal{J}_{i+1}, \cdots, \mathcal{J}_k$. While Fern et al. [2017] dealt with the query-varying setting under the assumption that $\mathcal{J}_1, \cdots, \mathcal{J}_k$ are generated from an identical distribution independently, we does not assume any probabilistic assumption on $\mathcal{J}_1, \cdots, \mathcal{J}_k$. We assume that $\mathcal{J}_1, \cdots, \mathcal{J}_k$ are determined in advance and do not change adaptively to the decision-maker's selection of batches.

We apply the batch-mode adaptive greedy algorithm to the query-varying setting. At each round, this algorithm selects a batch that approximately maximizes the expected marginal gain, that is, $\Delta(S_i|\psi_{i-1}) \geq \alpha \max_{S \in \mathcal{J}_i} \Delta(S|\psi_{i-1})$. The algorithmic description is given in Algorithm 10.

In comparison to an optimal batch-mode policy for the query constraints $\mathcal{J}_1, \cdots, \mathcal{J}_k$, the batch-mode adaptive greedy algorithm always gives a competitive performance. We can prove it by reducing the query-varying setting to the online setting.

Theorem 82. *Assume f is set-adaptive submodular and adaptive monotone with respect to p. Suppose an α-approximation algorithm for monotone submodular maximization subject to $X \in \mathcal{J}$ is used for the batch selection at each round. If π is a policy that encodes the batch-mode adaptive greedy algorithm and π^* is an optimal batch-mode policy in the query-varying setting, then we have*

$$f_{\mathrm{avg}}(\pi_{\mathrm{g}}) \geq \frac{\alpha}{1+\alpha} f_{\mathrm{avg}}(\pi^*).$$

Proof. We reduce the query-varying setting to the online setting by making k copies of each element $v \in V$. Let $\tilde{V} = \{(v, i) \mid v \in V,\ i \in [k]\}$ be the new ground set obtained by regarding a pair (v, i) of element $v \in V$ and index of the round $i \in [k]$ as an element. For any realization ϕ, the corresponding realization is defined to be $\tilde{\phi}\colon \tilde{V} \to \mathcal{Y}$ such that $\tilde{\phi}((v, i)) = \phi(v)$. We say $\tilde{\phi}$ is valid if there exists the corresponding realization ϕ. We define the new objective function $\tilde{f}\colon 2^{\tilde{V}} \times \mathcal{Y}^{\tilde{V}} \to \mathbb{R}_{\geq 0}$ such that

$$\tilde{f}(\tilde{S}, \tilde{\phi}) \coloneqq \begin{cases} f(S, \phi) & \text{if } \phi \text{ is valid} \\ 0 & \text{if } \phi \text{ is not valid} \end{cases}$$

for any $\tilde{S} \subseteq \tilde{V}$ and any $\tilde{\phi}$, where $S = \{v \in V \mid \exists i \in [k],\ (v, i) \in \tilde{S}\}$ is the corresponding to $\tilde{S}$ and ϕ is the realization corresponding to $\tilde{\phi}$. We define the probability distribution $\tilde{p}$ over $\tilde{\phi}$ such that $\tilde{p}(\tilde{\phi}) = p(\phi)$ if $\tilde{\phi}$ is valid and $\tilde{p}(\tilde{\phi}) = 0$ if $\tilde{\phi}$ is not valid.

First, we show the set-adaptive submodularity of $\tilde{f}$ with respect to $\tilde{p}$. Let $\tilde{\Delta}$ denote the expected marginal gain about $\tilde{f}$ and $\tilde{p}$. Fix any $\tilde{S} \subseteq \tilde{V}$ and partial realization $\tilde{\psi}$ and $\tilde{\psi}'$ such that $\tilde{\psi} \subseteq \tilde{\psi}'$. We can define the corresponding partial realization $\psi \coloneqq \{(v, y) \mid \exists((v, i), y) \in \tilde{\psi}\}$ and $\psi' \coloneqq \{(v, y) \mid \exists((v, i), y) \in \tilde{\psi}'\}$. We can show the set-adaptive submodularity of $\tilde{f}$ with respect to $\tilde{p}$ as

$$\tilde{\Delta}(\tilde{S}|\tilde{\psi}) = \Delta(S|\psi) \geq \Delta(S|\psi') = \tilde{\Delta}(\tilde{S}|\tilde{\psi}')$$

where the inequality is due to the set-adaptive submodularity of f with respect to p. Similarly, we show the adaptive monotonicity of $\tilde{f}$ with respect to $\tilde{p}$ as

$$\tilde{\Delta}((v,i)|\tilde{\psi}) = \Delta(v|\psi) \geq 0,$$

which is derived from the adaptive monotonicity of f with respect to p.

Finally, by considering $\tilde{\mathcal{J}}_i = \{\tilde{S} \subseteq \mathcal{V} \mid \exists S \in \mathcal{J}_i,\ \tilde{S} = \{(v,i) \mid v \in S\}\}$ as the batch constraint at the ith round, the query-varying setting can be reduced to the online setting. From Theorem 80, we obtain the statement of the theorem. □

Similarly to the online setting, we obtain the following results.

Corollary 83. *If $X \in \mathcal{J}$ is a matroid constraint, the batch-mode adaptive greedy with the continuous greedy algorithm [Călinescu et al., 2011] achieves $(\mathrm{e}-1)/(2\mathrm{e}-1)$-approximation. If $X \in \mathcal{J}$ is a knapsack constraint, the batch-mode adaptive greedy with the greedy algorithm with partial enumeration [Sviridenko, 2004] achieves $(\mathrm{e}-1)/(2\mathrm{e}-1)$-approximation. If $X \in \mathcal{J}$ is a p-system constraint, the batch-mode adaptive greedy with the greedy algorithm [Călinescu et al., 2011] achieves $(p+1)/(p+2)$-approximation.*

4.8 Experiments

In this section, we show the experimental results on adaptive submodular maximization with structured queries on several applications. First, we show results on two applications that satisfy set-adaptive submodularity: active learning and adaptive influence maximization in the independent cascade model. Next, we move to two applications where the adaptive submodularity ratio is bounded: bipartite influence maximization in the triggering model and adaptive feature selection.

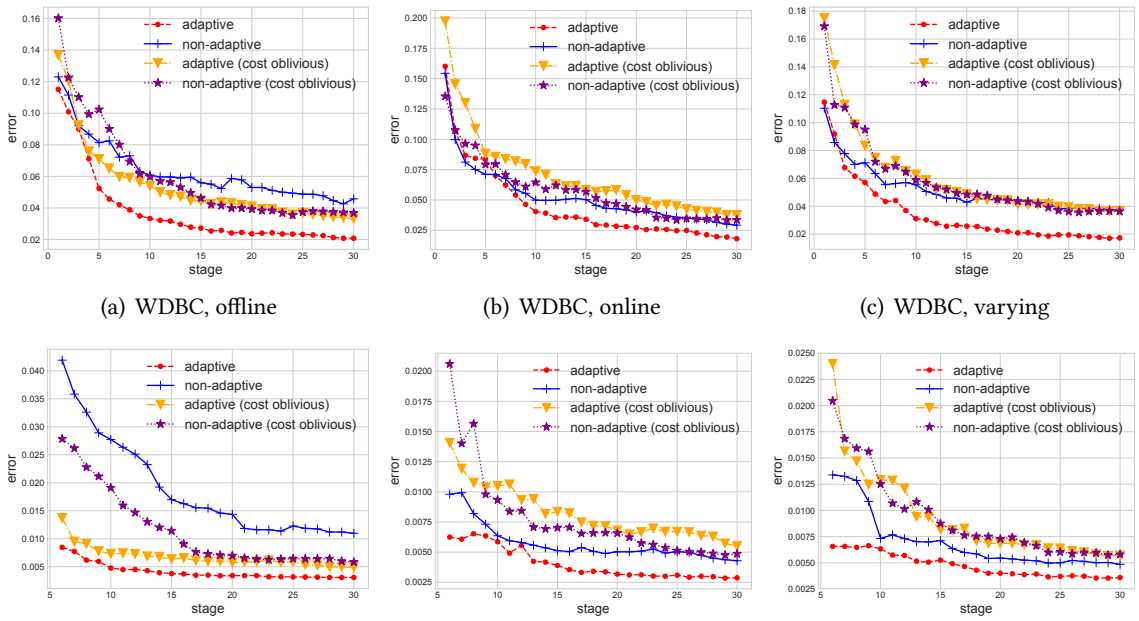
4.8.1 Experiments on Active Learning

We conduct experiments in the offline, online, and query-varying settings of active learning. In all settings, as batch constraints, we impose knapsack constraints. The weight of each item is generated from the uniform distribution on $[0, 1]$ and the capacity of the knapsack is set to 1.

Datasets. We use benchmark datasets, WDBC[1] and MNIST[2]. WDBC is a dataset of 569 cells with 32-dimensional feature and their diagnosis results. From MNIST, a dataset of handwritten digits, we extract images corresponding to digit 0 and 1. We apply PCA to reduce the dimensions 784 to 10, and consider the binary classification problem. Each dataset is normalized so that their mean is 0 and variance is 1. For fair comparison, the performance is measured only with the combination of the selected examples, that is, we train linear SVM with the obtained labels and calculating the test accuracy with this linear separator with the whole dataset.

Methods. Our implementation is based on ALuMA algorithm [Gonen et al., 2013] that we select among many active learning algorithms based on adaptive submodularity. ALuMA algorithm samples hypotheses consistent with observed labels and select the next example to be labeled that minimizes the number of consistent hypotheses in expectation. The original ALuMA algorithm is designed for linear separable datasets, then we adopt the modification proposed by Chen and Krause [2013], which makes it possible to deal with noisy datasets. We set the noise tolerance parameter ϵ to 0.1 and the number of hypotheses sampled at each step to 1000. As a benchmark, we implement the non-adaptive algorithm

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
[2] http://yann.lecun.com/exdb/mnist/



(a) WDBC, offline (b) WDBC, online (c) WDBC, varying

(d) MNIST, offline (e) MNIST, online (f) MNIST, varying

Figure 4.3: The experimental results for batch-mode active learning with knapsack batch constraints. In all figures, the horizontal axis indicates the number of rounds and the vertical axis indicates the test error obtained by linear SVM trained with the observed labels. (a), (b), and (c) are the results for WDBC dataset. (d), (e), and (f) are the results for MNIST dataset. (a) and (d) are the results for the offline setting, (b) and (e) are the results for the online setting, and (c) and (f) are the results for the query-varying setting.

that selects a batch S_i at each round by maximizing $\Delta(\mathrm{dom}(\psi_{i-1}) \cup S_i|\varnothing) - \Delta(\mathrm{dom}(\psi_{i-1})|\varnothing)$, not $\Delta(S_i|\psi_{i-1})$.

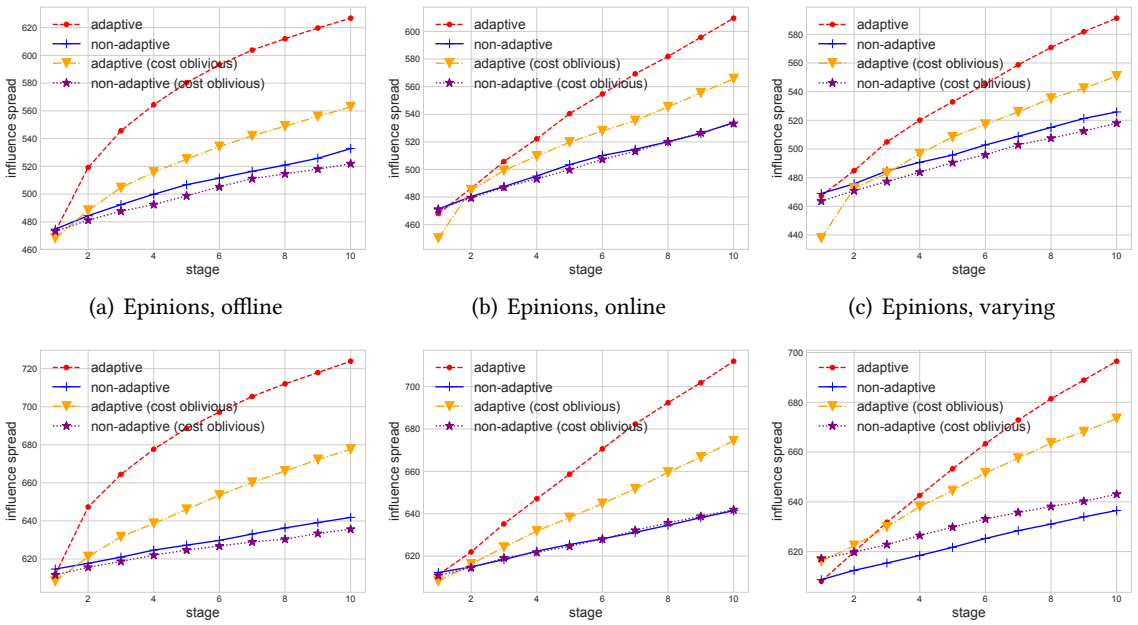
As a subroutine for the batch selection problem, we adopt the cost-effective forward selection algorithm though the best approximation ratio is $(1 - 1/\mathrm{e})$ by the greedy algorithm with partial enumeration. The known bound on the approximation ratio of the cost-effective forward selection algorithm is $\frac{1}{2}(1 - 1/\mathrm{e})$, but it runs in $\mathrm{O}(n^2)$ time [Leskovec et al., 2007a], which is much faster than $\mathrm{O}(n^5)$ time of the greedy algorithm with partial enumeration. In addition, we compare it with a different subroutine, the cost-oblivious greedy algorithm, which greedily selects an item that provides the largest improvement of the objective function and stops when adding any remaining item violates the knapsack constraint. We implement the batch-mode adaptive greedy algorithm and the non-adaptive algorithm both with these two subroutines.

Results. For each setting and dataset, we repeat experiments 20 times in the same setting and plot the average of the test accuracy. In all settings and datasets, the batch-mode adaptive greedy algorithm with the cost-effective forward selection outperforms the other methods.

4.8.2 Experiments on Adaptive Influence Maximization in the IC model

Datasets. We use datasets soc-Epinions1 and soc-Slashdot0801 from Stanford Large Network Dataset Collection (SNAP)[3]. The original network of soc-Epinions1 has 75879 nodes and 508377 edges, and

[3] https://snap.stanford.edu/



(a) Epinions, offline (b) Epinions, online (c) Epinions, varying

(d) Slashdot, offline (e) Slashdot, online (f) Slashdot, varying

Figure 4.4: The experimental results for batch-mode influence maximization with knapsack batch constraints. In all figures, the horizontal axis indicates the number of rounds and the vertical axis indicates the number of nodes influenced by the selected seed nodes. (a), (b), and (c) are the results for Epinions dataset. (d), (e), and (f) are the results for Slashdot dataset. (a) and (d) are the results for the offline setting, (b) and (e) are the results for the online setting, and (c) and (f) are the results for the query-varying setting.

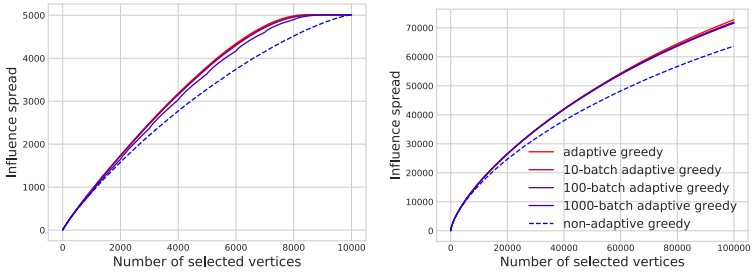
The one of soc-Slashdot0801 has 77360 nodes and 905468 edges. To scale down the problem size, we consider the subgraph induced by the top 1000 nodes that have the largest outdegree. We use the independent cascade model [Kempe et al., 2003] as the diffusion model and the full-adoption feedback model [Golovin and Krause, 2011a] as the feedback model. We set the probability that each edge is activated is set to 0.03.

Methods. We compare two adaptive methods and two non-adaptive methods similarly to the experiments on active learning. We implement batch-mode adaptive greedy algorithm with two different subroutines: the one is cost-effective greedy algorithm and the other is cost-oblivious greedy algorithm. In the same way, we implement the non-adaptive algorithm that selects a batch at every step by cost-effective greedy algorithm or cost-oblivious greedy algorithm. In all implemented methods, we use Monte Carlo sampling for estimating the objective value.

Results. For each setting and dataset, we conduct 20 trials and plot the average of the influence spread. In all settings and datasets, the adaptive greedy algorithm with the cost-effective forward selection performs much better than the other methods.

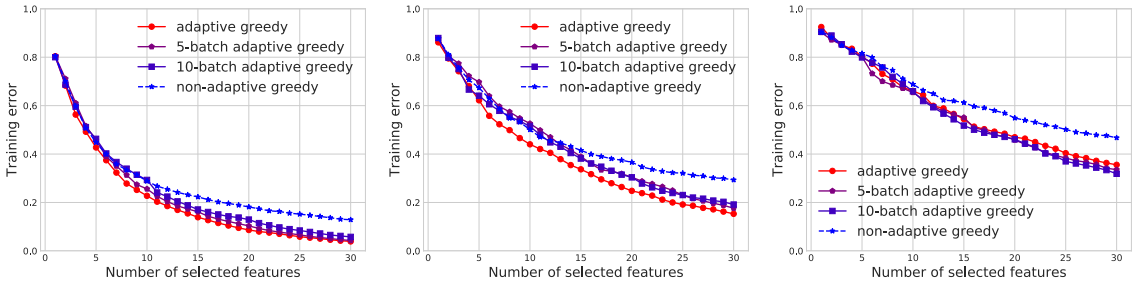
4.8.3 Experiments on Bipartite Influence Maximization in the Triggering Model

Datasets. We use bipartite graphs generated randomly and Yahoo! dataset [Yah]. The detailed description of these datasets can be found in Section 3.7.1.



(a) linear threshold, synthetic (b) linear threshold, Yahoo!

Figure 4.5: The experimental results on bipartite influence maximization with the triggering model. (a) is the result on the synthetic datasets under the linear threshold model. (b) is the result on Yahoo! dataset under the linear threshold model.



(a) regression, $\sigma = 0.1$ (b) regression, $\sigma = 0.2$ (c) regression, $\sigma = 0.3$

Figure 4.6: The experimental results on adaptive feature selection. (a), (b), and (c) are the results for noise parameter $\sigma = 0.1$, $\sigma = 0.2$, and $\sigma = 0.3$, respectively.

Benchmarks. We compare the adaptive greedy algorithm to batch-mode adaptive greedy algorithm with different batch sizes $10, 100, 1000$. Each of them selects a batch of the specified size by the non-adaptive greedy algorithm.

Results. We conduct experiments on the linear threshold model with the synthetic dataset and Yahoo! dataset, and experiments on the extended linear threshold model with the synthetic dataset. The results are shown in Figure 4.5. We can see that even if the batch size is 1000, its performance is competitive with the adaptive greedy algorithm and much better than the non-adaptive greedy algorithm.

4.8.4 Experiments on Adaptive Feature Selection

Datasets. We use synthetic datasets generated in the same way as Section 3.7.2. The noise parameter σ is set to 0.1, 0.2, and 0.3.

Benchmarks. We implement the adaptive greedy and non-adaptive greedy algorithms as well as the batch-mode adaptive greedy algorithm with batch size 5 and 10. Each algorithm selects a batch of the specified size by the non-adaptive greedy algorithm.

Results. The results are shown in Figure 4.6. In all settings, we can see the batch-mode adaptive greedy algorithms are competitive with the adaptive greedy algorithm.

4.9 Related Work

Batch-mode adaptive optimization. To our knowledge, the first study on batch-mode adaptive optimization is Chen and Krause [2013]. They analyzed the batch-mode adaptive greedy algorithm for the coverage version, but their assumptions are adaptive submodularity, pointwise submodularity, and pointwise monotonicity, which are not sufficient for their analysis as described in Section 4.4. Fern et al. [2017] first considered the query-varying setting with the same assumptions as Chen and Krause [2013]. Their problem setting assumes that the set of feasible batches is generated at each round independently from an identical distribution, which is different from our query-varying setting. Sakaue [2019] considered the multi-stage setting of monotone set function maximization, which is a non-adaptive counterpart of batch-mode adaptive optimization. They used the submodularity ratio and the supermodularity ratio in a similar way to ours, but their goal is to approximate an optimal fully-adaptive policy, not an optimal batch-mode policy.

Online submodular maximization. The problem setting of *online submodular maximization* [Streeter and Golovin, 2008, Streeter et al., 2009] is similar to our online setting in some sense. In online submodular maximization, the decision-maker selects a set under a cardinality constraint [Streeter and Golovin, 2008] or a matroid constraint [Streeter et al., 2009] to maximize a monotone submodular function. The largest difference from our online setting is that in their setting, the decision-maker must decide a set without knowing anything about the objective function. Since it is obviously impossible to devise an approximation algorithm in their setting, their goal is to minimize a criterion called α-*approximate regret*. Intuitively, the α-approximate regret represents an additive error in comparison to applying an α-approximation algorithm to the total objective functions for all rounds in hindsight. Streeter and Golovin [2008], Streeter et al. [2009] developed algorithms whose $(1 - 1/\mathrm{e})$-approximate regret can be bounded by $\mathrm{O}(\sqrt{T})$ where T is the number of rounds for a cardinality constraint or a matroid constraint, respectively. In contrast, the decision-maker knows the probability distribution that generates the objective function in our online setting. Hence, our online setting is an easier problem than online submodular maximization, and we can devise an approximation algorithm.

Adaptive submodular maximization in the bandit setting. Gabillon et al. [2013, 2014] considered the problem of maximizing an adaptive submodular maximization in the *bandit* setting. In this setting, the decision-maker repeatedly solves an adaptive submodular maximization problem, but the distribution of the states is unknown in the beginning. The decision-maker can gradually learn the distribution by selecting elements and observing their states. This setting is more difficult than our online setting, in which the distribution is known in advance, therefore their goal is to minimize the $(1-1/\mathrm{e})$-approximate regret similarly to online submodular maximization.

Adaptive submodular maximization in the stream-based setting. Fujii and Kashima [2016] considered the adaptive submodular maximization in the stream-based setting. In their setting, the decision-maker must select a subset out of the elements that arrive sequentially. While in our online setting, the elements are partitioned into small ground sets for each round, they considered the setting where the elements in the ground set arrive one by one. We can see the difference more clearly by removing the adaptive factor from the problem settings. Our online setting coincides with online submodular welfare maximization [Lehmann et al., 2006], while their setting coincides with submodular secretary problem [Bateni et al., 2013].

4.10 Summary and Future Work

In this chapter, we considered several settings of batch-mode adaptive optimization with structured queries. First we defined the notion of set-adaptive submodularity to refine the analysis by Chen and Krause [2013] and showed that this property holds in important applications. By assuming set-adaptive submodularity, we bounded the approximation ratio of the batch-mode adaptive greedy algorithm. Furthermore, by utilizing the framework of the adaptive submodularity ratio, we dealt with the case where set-adaptive submodularity does not hold. We also devised greedy algorithms for outer matroid constraints, the online setting, and the query-varying setting. Our experimental results demonstrated the efficiency of the batch-mode adaptive greedy algorithms.

A possible direction for future research is to analyze the gap between the batch-mode adaptive setting and the fully adaptive setting. In the fully adaptive setting, we can achieve a better objective value than in the batch-mode adaptive setting, but their difference is not well understood. We can define this gap in a similar way to the adaptivity gap. If we obtain a general bound on the gap, we can better understand the trade-off between efficiency and effectiveness of the batch-mode adaptive setting.

5 Local Search for Feature Selection with Structured Constraints

This chapter is organized as follows. In Section 5.1, we explain the background and overview of this chapter. Section 5.2 formulates the problem setting of feature selection with combinatorial constraints. Section 5.3 describes preliminary notations and lemmas. Section 5.4 introduces the notion of approximate submodularity for local search. In Section 5.5, we describe applications of our problem settings: sparse regression and structure learning of graphical models. In Sections 5.6 and 5.7, we propose local search algorithms for a matroid constraint and a p-matroid intersection or p-exchange system constraint, respectively. In Section 5.8, we empirically compare our proposed algorithms with existing methods. Section 5.9 provides a summary and future work of this chapter.

5.1 Background and Overview

In compressed sensing and machine learning, we are often faced with high-dimensional learning problems. A reasonable way to obtain a better solution for high-dimensional problems is to reduce the number of features by selecting an appropriate subset of the original numerous features. This problem is called *feature selection*. By appropriately reducing the number of features, we obtain a model that is more robust and more interpretable than the original high-dimensional model. Feature selection is thought to be an important combinatorial optimization problem in machine learning.

As described in Section 2.2, feature selection can be formulated as an optimization problem of finding a sparse support that maximizes a continuous function $u\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, that is,

$$\begin{aligned} &\text{Maximize} && u(\mathbf{w}) \\ &\text{subject to} && \|\mathbf{w}\|_0 \leq s, \end{aligned}$$

where $s \in \mathbb{Z}_{\geq 0}$ is the sparsity parameter. Since Natarajan [1995] proved that this problem is NP-hard even in the case of sparse linear regression, several studies have developed approximation algorithms for this problem [Das and Kempe, 2011, Elenberg et al., 2018].

Here, we focus on feature selection problems with *structured constraints*. In a realistic setting, we can improve the quality of the estimation by using prior knowledge of structures of the sparse support. Structured sparsity regularization [Huang et al., 2009, Bach et al., 2012] is a prevalent framework for learning a sparse support with various structures by incorporating such prior knowledge into the regularization term. However, there are still many structures that are not handled by the existing framework of structured sparsity regularization. For example, to the best of our knowledge, the case where feasible supports are expressed as a b-matching constraint, which allows us to select any subset of edges such that the degree of each vertex is at most a predetermined number, has not been studied in the existing framework.

To deal with such a structured constraint, we develop a novel framework for feature selection based on local search. Local search is a well-known algorithm design technique for combinatorial optimization problems, and widely used for feature selection in practice. The idea of local search is to start with some initial solution and to repeatedly update the solution until it reaches sufficient quality. Local search

performs well in many real-world situations, but in most cases, any theoretical guarantee is not known. The goal of this study is to provide a general framework for designing local search algorithms with theoretical guarantees for feature selection with structured constraints.

As we illustrated in Chapter 2, submodularity provides approximation ratio bounds for local search algorithms, but the objective function of feature selection does not always satisfy submodularity. Therefore, we propose a novel property called *approximate submodularity for local search*. We then show two applications of this property: sparse regression and structure estimation of graphical models. We devise local search algorithms for a matroid constraint and show approximation ratio guarantees based on this property. We also develop accelerated variants of the proposed local search algorithm, namely, semi-oblivious and non-oblivious variants. By using the theoretical techniques developed by Lee et al. [2010] and Feldman et al. [2011], our proposed framework can be extended to two classes of more complicated structured constraints: p-matroid intersection constraints and p-exchange system constraints. These classes contain several important constraints that cannot be handled by the framework of structured sparsity regularization such as b-matching constraints.

Our contributions are summarized as follows.

- We show that feature selection problems with a strongly concave and smooth objective function can be regarded as a problem of maximizing an approximately submodular function.
- We develop local search algorithms with approximation guarantees for this problem.
- We show how to accelerate this algorithm by borrowing an idea from orthogonal matching pursuit while keeping the approximation guarantee.
- We extend the proposed algorithm for more complicated constraints such as p-matroid intersection and p-exchange systems.

5.1.1 Related Work

Feature selection as approximate submodular maximization. As described in Section 2.2.2, Das and Kempe [2011] tackled feature selection for sparse linear regression from the viewpoint of submodularity. They proposed the notion of submodularity ratio and showed the approximation ratio of forward regression and orthogonal matching pursuit. Elenberg et al. [2018] extended their results to more general feature selection problems, and showed that the submodularity ratio can be bounded by the restricted strong concavity parameter and the restricted smoothness parameter. To our knowledge, Chen et al. [2018a] is the only result that deals with feature selection beyond cardinality constraints. They showed the random residual greedy algorithm achieves $(\gamma/(1+\gamma))^2$-approximation for a matroid constraint (They did not specify the subscripts of γ, but it is no larger than $\min_{i=1,\cdots,s} \gamma_{i-1,s-i}$). Note that the results of Das and Kempe [2011] and Chen et al. [2018a] hold for any monotone set function whose submodularity ratio is bounded, while we utilize a stronger property derived from restricted strong concavity and restricted smoothness. These results are summarized in Table 5.1.

Sparse recovery analyses of local search. Despite our interest lies in bounds on approximation ratios, most existing studies on feature selection focus on sparse recovery guarantees. In the context of sparse recovery, several algorithms similar to local search have been developed. CoSaMP [Needell and Tropp, 2010] and its generalization GraSP [Bahmani et al., 2013] are algorithms similar to our proposed non-oblivious local search, but their analysis aims to prove sparse recovery guarantees.

Learning the structure of graphical models. Learning the structure of graphical models from random samples is a fundamental problem in machine learning. For a special case where the underlying

Table 5.1: Comparison of existing bounds on approximation ratios of local search algorithms, greedy algorithms, and modular approximation. The result of Das and Kempe [2011] is indicated by †. The result of Chen et al. [2018a] is indicated by ‡.

Constraint	Local search	Greedy-based	Modular Approx.
Cardinality	$\frac{m_{2s}^2}{M_{s,2}^2} - \epsilon$	$1 - \exp\left(-\frac{m_{2s}}{M_{s,1}}\right)$ †	$\frac{m_1 m_s}{M_1 M_s}$
Matroid	$\frac{m_{2s}^2}{M_{s,2}^2} - \epsilon$	$\frac{1}{(1+\frac{M_{s,1}}{m_s})^2}$ ‡	$\frac{m_1 m_s}{M_1 M_s}$
p-Exchange system	$\frac{1}{p-1+1/q}\frac{m_{2s}^2}{M_{s,2}^2} - \epsilon$	N/A	$\frac{1}{p-1+1/q}\frac{m_1 m_s}{M_1 M_s} - \epsilon$

graph is a tree, Chow and Liu [1968] devised an efficient algorithm. Jalali et al. [2011] provided sparse recovery guarantees for the forward-backward greedy method by assuming the strong concavity of the scoring function. For the setting without the incoherence assumption, Bresler [2015] and Klivans and Meka [2017] devised algorithms with theoretical guarantees for Ising models and Markov random fields, respectively.

5.2 Problem Setting

In this section, we introduce the problem setting of feature selection with structured constraints. This problem is an extension of the sparse optimization problem introduced in Section 2.2.3.

We consider the problem of finding a sparse solution that maximizes a continuous function $u\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$. To incorporate prior knowledge of the sparsity, we consider generalized constraints on the support. Let $V = [n]$ be the set of all features and $\mathcal{I} \subseteq 2^V$ a family of feasible supports. We can write the feature selection problem as

$$\begin{aligned} &\text{Maximize} && u(\mathbf{w}) \\ &\text{subject to} && \operatorname{supp}(\mathbf{w}) \in \mathcal{I}. \end{aligned}$$

The optimization problem introduced in Section 2.2.3 is a special case for $\mathcal{I} = \{X \subseteq V : |X| \leq s\}$.

By introducing a set function $f\colon 2^V \to \mathbb{R}_{\geq 0}$ defined as

$$f(X) = \max_{\operatorname{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w}),$$

we can formulate this problem as a set function optimization problem

$$\begin{aligned} &\text{Maximize} && f(X) \\ &\text{subject to} && X \in \mathcal{I}. \end{aligned}$$

Thus, we can regard the problem of finding a sparse solution as a set function optimization problem. We consider three classes of constraints: matroid constraints, p-matroid intersection, and p-exchange systems, which are introduced in Section 2.1.

5.3 Preliminaries

Notation. Let $\Omega_s = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{y}\|_0 \leq s\}$ and $\Omega_{s,t} = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n \mid \|\mathbf{x}\|_0 \leq s,\ \|\mathbf{y}\|_0 \leq s,\ \|\mathbf{x} - \mathbf{y}\|_0 \leq t\}$. Let m_s be strongly concavity parameter on Ω_s and $M_{s,t}$-smooth on $\Omega_{s,t}$ for any positive integer $s, t \in \mathbb{Z}_{>0}$. Due to the strong concavity of u, $\operatorname{argmax}_{\mathrm{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w})$ is uniquely determined. We denote this maximizer by $\mathbf{w}^{(X)}$.

Here we provide the basic facts that are used in the proofs. First, we show the exchange property of matroids.

Lemma 84 (Corollary 39.12a in Schrijver [2003]). *Let $\mathcal{M} = (V, \mathcal{I})$ be a matroid and $I, J \in \mathcal{I}$ with $|I| = |J|$. There exists a bijection $\varphi\colon I \setminus J \to J \setminus I$ such that $I - v + \varphi(v) \in \mathcal{I}$ for all $v \in I \setminus J$.*

The following lemma is on the exchange property of p-matroid intersection, which was first used for analyzing local search algorithms for submodular maximization.

Lemma 85 ([Lee et al., 2010]). *Suppose $\mathcal{I}$ is a p-matroid intersection. Let $q \in \mathbb{Z}$ be any positive integer. For any $S, T \in \mathcal{I}$, there exists a multiset $\mathcal{P} \subseteq 2^V$ and an integer ℓ (depending on p and q) that satisfies the following conditions.*

1. *For all $P \in \mathcal{P}$, the symmetric difference is feasible, i.e., $S \triangle P \in \mathcal{I}$, and $S \triangle P$ is q-reachable from S.*
2. *Each element $v \in T \setminus S$ appears in exactly $q\ell$ sets in $\mathcal{P}$.*
3. *Each element $v \in S \setminus T$ appears in at most $(pq - q + 1)\ell$ sets in $\mathcal{P}$.*

A property similar to that for p-matroid intersection was known for p-exchange systems as follows.

Lemma 86 ([Feldman et al., 2011]; The full proof can be found in Feldman [2013]). *Suppose $\mathcal{I}$ is a p-exchange system. Let $q \in \mathbb{Z}$ be any positive integer. For any $S, T \in \mathcal{I}$, there exists a multiset $\mathcal{P} \subseteq 2^V$ and an integer ℓ (depending on p and q) that satisfies the following conditions.*

1. *For all $P \in \mathcal{P}$, the symmetric difference is feasible, i.e., $S \triangle P \in \mathcal{I}$, and $S \triangle P$ is q-reachable from S.*
2. *Each element $v \in T \setminus S$ appears in exactly $q\ell$ sets in $\mathcal{P}$.*
3. *Each element $v \in S \setminus T$ appears in at most $(pq - q + 1)\ell$ sets in $\mathcal{P}$.*

To analyze the singleton with the largest objective, which is used as an initial solution for our proposed algorithms, we use the following fact.

Lemma 87. *Let $v^* \in \operatorname{argmax}\{f(v) \mid v \in V\}$. We have $f(\{v^*\}) \geq \frac{m_s}{sM_1} f(X)$ for any $X \in \mathcal{I}$, where $s = \max\{|X| \mid X \in \mathcal{I}\}$.*

Proof. From the submodularity ratio of f, we have

$$sf(\{v^*\}) \geq \sum_{v \in X} f(\{v\}) \geq \frac{m_s}{M_1} f(X).$$

□

5.3.1 Modular Approximation

Modular approximation is a generic method for feature selection. This method maximizes a modular function that approximates the original objective function. An algorithm for maximizing a modular function over several constraints can be utilized as a subroutine. We can regard this method as a trivial benchmark for feature selection.

Proposition 88. *Suppose we have an α-approximation algorithm for maximizing a modular function under constraint $\mathcal{I}$. Then there is an $\alpha m_1 m_s/(M_1 M_s)$-approximation algorithm for maximizing the objective function of feature selection, where $s = \max\{|X| \colon X \in \mathcal{I}\}$.*

Proof. We consider a set function

$$\tilde{f}(X) = f(\varnothing) + \sum_{x \in X} f(\{x\}|\varnothing)$$

to be a modular approximation of f. By using the restricted strong concavity and restricted smoothness of u, we have

$$\frac{m_s}{M_1} f(X) \leq \tilde{f}(X) \leq \frac{M_s}{m_1} f(X) \tag{5.1}$$

for any $X \subseteq V$ with $|X| \leq s$. Let X be the output of the α-approximation algorithm applied to maximizing $\tilde{f}(X)$ subject to $X \in \mathcal{I}$. Then we have

$$\tilde{f}(X) \geq \alpha \tilde{f}(X^*)$$

where $X^* \in \operatorname{argmax}_{X \in \mathcal{I}} f(X)$. From (5.1), we have

$$f(X) \geq \frac{m_1}{M_s} \tilde{f}(X) \geq \alpha \frac{m_1}{M_s} \tilde{f}(X^*) \geq \alpha \frac{m_1 m_s}{M_1 M_s} f(X^*).$$

□

Since there exists an exact algorithm for maximizing a linear function over a matroid constraint and $(1/(p - 1 + 1/q) - \epsilon)$-approximation algorithms for a p-matroid intersection or p-exchange system constraint, we obtain the following approximation ratio bound for modular approximation.

Corollary 89. *Modular approximation is $\frac{m_1 m_s}{M_1 M_s}$-approximation for a matroid constraint and $(\frac{1}{p-1+1/q} \frac{m_1 m_s}{M_1 M_s} - \epsilon)$-approximation for a p-matroid intersection or p-exchange system constraint.*

5.4 Approximate Submodularity for Local Search

In this section, we provide a property of the objective function of feature selection, which we call *approximate submodularity for local search*. The starting point is the following property of submodular functions, which is shown in the process of analyzing local search algorithms by Lee et al. [2010].

Proposition 90 (Implicitly proved in the proof of Lee et al. [2010, Lemma 3.1])**.** *Suppose $f \colon 2^V \to \mathbb{R}$ is non-negative, monotone, and submodular, and $X, X^* \subseteq V$ are arbitrary subsets. If $\mathcal{P}$ is a collection of subsets of V such that each element in $X^* \setminus X$ appears at least k times in $\mathcal{P}$ and each element in $X^* \setminus X$ appears at most ℓ times in $\mathcal{P}$, then we have*

$$\sum_{P \in \mathcal{P}} \{f(X \triangle P) - f(X)\} \geq k f(X^*) - (k + \ell) f(X).$$

Intuitively, this property represents that an exchange of a small number of elements increases the objective function significantly. This property plays an important role in the analyses of local search algorithms in Lee et al. [2010] and Feldman et al. [2011].

The objective function of feature selection does not satisfy this property in general, but satisfies an approximate version of this property. In this dissertation, we call this approximate version *approximate submodularity for local search*. It can be expressed by using restricted strong concavity and restricted smoothness constants as follows.

Proposition 91. *Suppose $f\colon 2^V \to \mathbb{R}$ is a set function defined as $f(X) = \max_{\mathrm{supp}(\mathbf{w}) \subseteq X} u(\mathbf{w})$, and $X, X^* \subseteq V$ are arbitrary subsets. If $\mathcal{P}$ is a collection of subsets of V such that each element in $X^* \setminus X$ appears at least k times in $\mathcal{P}$ and each element in $X^* \setminus X$ appears at most ℓ times in $\mathcal{P}$, then we have*

$$\sum_{P \in \mathcal{P}} \{f(X \triangle P) - f(X)\} \geq \frac{m_{s+s^*}}{M_{s,t}} k f(X^*) - \frac{M_{s,t}}{m_{s+s^*}} \ell f(X),$$

where $s = |X|$, $s^* = |X^*|$, and $t = \max_{P \in \mathcal{P}} |P|$.

This proposition can be proved by using the following two lemmas.

Lemma 92. *For any $X, X' \subseteq V$ with $s = \max\{|X|, |X'|\}$ and $t = |X \triangle X'|$, we have*

$$f(X') - f(X) \geq \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{X' \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{X \setminus X'} \right\|^2.$$

Proof. From the restricted smoothness of u, for any $\mathbf{z} \in \mathbb{R}^n$ with $\mathrm{supp}(\mathbf{z}) \subseteq X' \setminus X$, we have

$$\begin{aligned} f(X') - f(X) &= u(\mathbf{w}^{(X')}) - u(\mathbf{w}^{(X)}) \\ &\geq u((\mathbf{w}^{(X)})_{X \cap X'} + \mathbf{z}) - u(\mathbf{w}^{(X)}) \\ &\geq \left\langle \nabla u(\mathbf{w}^{(X)}), \mathbf{z} - (\mathbf{w}^{(X)})_{X \setminus X'} \right\rangle - \frac{M_{s,t}}{2} \left\| \mathbf{z} - (\mathbf{w}^{(X)})_{X \setminus X'} \right\|^2. \end{aligned}$$

Since this inequality holds for every $\mathbf{z}$ with $\mathrm{supp}(\mathbf{z}) \subseteq X' \setminus X$, by optimizing it for $\mathbf{z}$, we obtain

$$f(X') - f(X) \geq \frac{1}{2M_{s,t}} \left\| \nabla u(\mathbf{w}^{(X)})_{X' \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| (\mathbf{w}^{(X)})_{X \setminus X'} \right\|^2.$$

□

Lemma 93. *For any $X, X' \subseteq V$ with $s = |X|$ and $s^* = |X^*|$, we have*

$$f(X^*) - f(X) \leq \frac{1}{2m_{s+s^*}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{X^* \setminus X} \right\|^2 - \frac{m_{s+s^*}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{X \setminus X^*} \right\|^2.$$

Proof. From the restricted strong concavity of u, we obtain

$$\begin{aligned} f(X^*) - f(X) &= u(\mathbf{w}^{(X^*)}) - u(\mathbf{w}^{(X)}) \\ &\leq \left\langle \nabla u(\mathbf{w}^{(X)}), \mathbf{w}^{(X^*)} - \mathbf{w}^{(X)} \right\rangle - \frac{m_{s+s^*}}{2} \left\| \mathbf{w}^{(X^*)} - \mathbf{w}^{(X)} \right\|^2 \\ &\leq \max_{\mathbf{z}\colon \mathrm{supp}(\mathbf{z}) \subseteq X^*} \left\{ \left\langle \nabla u(\mathbf{w}^{(X)}), \mathbf{z} - \mathbf{w}^{(X)} \right\rangle - \frac{m_{s+s^*}}{2} \left\| \mathbf{z} - \mathbf{w}^{(X)} \right\|^2 \right\} \\ &= \frac{1}{2m_{s+s^*}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{X^* \setminus X} \right\|^2 - \frac{m_{s+s^*}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{X \setminus X^*} \right\|^2. \end{aligned}$$

□

From these two lemmas, we can show the proof of Proposition 91.

Proof of Proposition 91. From Lemma 92, we have

$$f(X \triangle P) - f(X) \geq \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{P \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{P \cap X} \right\|^2$$

for all $P \in \mathcal{P}$. By adding this inequality for each $P \in \mathcal{P}$, we obtain

$$\sum_{P \in \mathcal{P}} \{f(X \triangle P) - f(X)\} \geq k \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{X^* \setminus X} \right\|^2 - \ell \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{X \setminus X^*} \right\|^2,$$

where we used the fact that each element in $v \in X^* \setminus X$ appears in at least k sets in $\mathcal{P}$ and each element in $X \setminus X^*$ appears in at most ℓ sets in $\mathcal{P}$. From the strong concavity of u, by applying Lemma 93, we obtain

$$f(X^*) - f(X) \leq \frac{1}{2m_{s+s^*}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{X^* \setminus X} \right\|^2 - \frac{m_{s+s^*}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{X \setminus X^*} \right\|^2$$

and

$$\begin{aligned} -f(X) &\leq f(\varnothing) - f(X) \\ &\leq -\frac{m_{s+s^*}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_X \right\|^2 \\ &\leq -\frac{m_{s+s^*}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{X \setminus X^*} \right\|^2. \end{aligned}$$

By combining these inequalities, we have

$$\sum_{P \in \mathcal{P}} f(X \triangle P) - f(X) \geq \frac{m_{s+s^*}}{M_{s,t}} k f(X^*) - \frac{M_{s,t}}{m_{s+s^*}} \ell f(X).$$

□

In the following sections, we analyze the approximation ratio of local search algorithms by utilizing this property.

Remark 94. Though we focus on the objective function of feature selection for simplicity, we can provide similar approximation guarantees for other objective functions that satisfy the inequality in Proposition 91 with different constants.

5.5 Applications

5.5.1 Sparse Regression

Our framework can be applied to sparse regression with constraints that are more general than cardinality constraints. As described in Sections 2.2.2 and 2.2.3, sparse regression with cardinality constraints were studied by Das and Kempe [2011] and Elenberg et al. [2018]. Here, we want to consider more involved settings of sparse regression. Suppose the features are partitioned into several categories, and we should select almost the equal number of features from each category. This constraint is a special case of matroid constraints, thus our framework can be applied. As mentioned in Chen et al. [2018a], the problem of detecting splice sites in precursor messenger RNAs can be formulated as a matroid constraint as well. If there are multiple matroid constraints, we can formulate them as a p-matroid intersection constraint. To our knowledge, our proposed algorithms are the first to deal with multiple matroid constraints.

5.5.2 Structure Learning of Graphical Models

We consider the problem of estimating the graph structure of undirected graphical models, or Markov random fields, given independent and identically distributed samples from this MRF. A graphical model is an undirected graph $G = (\mathcal{V}, E)$ that represents the independence among random variables $\mathbf{X} = \{X_j \mid j \in \mathcal{V}\}$ indexed by $\mathcal{V}$. More precisely, the graphical model implies the local Markov property, that is, for each $i \in \mathcal{V}$, X_i and $\{X_j \mid j \in \mathcal{V} \setminus (N(i) \cup \{i\})\}$ are conditionally independent given $N(i)$, where $N(i)$ is the set of all adjacent vertices of i. In particular, we focus on Ising models, in which the joint probability distribution of the random variables can be expressed as the product of distributions for each edge

$$\begin{aligned} p(\mathbf{x}|\mathbf{w}) &= \frac{1}{Z(\mathbf{w})} \prod_{(i,j)\in E} \exp\left(w_{ij} x_i x_j\right) \prod_{i\in\mathcal{V}} \exp(w_i x_i) \\ &= \frac{1}{Z(\mathbf{w})} \exp\left(\sum_{(i,j)\in E} w_{ij} x_i x_j + \sum_{i\in\mathcal{V}} w_i x_i \right), \end{aligned}$$

where $Z(\mathbf{w}) = \sum_{\mathbf{x}\in\{0,1\}^{\mathcal{V}}} \exp\left(\sum_{(i,j)\in E} w_{ij} x_i x_j\right)$ is the normalization constant and $\mathbf{w}$ is the parameter vector for edges and vertices. We want to consider the problem of inferring the true parameter $\mathbf{w}$ from samples $\{\mathbf{x}^1, \cdots, \mathbf{x}^N\}$ generated by the distribution $p(\mathbf{x}|\mathbf{w})$. In general, maximizing the likelihood requires the computation of the value of $Z(\mathbf{w})$, and it is intractable. Besag [1975] proposed an approximate version of the likelihood function, called *pseudo likelihood*, which is defined as

$$\begin{aligned} u_{\mathrm{PL}}(\mathbf{w}) &= \frac{1}{N} \sum_{t=1}^{N} \sum_{j\in\mathcal{V}} \log p(x_j^t | (x_i^t)_{i\in\mathcal{V}\setminus\{j\}}, \mathbf{w}) \\ &= -\frac{1}{N} \sum_{t=1}^{N} \sum_{j\in\mathcal{V}} \log\left(1 + \exp\left(-2x_j^t \sum_{i\in\mathcal{V}\setminus\{j\}} w_{ij} x_i^t - 2w_j x_j^t\right)\right). \end{aligned}$$

In contrast to the original likelihood, the pseudo likelihood is easily computable. Since the pseudo likelihood is concave, we can apply convex optimization method to obtain the optimal parameter.

To obtain a more interpretable and robust solution, we often assume the sparsity of the edges of the true graphical model. Here, we focus on the setting where we have a rough estimate of an upper bound on the degree of each vertex. This sparsity constraint can be formulated as the degree constraints on each vertex, i.e., a b-matching constraint. By applying our proposed methods to maximizing the normalized objective function $u_{\mathrm{PL}}(\mathbf{w}) - u_{\mathrm{PL}}(\mathbf{0})$ under a b-matching constraint on the support of $\mathbf{w}$, we can estimate the set of edges in the whole graph simultaneously, while the existing method by Jalali et al. [2011] estimates the set of edges incident to each vertex separately.

5.6 Algorithms for a Matroid Constraint

In this section, we describe the proposed algorithms for a matroid constraint. The algorithm starts with an initial solution, which is any base of the given matroid. The main procedure of the algorithm is to improve the solution again and again by replacing an element in the solution with a new element. At each iteration, the algorithm seeks a pair of an element $x \in X$ and another element $x' \in V \setminus X$ that maximize $f(X - x + x')$.

We consider two other variants of the above algorithm with different criteria. Since the original one described above uses the objective value itself for judging quality of pair (x, x'), it can be called

Algorithm 11 Local search algorithms for a matroid constraint

Let $X \leftarrow \varnothing$.
Add arbitrary elements to X until X is maximal in $\mathcal{I}$.
for $i = 1, \cdots, T$ **do**
  Determine the pair of $x \in X$ and $x' \in V \setminus X$ by the following rules:

$$\begin{cases} (x, x') \in \operatorname{argmax}\{f(X - x + x') \mid X - x + x' \in \mathcal{I}\} & \text{(oblivious)} \\ \text{Let } x' \in \operatorname{argmax}\{f(X - \phi_X(x') + x') - f(X)\} \text{ and } x = \phi_X(x'), & \\ \quad \text{where } \phi_X \colon V \setminus X \to X \text{ is a map that satisfies} & \\ \quad \phi_X(x') \in \operatorname{argmin}_{x \in X \colon X - x + x' \in \mathcal{I}} (\mathbf{w}^{(X)})^2_x & \text{(semi-oblivious)} \\ (x, x') \in \underset{(x,x') \colon X - x + x'}{\operatorname{argmax}} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)}) \right)^2_{x'} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)} \right)^2_x \right\} & \text{(non-oblivious)} \end{cases}$$

  if $\begin{cases} f(X - x + x') - f(X) > 0 & \text{(oblivious or semi-oblivious)} \\ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)}) \right)^2_{x'} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)} \right)^2_x > 0 & \text{(non-oblivious)} \end{cases}$ **then**
    Update the solution $X \leftarrow X - x + x'$.
  else
    return X.
return X.

oblivious. The oblivious version computes the value of $f(X - x + x')$ for $\mathrm{O}(sn)$ pairs of (x, x') at each step. We can reduce the computational cost by utilizing the structure of set function f. The first variant can be called *semi-oblivious*. For each element $x' \in V \setminus X$ to be added, the semi-oblivious version computes the value of $f(X - x + x')$ only for $x \in X$ with the smallest $(\mathbf{w}^{(X)})^2_x$ among those satisfying $X - x + x' \in \mathcal{I}$. Thus, we can reduce the number of computing the value of $f(X - x + x')$ from $\mathrm{O}(sn)$ to $\mathrm{O}(n)$.

The second variant can be called *non-oblivious*. The non-oblivious version uses the value of

$$\frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)}) \right)^2_{x'} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)} \right)^2_x$$

in place of the increase of the objective function $f(X - x + x') - f(X)$. We need to evaluate $\nabla u(\mathbf{w}^{(X)})$ and $\mathbf{w}^{(X)}$ at the beginning of each iteration, but need not compute the value of $f(X - x + x')$. The detailed description of these algorithms are given in Algorithm 11.

We can provide the same approximation ratio bound for all these algorithms as follows.

Theorem 95. *Suppose $\mathcal{I}$ is the independence set family of a matroid. If X is the solution obtained by executing T iterations of Algorithm 11 and X^* is an optimal solution, then we have*

$$f(X) \geq \frac{m^2_{2s}}{M^2_{s,2}} \left(1 - \exp\left(-\frac{M_{s,2} T}{s m_{2s}} \right) \right) f(X^*),$$

where $s = \max\{|X| \colon X \in \mathcal{I}\}$ is the rank of the matroid. If X is the output returned by Algorithm 11 when it stops by finding no pair to improve the solution, then we have

$$f(X) \geq \frac{m^2_{2s}}{M^2_{s,2}} f(X^*).$$

Proof. Let X be the output of the algorithm and X^* an optimal solution. Suppose at some iteration, the solution is updated from X to $X - x + x'$. From Lemma 84, we have a bijection $\phi \colon X^* \setminus X \to X \setminus X^*$

such that $X - \phi(x^*) + x^* \in \mathcal{I}$ for all $x^* \in X^* \setminus X$. Here we show that

$$f(X - x + x') - f(X) \geq \frac{1}{n}\frac{M_{s,2}}{m_{2s}}\left\{\frac{m_{2s}^2}{M_{s,2}^2}f(X^*) - f(X)\right\}$$

holds at each iteration of all three variants.

When using the oblivious variant, we have

$$\begin{aligned}
&f(X - x + x') - f(X) \\
&= \max_{(x,x'):\, X-x+x' \in \mathcal{I}} f(X - x + x') - f(X) \\
&\geq \frac{1}{s}\sum_{x^* \in X^* \setminus X} \{f(X - \phi(x^*) + x^*) - f(X)\}.
\end{aligned}$$

By setting $\mathcal{P} = \{x^*, \phi(x^*)\}$, each element in $X^* \setminus X$ and $X \setminus X^*$ appears exactly once in $\mathcal{P}$. Thus, we can apply Proposition 91, and obtain

$$\sum_{x^* \in X^* \setminus X} \{f(X - \phi(x^*) + x^*) - f(X)\} \geq \frac{m_{2s}}{M_{s,2}}f(X^*) - \frac{M_{s,2}}{m_{2s}}f(X).$$

By combining these inequalities, we have

$$f(X - x + x') - f(X) \geq \frac{1}{s}\left\{\frac{m_{2s}}{M_{s,2}}f(X^*) - \frac{M_{s,2}}{m_{2s}}f(X)\right\}.$$

When using the semi-oblivious variant, due to the property of the algorithm, we have

$$(\mathbf{w}^{(X)})_{\tilde{x}}^2 \geq (\mathbf{w}^{(X)})_x^2$$

for any $\tilde{x} \in X$ such that $X - \tilde{x} + x' \in \mathcal{I}$. If $\phi_X \colon V \setminus X \to X$ is a map defined as $\phi_X(x') \in \operatorname{argmin}_{x \in X}\{(\mathbf{w}^{(X)})_x^2 \mid X - x + x' \in \mathcal{I}\}$, then

$$\begin{aligned}
&f(X - x + x') - f(X) \\
&= \max_{x' \in V \setminus X} f(X - \phi_X(x') + x') - f(X) \\
&\geq \frac{1}{s}\sum_{x^* \in X^* \setminus X} \{f(X - \phi_X(x^*) + x^*) - f(X)\} \\
&\geq \frac{1}{s}\sum_{x^* \in X^* \setminus X} \left\{\frac{1}{2M_{s,2}}\left(\nabla u(\mathbf{w}^{(X)})\right)_{x^*}^2 - \frac{M_{s,2}}{2}\left(\mathbf{w}^{(X)}\right)_{\phi_X(x^*)}^2\right\} && \text{(From Lemma 92)} \\
&\geq \frac{1}{s}\sum_{x^* \in X^* \setminus X} \left\{\frac{1}{2M_{s,2}}\left(\nabla u(\mathbf{w}^{(X)})\right)_{x^*}^2 - \frac{M_{s,2}}{2}\left(\mathbf{w}^{(X)}\right)_{\phi(x^*)}^2\right\} \\
&\qquad\qquad \text{(since } \left(\mathbf{w}^{(X)}\right)_{\phi(x^*)}^2 \geq \left(\mathbf{w}^{(X)}\right)_{\phi_X(x^*)}^2\text{)} \\
&= \frac{1}{s}\left\{\frac{1}{2M_{s,2}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{X^* \setminus X}\right\|^2 - \frac{M_{s,2}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{X \setminus X^*}\right\|^2\right\} \\
&\geq \frac{1}{s}\left\{\frac{m_{2s}}{M_{s,2}}f(X^*) - \frac{M_{s,2}}{m_{2s}}f(X)\right\}, && \text{(From Lemma 93)}
\end{aligned}$$

where we used Lemma 92 and Lemma 93 as in the oblivious case.

When using the non-oblivious variant, we have

$$
\begin{aligned}
& f(X - x + x') - f(X) \\
& \geq \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)}) \right)_{x'}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)} \right)_x^2 && \text{(From Lemma 92)} \\
& = \max_{(x,x'):\, X-x+x' \in \mathcal{I}} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)}) \right)_{x'}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)} \right)_x^2 \right\} \\
& \geq \frac{1}{s} \sum_{x^* \in X^* \setminus X} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)}) \right)_{x^*}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)} \right)_{\phi(x^*)}^2 \right\} \\
& = \frac{1}{s} \left\{ \frac{1}{2M_{s,2}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{X^* \setminus X} \right\|^2 - \frac{M_{s,2}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{X \setminus X^*} \right\|^2 \right\} \\
& \geq \frac{1}{s} \left\{ \frac{m_{2s}}{M_{s,2}} f(X^*) - \frac{M_{s,2}}{m_{2s}} f(X) \right\}. && \text{(From Lemma 93)}
\end{aligned}
$$

Therefore, in all three variants, we have

$$
\begin{aligned}
f(X - x + x') - f(X) &\geq \frac{1}{s} \left\{ \frac{m_{2s}}{M_{s,2}} f(X^*) - \frac{M_{s,2}}{m_{2s}} f(X) \right\} \\
&= \frac{1}{s} \frac{M_{s,2}}{m_{2s}} \left\{ \frac{m_{2s}^2}{M_{s,2}^2} f(X^*) - f(X) \right\},
\end{aligned}
$$

which implies that the distance from the current solution to $m_{2s}^2/M_{s,2}^2$ times the optimal value is decreased by rate $1 - M_{s,2}/(sm_{2s})$ at each iteration. Hence, the approximation ratio after T iterations can be bounded as

$$
\begin{aligned}
f(X) &\geq \frac{m_{2s}^2}{M_{s,2}^2} \left(1 - \left(1 - \frac{M_{s,2}}{sm_{2s}} \right)^T \right) f(X^*) \\
&\geq \frac{m_{2s}^2}{M_{s,2}^2} \left(1 - \exp\left(-\frac{M_{s,2} T}{sm_{2s}} \right) \right) f(X^*),
\end{aligned}
$$

which proves the first statement of the theorem.

Next, we consider the case where the algorithm stops by finding no pair to improve the objective value. For all three variants, we show that

$$
0 \geq \frac{m_{2s}}{M_{s,2}} f(X^*) - \frac{M_{s,2}}{m_{2s}} f(X)
$$

holds when the algorithm stops, from which the second statement of the theorem follows. When the oblivious variant stops, we have $f(X) \geq f(X - x + x')$ for all $x \in X$ and $x' \in V \setminus X$ such that $X - x + x' \in \mathcal{I}$. In the same way as the above analysis, we obtain

$$
\begin{aligned}
0 &\geq \sum_{x^* \in X^* \setminus X} \{ f(X - \phi(x^*) + x^*) - f(X) \} \\
&\geq \frac{m_{2s}}{M_{s,2}} f(X^*) - \frac{M_{s,2}}{m_{2s}} f(X).
\end{aligned}
$$

Similarly, when the semi-oblivious variant stops, we have $f(X) \geq f(X - \phi_X(x') + x')$ for all $x' \in V \setminus X$, where $\phi_X(x')$ is defined in the algorithm. Hence, in the same way as the above analysis, we obtain

$$\begin{aligned} 0 &\geq \sum_{x^* \in X^* \setminus X} \{f(X - \phi_X(x^*) + x^*) - f(X)\} \\ &\geq \frac{m_{2s}}{M_{s,2}} f(X^*) - \frac{M_{s,2}}{m_{2s}} f(X). \end{aligned}$$

When the non-oblivious variant stops, we have

$$0 \geq \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)_{x'}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)_x^2$$

for all $x \in X$ and $x' \in V \setminus X$ such that $X - x + x' \in \mathcal{I}$. Therefore, we have

$$\begin{aligned} 0 &\geq \sum_{x^* \in X^* \setminus X} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)_{x'}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)_x^2 \right\} \\ &\geq \frac{m_{2s}}{M_{s,2}} f(X^*) - \frac{M_{s,2}}{m_{2s}} f(X) \end{aligned}$$

by using the above analysis for the first statement. □

The time complexity of each iteration depends on the time complexity of evaluating the value of f, that is, maximizing u with a fixed support. In the case of the square loss and a uniform matroid constraint, it is easy to evaluate the time complexity as follows.

Proposition 96. *If u is the square loss and $\mathcal{I} = \{X \colon |X| \leq s\}$, time complexity of each iteration of the oblivious, semi-oblivious, non-oblivious local search algorithms is $\mathrm{O}(s^2dn)$, $\mathrm{O}(sdn)$, and $\mathrm{O}(sd + n)$, respectively.*

Proof. By using a technique of rank-one updates, we can obtain SVD of $\mathbf{A}_{X-x+x'}$ based on SVD of $\mathbf{A}_X$ in $\mathrm{O}(sd)$ time. The oblivious and semi-oblivious variants need to compute the objective value $\mathrm{O}(sn)$ and $\mathrm{O}(s)$ times at each iteration. The non-oblivious variant computes $\nabla u(\mathbf{w}^{(X)})$ and $\mathbf{w}^{(X)}$ at the beginning of each iteration by using the rank-one update technique in $\mathrm{O}(sd)$ time, and finds $\operatorname{argmax}_{x' \in V \setminus X} \left(\nabla u(\mathbf{w}^{(X)})\right)_{x'}^2$ and $\operatorname{argmin}_{x \in X} \left(\mathbf{w}^{(X)}\right)_x^2$ in $\mathrm{O}(n)$ time. Therefore, time complexity of each iteration of three variants is $\mathrm{O}(s^2dn)$, $\mathrm{O}(sdn)$, and $\mathrm{O}(sd + n)$, respectively. □

5.6.1 Variants of Geometric Improvement

Here we introduce other variants of local search algorithms that use different type of criteria for finding a pair (x, x') to improve the solution. These new variants use any pair that increase some function by rate $(1 + \delta)$, while the previously introduced variants find the pair that yields the largest improvement of some function. We consider three variants, the oblivious, semi-oblivious, and non-oblivious, similarly to the previous ones. The oblivious variant searches for any pair (x, x') that increases the objective function by rate $(1 + \delta)$, that is, $f(X - x + x') \geq (1 + \delta) f(X)$. The semi-oblivious variant constructs a map $\phi_X \colon V \setminus X \to X$ that satisfies $\phi_X(x') \in \operatorname{argmin}_{x \in X \colon X - x + x' \in \mathcal{I}} (\mathbf{w}^{(X)})_x^2$ and searches for $x' \in V \setminus X$ with $f(x - \phi_X(x') + x') \geq (1 + \delta) f(X)$. The non-oblivious variant searches for any (x, x') that satisfies

$$\frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)_{x'}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)_x^2 \geq \delta f(X).$$

Algorithm 12 Local search algorithms for a matroid constraint with geometric improvement

Let $\delta \leftarrow \epsilon/n$.
Let $X \leftarrow \operatorname{argmax}\{f(v) \mid v \in V\}$.
Add arbitrary elements to X until X is maximal in $\mathcal{I}$.
loop
  Search for a pair of $x \in X$ and $x' \in V \setminus X$ such that $X - x + x' \in \mathcal{I}$ and

$$\begin{cases} f(X - x + x') \geq (1+\delta) f(X) & \text{(oblivious)} \\ f(X - \phi_X(x') + x') \geq (1+\delta) f(X) \text{ and } x = \phi_X(x'), & \\ \quad \text{where } \phi_X \colon V \setminus X \to X \text{ is a map that satisfies} & \\ \quad \phi_X(x') \in \operatorname{argmin}_{x \in X \colon X - x + x' \in \mathcal{I}} (\mathbf{w}^{(X)})_x^2 & \text{(semi-oblivious)} \\ \dfrac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)_{x'}^2 - \dfrac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)_x^2 \geq \delta f(X) & \text{(non-oblivious)} \end{cases}$$

  if $\exists (x, x')$ satisfying the above condition **then**
    Let $X \leftarrow X - x + x'$.
  else
    return X.

All variants stop when they do not find any solution that satisfies the criteria.

The detailed description of these algorithms are given in Algorithm 12.

We can provide bounds on the approximation ratio of these variants as follows.

Theorem 97. *Suppose $\mathcal{I}$ is the independence set family of a matroid. Algorithm 12 stops after at most $\mathrm{O}(\frac{n}{\epsilon} \ln(\frac{sM_1}{m_s}))$ iterations, and returns an output X that satisfies*

$$f(X) \geq \left(\frac{m_{2s}^2}{M_{s,2}^2} - \epsilon \right) f(X^*),$$

where X^* is an optimal solution and $s = \max\{|X| \colon X \in \mathcal{I}\}$ is the rank of the matroid.

Proof. Let X be the output of the algorithm. Let X^* be an optimal solution. From Lemma 84, we have a bijection $\phi \colon X^* \setminus X \to X \setminus X^*$ such that $X - \phi(x^*) + x^* \in \mathcal{I}$ for all $x^* \in X^* \setminus X$. For each of three variants, we prove

$$0 \geq \sum_{x^* \in X^* \setminus X} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)_{x^*}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)_{\phi(x^*)}^2 - \delta f(X) \right\},$$

which implies

$$\begin{aligned} 0 &\geq \frac{1}{2M_{s,2}} \left\| \left(\nabla u(\mathbf{w}^{(X)})\right)_{X^* \setminus X} \right\|^2 - \frac{M_{s,2}}{2} \left\| \left(\mathbf{w}^{(X)}\right)_{X \setminus X^*} \right\|^2 - \delta n f(X) \\ &\geq \frac{m_{2s}}{M_{s,2}} f(X^*) - \left(\frac{M_{s,2}}{m_{2s}} + \delta n \right) f(X), \end{aligned}$$

where the second inequality is due to Lemma 93. Since we set $\delta = \epsilon/n$, we obtain

$$f(X) \geq \left(\frac{m_{2s}^2}{M_{s,2}^2} - \epsilon \right) f(X).$$

In the case of the oblivious variant, since $f(X-x+x') \leq (1+\delta)f(X)$ for all $x \in X$ and $x' \in V \setminus X$, we have

$$\begin{aligned} 0 &\geq \sum_{x^* \in X^* \setminus X} \{f(X - \phi(x^*) + x^*) - (1+\delta)f(X)\} \\ &\geq \sum_{x^* \in X^* \setminus X} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{x^*} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)^2_{\phi(x^*)} - \delta f(X) \right\} \end{aligned}$$

in a similar way to the proof of Theorem 95. In the case of the semi-oblivious variant, since $f(X - \phi_X(x') + x')$ for any $x' \in V \setminus X$, we have

$$\begin{aligned} 0 &\geq \sum_{x^* \in X^* \setminus X} \{f(X - \phi_X(x^*) + x^*) - (1+\delta)f(X)\} \\ &\geq \sum_{x^* \in X^* \setminus X} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{x^*} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)^2_{\phi_X(x^*)} - \delta f(X) \right\} \\ &\geq \sum_{x^* \in X^* \setminus X} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{x^*} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)^2_{\phi(x^*)} - \delta f(X) \right\}. \end{aligned}$$

When we use the non-oblivious variant, since

$$0 \geq \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{x^*} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)^2_{\phi(x^*)} - \delta f(X)$$

for all $x^* \in X^* \setminus X$, we obtain

$$0 \geq \sum_{x^* \in X^* \setminus X} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{x^*} - \frac{M_{s,2}}{2} \left(\mathbf{w}^{(X)}\right)^2_{\phi(x^*)} - \delta f(X) \right\}.$$

Finally, we bound the number of iterations. At each step, the objective value is improved at least at a rate of $(1+\delta)$. From Lemma 87, the initial solution is $\frac{m_s}{sM_1}$-approximation. Therefore, the number of iterations is at most $\log_{1+\delta}(\frac{sM_1}{m_s}) = \mathrm{O}(\frac{n}{\epsilon}\ln(\frac{sM_1}{m_s}))$. □

To obtain the same bound by Algorithm 11, the number of iterations need to be larger than

$$T = \frac{sM_{s,2}}{m_{2s}} \log\left(\frac{m_{2s}^2}{\epsilon M_{s,2}^2}\right),$$

which can be larger than Algorithm 12 in some cases and smaller in other cases.

5.7 Algorithms for p-Matroid Intersection and p-Exchange Systems

In this section, we consider two more general constraints, p-matroid intersection and p-exchange system constraints with $p \geq 2$. The proposed algorithms for these two constraints can be described as almost the same procedure by using the different definitions of q-reachability as in Definitions 12 and 15. We denote by $\mathcal{F}_q(X)$ the set of all q-reachable sets from X with each definition of q-reachability for p-matroid intersection or p-exchange systems.

First, we must decide $q \in \mathbb{Z}_{\geq 1}$ that determines the neighborhood to be searched for each solution. When we select larger q, we search larger solution space for improvement at each step, thus we can

Algorithm 13 Local search algorithms for a p-matroid intersection or p-exchange system ($p \geq 2$)

Let $t = \begin{cases} 2p(q+1) & \text{in the case of } p\text{-matroid intersection constraints} \\ pq+1 & \text{in the case of } p\text{-exchange system constraints.} \end{cases}$
Let $X \leftarrow \varnothing$.
Add arbitrary elements to X until X is maximal in $\mathcal{I}$.
for $i = 1, \cdots, T$ **do**
  Determine X' that is q-reachable from X such that:

$$\begin{cases} X' \in \underset{X' \in \mathcal{F}_q(X)}{\operatorname{argmax}} f(X') & \text{(oblivious)} \\ \text{Let } X' \in \underset{X' \in \mathcal{F}_q(X):\, \exists S,\, X' = (X \cup S) \setminus \phi_X(S)}{\operatorname{argmax}} f(X'), \\ \quad \text{where } \phi_X : 2^V \to 2^V \text{ is a map that satisfies} \\ \quad \phi_X(S) \in \operatorname{argmin}_{T:\, (X \cup S) \setminus T \in \mathcal{F}_q(X)} \|(\mathbf{w}^{(X)})_T\|^2 & \text{(semi-oblivious)} \\ X' \in \underset{X' \in \mathcal{F}_q(X)}{\operatorname{argmax}} \left\{ \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)})\right)_{X' \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)}\right)_{X \setminus X'} \right\|^2 \right\} & \text{(non-oblivious)} \end{cases}$$

  if $\begin{cases} f(X') - f(X) > 0 & \text{(oblivious or semi-oblivious)} \\ \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)})\right)_{X' \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)}\right)_{X \setminus X'} \right\|^2 > 0 & \text{(non-oblivious)} \end{cases}$ **then**
    Update the solution $X \leftarrow X - x + x'$.
  else
    return X.
return X.

achieve a better bound on approximation ratio, while the running time becomes larger as well. The initial solution of the proposed algorithms is any feasible solution. Then the algorithms repeatedly replace the solution with a q-reachable solution that is best under a certain criterion. Similarly to the case of matroid constraints, we can develop the oblivious, semi-oblivious and non-oblivious variants. The oblivious variant selects the next solution X' that improves the objective value $f(X')$ the most. The semi-oblivious variant computes $\phi_X(S)$ that minimizes $\left\| \left(\mathbf{w}^{(X)}\right)_{\phi_X(S)} \right\|^2$ subject to $(X \cup S) \setminus \phi_X(S) \in \mathcal{F}_q(X)$ for each $S \subseteq V \setminus X$ such that $|S| \leq q$, and selects $X' = (X \cup S) \setminus \phi_X(S)$ that maximizes $f(X')$. The non-oblivious version selects the solution $X' \in \mathcal{F}_q(X)$ that maximizes

$$\frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)})\right)_{X' \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)}\right)_{X \setminus X'} \right\|^2.$$

The detailed description of these algorithms is given in Algorithm 13.

Theorem 98. *Suppose $\mathcal{I}$ is the independence set family of a p-matroid intersection or p-exchange system. Let $t = 2p(q+1)$ for the p-matroid intersection case and $t = pq+1$ for the p-exchange system case. If X is the output obtained by executing T iterations of Algorithm 13 and X^* is an optimal solution, then we have*

$$f(X) \geq \frac{1}{p - 1 + 1/q} \frac{m_{2s}^2}{M_{s,t}^2} \left(1 - \exp\left(-\frac{(p - 1 + 1/q) M_{s,t} T}{s m_{2s}}\right)\right) f(X^*),$$

where $s = \max\{|X| \colon X \in \mathcal{I}\}$. If X is the output returned by Algorithm 13 when it stops by finding no

better q-reachable solution, then we have

$$f(X) \geq \frac{1}{p-1+1/q}\frac{m_{2s}^2}{M_{s,t}^2} f(X^*).$$

Proof. Let X be the output of the algorithm and X^* an optimal solution. Suppose at some iteration, the solution is updated from X to X'. From Lemma 85 and Lemma 86 for each case, respectively, we can see that there exists a multiset $\mathcal{P} \subseteq 2^V$ and an integer ℓ that satisfies the following conditions.

1. For all $P \in \mathcal{P}$, the symmetric difference is q-reachable from X, i.e., $X \triangle P \in \mathcal{F}_q(X)$.
2. Each element $v \in X^* \setminus X$ appears in exactly $q\ell$ sets in $\mathcal{P}$.
3. Each element $v \in X \setminus X^*$ appears in at most $(pq - q + 1)\ell$ sets in $\mathcal{P}$.

Here we show that

$$f(X') - f(X) \geq \frac{1}{s}\left\{\frac{m_{2s}}{M_{s,t}} f(X^*) - (p-1+1/q)\frac{M_{s,t}}{m_{2s}} f(X)\right\}.$$

holds at each iteration of all three variants.

When using the oblivious variant, we have

$$\begin{aligned} &f(X') - f(X) \\ &= \max_{X' \in \mathcal{F}_q(X)} f(X') - f(X) \\ &\geq \frac{1}{|\mathcal{P}|}\sum_{P \in \mathcal{P}} \{f(X \triangle P) - f(X)\}. \end{aligned}$$

From Proposition 91, we have

$$\sum_{P \in \mathcal{P}} \{f(X \triangle P) - f(X)\} \geq q\ell\frac{m_{2s}}{M_{s,t}} f(X^*) - (pq - q + 1)\ell\frac{M_{s,t}}{m_{2s}} f(X).$$

By combining these inequalities, we have

$$f(X') - f(X) \geq \frac{1}{|\mathcal{P}|}\left\{q\ell\frac{m_{2s}}{M_{s,t}} f(X^*) - (pq - q + 1)\ell\frac{M_{s,t}}{m_{2s}} f(X)\right\}.$$

Since each element in $T \setminus S$ appears in $q\ell$ sets in $\mathcal{P}$ and $|T \setminus S| \leq s$, it holds that $|\mathcal{P}| \leq sq\ell$. Hence, we obtain

$$f(X') - f(X) \geq \frac{1}{s}\left\{\frac{m_{2s}}{M_{s,t}} f(X^*) - (p-1+1/q)\frac{M_{s,t}}{m_{2s}} f(X)\right\}.$$

When using the semi-oblivious variant, due to the property of the algorithm, we have

$$\|(\mathbf{w}^{(X)})_T\|^2 \geq \|(\mathbf{w}^{(X)})_{X \setminus X'}\|^2$$

for any $T \subseteq X$ such that $(X \cup X') \setminus T \in \mathcal{F}_q(X)$. If $\phi_X \colon 2^V \to 2^V$ is a map defined as $\phi_X(S) \in \operatorname{argmin}_{T \colon (X \cup S) \setminus T \in \mathcal{F}_q(X)} \|(\mathbf{w}^{(X)})_T\|^2$, then

$$\begin{aligned} &f(X') - f(X) \\ &= \max_{X' \in \mathcal{F}_q(X) \colon \exists S,\ X' = (X \cup S) \setminus \phi_X(S)} f(X') - f(X) \end{aligned}$$

$$
\begin{aligned}
&\geq \frac{1}{|\mathcal{P}|}\sum_{P\in\mathcal{P}}\{f((X\cup P)\setminus\phi_X(P\setminus X))-f(X)\}\\
&\geq \frac{1}{|\mathcal{P}|}\sum_{P\in\mathcal{P}}\left\{\frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{P\setminus X}\right\|^2-\frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{\phi_X(P\setminus X)}\right\|^2\right\} && \text{(From Lemma 92)}\\
&\geq \frac{1}{|\mathcal{P}|}\sum_{P\in\mathcal{P}}\left\{\frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{P\setminus X}\right\|^2-\frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{P\cap X}\right\|^2\right\}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(since } \left\|\left(\mathbf{w}^{(X)}\right)_{P\cap X}\right\|^2\geq\left\|\left(\mathbf{w}^{(X)}\right)_{\phi_X(P\setminus X)}\right\|^2\text{)}\\
&\geq \frac{1}{|\mathcal{P}|}\left\{q\ell\frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{X^*\setminus X}\right\|^2-(pq-q+1)\ell\frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{X\setminus X^*}\right\|^2\right\}\\
&\geq \frac{1}{|\mathcal{P}|}\left\{q\ell\frac{m_{2s}}{M_{s,t}}f(X^*)-(pq-q+1)\ell\frac{M_{s,t}}{m_{2s}}f(X)\right\} && \text{(From Lemma 93)}\\
&\geq \frac{1}{s}\left\{\frac{m_{2s}}{M_{s,t}}f(X^*)-(p-1+1/q)\frac{M_{s,t}}{m_{2s}}f(X)\right\},
\end{aligned}
$$

where we used Lemma 92 and Lemma 93 as in the oblivious case.

When using the non-oblivious variant, we have

$$
\begin{aligned}
&f(X')-f(X)\\
&\geq \frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{X'\setminus X}\right\|^2-\frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{X\setminus X'}\right\|^2\\
&= \max_{X'\in\mathcal{F}_q(X)}\left\{\frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{X'\setminus X}\right\|^2-\frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{X\setminus X'}\right\|^2\right\}\\
&\geq \frac{1}{|\mathcal{P}|}\sum_{P\in\mathcal{P}}\left\{\frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{P\setminus X}\right\|^2-\frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{P\cap X}\right\|^2\right\}\\
&\geq \frac{1}{|\mathcal{P}|}\left\{q\ell\frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{X^*\setminus X}\right\|^2-(pq-q+1)\ell\frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{X\setminus X^*}\right\|^2\right\}\\
&\geq \frac{1}{|\mathcal{P}|}\left\{q\ell\frac{m_{2s}}{M_{s,t}}f(X^*)-(pq-q+1)\ell\frac{M_{s,t}}{m_{2s}}f(X)\right\}\\
&\geq \frac{1}{s}\left\{\frac{m_{2s}}{M_{s,t}}f(X^*)-(p-1+1/q)\frac{M_{s,t}}{m_{2s}}f(X)\right\},
\end{aligned}
$$

where we used $|\mathcal{P}|\leq sq\ell$ in the last inequality. Therefore, in all three variants, we have

$$
f(X')-f(X)\geq(p-1+1/q)\frac{M_{s,t}}{sm_{2s}}\left\{\frac{1}{p-1+1/q}\frac{m_{2s}^2}{M_{s,t}^2}f(X^*)-f(X)\right\}.
$$

which implies that the distance from the current solution to $\frac{1}{p-1+1/q}\frac{m_{2s}^2}{M_{s,2}^2}$ times the optimal value is decreased by rate $1-(p-1+1/q)m_{2s}/(sM_{s,2})$ at each iteration. Hence, the approximation ratio after T iterations can be bounded as

$$
f(X)\geq\frac{1}{p-1+1/q}\frac{m_{2s}^2}{M_{s,2}^2}\left(1-\left(1-\frac{(p-1+1/q)M_{s,t}}{sm_{2s}}\right)^T\right)f(X^*)
$$

$$\geq \frac{1}{p-1+1/q}\frac{m_{2s}^2}{M_{s,2}^2}\left(1-\exp\left(-\frac{(p-1+1/q)M_{s,2}T}{sm_{2s}}\right)\right)f(X^*),$$

which proves the first statement of the theorem.

Next, we consider the case where the algorithm stops by finding no pair to improve the objective value. For all three variants, we show that

$$0 \geq q\ell\frac{m_{2s}}{M_{s,t}}f(X^*) - (pq-q+1)\ell\frac{M_{s,t}}{m_{2s}}f(X)$$

holds when the algorithm stops, from which the second statement of the theorem follows. When the oblivious variant stops, we have $f(X) \geq f(X')$ for all $X' \in \mathcal{F}_q(X)$. In the same way as the above analysis, we obtain

$$\begin{aligned}0 &\geq \sum_{P\in\mathcal{P}}\{f(X\triangle P) - f(X)\}\\ &\geq q\ell\frac{m_{2s}}{M_{s,t}}f(X^*) - (pq-q+1)\ell\frac{M_{s,t}}{m_{2s}}f(X).\end{aligned}$$

Similarly, when the semi-oblivious variant stops, we have $f(X) \geq f(X-\phi_X(x')+x')$ for all $x' \in V\setminus X$, where $\phi_X(x')$ is defined in the algorithm. Hence, in the same way as the above analysis, we obtain

$$\begin{aligned}0 &\geq \sum_{P\in\mathcal{P}}\{f((X\cup P)\setminus\phi_X(P\setminus X)) - f(X)\}\\ &\geq q\ell\frac{m_{2s}}{M_{s,t}}f(X^*) - (pq-q+1)\ell\frac{M_{s,t}}{m_{2s}}f(X).\end{aligned}$$

When the non-oblivious variant stops, we have

$$0 \geq \frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{X'\setminus X}\right\|^2 - \frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{X\setminus X'}\right\|^2$$

for all $X' \in \mathcal{F}_q(X)$. Therefore, we have

$$\begin{aligned}0 &\geq \sum_{P\in\mathcal{P}}\left\{\frac{1}{2M_{s,t}}\left\|\left(\nabla u(\mathbf{w}^{(X)})\right)_{P\setminus X}\right\|^2 - \frac{M_{s,t}}{2}\left\|\left(\mathbf{w}^{(X)}\right)_{P\cap X}\right\|^2\right\}\\ &\geq q\ell\frac{m_{2s}}{M_{s,t}}f(X^*) - (pq-q+1)\ell\frac{M_{s,t}}{m_{2s}}f(X).\end{aligned}$$

by using the above analysis for the first statement. □

5.7.1 Variants of Geometric Improvement

Here we provide local search algorithms with geometric improvement for p-matroid intersection and p-exchange system constraints. The algorithms start with an singleton with the largest objective value.

Theorem 99. *Suppose $\mathcal{I}$ is the independence set family of a p-matroid intersection or p-exchange system. Let $t = 2p(q+1)$ for the p-matroid intersection case and $t = pq+1$ for the p-exchange system case. Algorithm 14 stops after at most $\mathrm{O}(\frac{s}{\epsilon}\ln(\frac{sM_1}{m_s}))$ iterations, and returns an output X that satisfies*

$$f(X) \geq \left(\frac{1}{p-1+1/q}\frac{m_{2s}^2}{M_{s,t}^2} - \epsilon\right)f(X^*),$$

where X^* is an optimal solution and $s = \max\{|X|\colon X \in \mathcal{I}\}$.

Algorithm 14 Local search algorithms for p-matroid intersection or p-exchange system ($p \geq 2$)

Let $\delta \leftarrow \epsilon/s$.
Let $t = \begin{cases} 2p(q+1) & \text{in the case of } p\text{-matroid intersection constraints} \\ pq+1 & \text{in the case of } p\text{-exchange system constraints.} \end{cases}$
Let $X \leftarrow \operatorname{argmax}\{f(v) \mid v \in V\}$.
loop
  Search for X' that is q-reachable from X such that
$$\begin{cases} f(X') \geq (1+\delta) f(X) & \text{(oblivious)} \\ \exists S,\ X' = (X \cup S) \setminus \phi_X(S) \quad \text{and} \quad f(X') \geq (1+\delta) f(X) & \\ \quad \text{where } \phi_X : 2^V \to 2^V \text{ is a map that satisfies} & \\ \quad \phi_X(S) \in \operatorname{argmin}_{T:\ (X \cup S) \setminus T \in \mathcal{F}_q(X)} \|(\mathbf{w}^{(X)})_T\|^2 & \text{(semi-oblivious)} \\ \dfrac{1}{2M_{s,t}} \left(\nabla u(\mathbf{w}^{(X)})\right)^2_{X' \setminus X} - \dfrac{M_{s,t}}{2} \left(\mathbf{w}^{(X)}\right)^2_{X \setminus X'} > \delta f(X) & \text{(non-oblivious)} \end{cases}$$
  if $\exists\ X'$ satisfying the above condition **then**
    Let $X \leftarrow X'$.
  else
    return X.

Proof. Let X be the output of the algorithm and X^* an optimal solution. From Lemma 85 and Lemma 86 for each case, respectively, we can see that there exists a multiset $\mathcal{P} \subseteq 2^V$ and an integer ℓ that satisfies the following conditions.

1. For all $P \in \mathcal{P}$, the symmetric difference is q-reachable from X, i.e., $X \triangle P \in \mathcal{F}_q(X)$.
2. Each element $v \in X^* \setminus X$ appears in exactly $q\ell$ sets in $\mathcal{P}$.
3. Each element $v \in X \setminus X^*$ appears in at most $(pq - q + 1)\ell$ sets in $\mathcal{P}$.

For each of three variants, we prove

$$0 \geq \sum_{P \in \mathcal{P}} \left\{ \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)})\right)_{P \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)}\right)_{P \cap X} \right\|^2 - \delta f(X) \right\},$$

which implies

$$\begin{aligned} 0 &\geq q\ell \frac{1}{2M_{s,2}} \left\| \left(\nabla u(\mathbf{w}^{(X)})\right)_{X^* \setminus X} \right\|^2 - (pq - q + 1)\ell \frac{M_{s,2}}{2} \left\| \left(\mathbf{w}^{(X)}\right)_{X \setminus X^*} \right\|^2 - \delta |\mathcal{P}| f(X) \\ &\geq q\ell \left\{ \frac{m_{2s}}{M_{s,2}} f(X^*) - \left((p - 1 + 1/q) \frac{M_{s,2}}{m_{2s}} + \delta s \right) f(X) \right\}, \end{aligned}$$

where the second inequality is due to Lemma 93. Since we set $\delta = \epsilon/s$, we obtain

$$f(X) \geq \left(\frac{1}{p - 1 + 1/q} \frac{m_{2s}^2}{M_{s,2}^2} - \epsilon \right) f(X).$$

In the case of the oblivious variant, since $f(X') \leq (1+\delta) f(X)$ for all $X' \in \mathcal{F}_q(X)$, we have

$$0 \geq \sum_{P \in \mathcal{P}} \{ f(X \triangle P) - (1+\delta) f(X) \}$$

$$\geq \sum_{P \in \mathcal{P}} \left\{ \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{P \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{P \cap X} \right\|^2 - \delta f(X) \right\},$$

where the second inequality is due to Lemma 92. In the case of the semi-oblivious variant, since $f((X \cup P) \setminus \phi_X(P \setminus X)) \leq (1+\delta) f(X)$ for any $P \in \mathcal{P}$, we have

$$\begin{aligned}
0 &\geq \sum_{P \in \mathcal{P}} \{ f((X \cup P) \setminus \phi_X(P \setminus X)) - (1+\delta) f(X) \} \\
&\geq \sum_{P \in \mathcal{P}} \left\{ \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{P \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{\phi_X(P \setminus X)} \right\|^2 - \delta f(X) \right\} \\
&\geq \sum_{P \in \mathcal{P}} \left\{ \frac{1}{2M_{s,t}} \left\| \left(\nabla u(\mathbf{w}^{(X)}) \right)_{P \setminus X} \right\|^2 - \frac{M_{s,t}}{2} \left\| \left(\mathbf{w}^{(X)} \right)_{P \cap X} \right\|^2 - \delta f(X) \right\}.
\end{aligned}$$

When we use the non-oblivious variant, since

$$0 \geq \frac{1}{2M_{s,t}} \left(\nabla u(\mathbf{w}^{(X)}) \right)^2_{X' \setminus X} - \frac{M_{s,t}}{2} \left(\mathbf{w}^{(X)} \right)^2_{X \setminus X'} - \delta f(X)$$

for all $X' \in \mathcal{F}_q(X)$, we obtain

$$0 \geq \sum_{P \in \mathcal{P}} \left\{ \frac{1}{2M_{s,t}} \left(\nabla u(\mathbf{w}^{(X)}) \right)^2_{P \setminus X} - \frac{M_{s,t}}{2} \left(\mathbf{w}^{(X)} \right)^2_{P \cap X} - \delta f(X) \right\}.$$

Finally, we bound the number of iterations. At each step, the objective value is improved at least at a rate of $(1+\delta)$. From Lemma 87, the initial solution is $\frac{m_s}{sM_1}$-approximation. Therefore, the number of iterations is at most $\log_{1+\delta}(\frac{sM_1}{m_s}) = \mathrm{O}(\frac{s}{\epsilon} \ln(\frac{sM_1}{m_s}))$. □

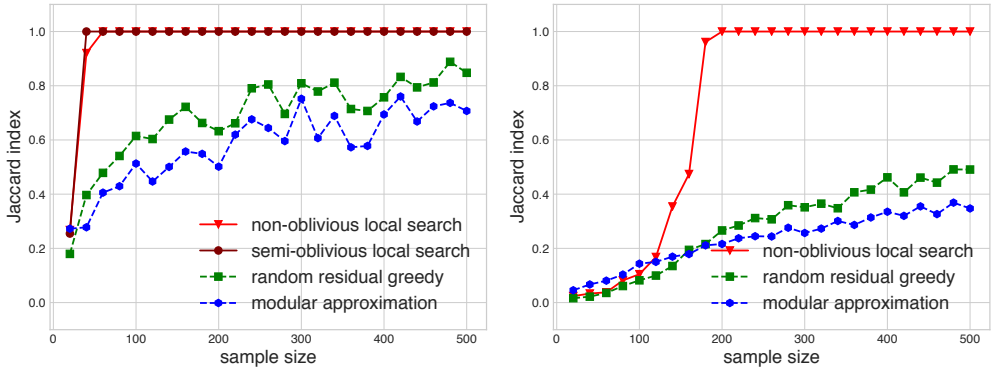
5.8 Experiments

In this section, we conduct experiments on two applications: sparse regression and structure learning of graphical models.

5.8.1 Experiments on Sparse Regression

Datasets. We generate synthetic datasets with a partition matroid constraint. First, we determine the design matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ by generating its each entry according to the uniform distribution on $[0, 1]$. Then we normalize its each column so that the mean is 0 and the standard deviation is 1. Suppose the set of all features are partitioned into k small categories. We randomly select a sparse subset S^* by selecting just one parameter from each category. The response vector is determined by $\mathbf{y} = \mathbf{A}_{S^*}\mathbf{w}$, where $\mathbf{w}$ is a random vector generated from the standard normal distribution. We consider two settings with different dataset sizes. We set $n = 100$ and $k = 10$ in one setting, and we set $n = 1000$ and $k = 50$ in the other setting. For each parameter, we conduct 10 trials and plot the average.

Methods. We implement the non-oblivious local search and semi-oblivious local search algorithms with $q = 1$ out of our proposed methods. For larger datasets with $N = 1000$, the semi-oblivious local search cannot be applied due to its slow running time. As a benchmark, we select the random residual greedy algorithm proposed by Chen et al. [2018a], which randomly selects the element to be added based on the marginal gain at each step. We also implement the modular approximation as a trivial benchmark.



(a) regression, $n = 100$ (b) regression, $n = 1000$

Figure 5.1: The experimental results for sparse regression. 5.1(a) is the result on the case where $n = 100$ and 5.1(b) is the result on the case where $n = 1000$.

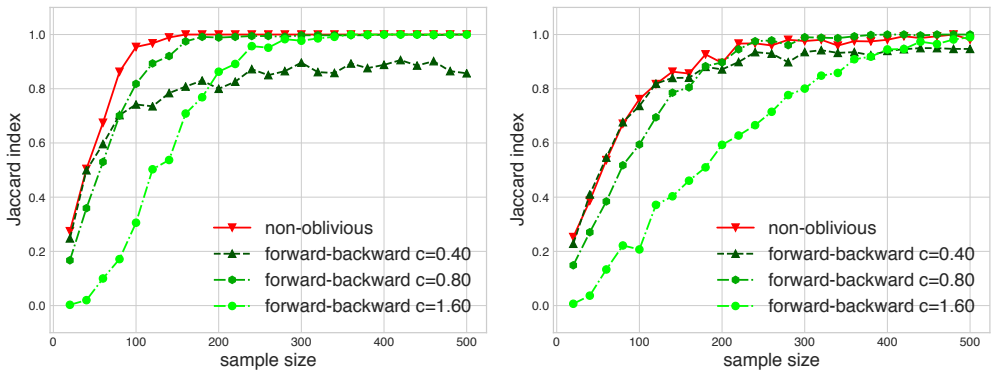
Results. The results are shown in Figure 5.1. First, we compare the proposed methods with $n = 100$. The non-oblivious local search algorithm is competitive with the semi-oblivious local search algorithm. These methods perform better than the benchmark methods the random residual greedy algorithm and the modular approximation in all sample sizes N. Next, we conduct experiments on larger datasets with $n = 1000$. In this setting, we can see the non-oblivious local search algorithms recover the true support for $N \geq 200$, while the other methods do not recover the true support even for $N = 500$.

5.8.2 Experiments on Structure Learning of Graphical Models

Datasets. We synthetically generate samples by Gibbs sampling from two types of Ising models. The one is a path graph and the other is a grid graph. The number of nodes is set to 36 in both settings. For each edge $(u, v) \in E$, we set the parameter $w_{uv} = +0.5$ or $w_{uv} = -0.5$ uniformly at random. At each trial, we determine the parameter randomly and generate a sample of size N by using Gibbs sampling. The sample size N is set to $20i$ for every $i \in \{1, \cdots, 25\}$. We measure the performance of each algorithm by Jaccard index between the true edge set and the edge set returned by the algorithm. For each N, we conduct 10 trials and plot the average.

Methods. Since the oblivious and semi-oblivious methods are too slow to apply to this size of datasets, we select the non-oblivious local search algorithm with $q = 1$ as our proposed method. We use an upper bound $4 \sum_{i=1}^{N} \|\mathbf{x}^i\|_2^3$ instead of $M_{s,3}$ in the non-oblivious local search. As a benchmark, we select an algorithm proposed by Jalali et al. [2011], which solves the edge selection problem for each node separately by the forward-backward greedy algorithm. This algorithm has two parameters: ϵ for forward steps and ν for backward steps. As suggested by Jalali et al. [2011], we set $\epsilon = c \log(Nn)/N$ depending on the number of features and sample size, where c is a tuning constant. We show results for different c. Since we observe that the value of ν does not give an effect on results so much, we show only results where $\nu = 0.1$.

Results. The results are indicated in Figure 5.2. The proposed non-oblivious local search algorithm performs better than the forward-backward greedy algorithm for almost all sample sizes N. The forward-backward greedy algorithm with $c = 0.4$ is competitive with the non-oblivious local search algorithm for the case where the sample size is small, while degrade the performance in the case where the sample size is large. On the other hand, the forward-backward greedy algorithm with $c = 0.8$



(a) graphical, path

(b) graphical, grid

Figure 5.2: The experimental results for structure estimation of graphical models. 5.2(a) is the result on path graphs and 5.2(b) is the result on grid graphs.

performs well for large sample sizes, but performs worse than the non-oblivious local search for small sample sizes.

5.9 Summary and Future Work

In this chapter, we proposed the notion of approximate submodularity for local search and showed that the objective function of feature selection satisfies this property. By utilizing this property, we developed local search algorithms for each of matroid constraints, p-matroid intersection constraints, and p-exchange system constraints. We also devised variants that increase the objective value geometrically. By accelerating each of the proposed local search algorithms, we obtained two faster variants. One is the semi-oblivious local search that quickly decides elements to be removed. The other is the non-oblivious local search that quickly decides elements to be removed and elements to be added. Empirical results on sparse regression and structure estimation of graphical models illustrated the effectiveness of our approach.

A promising direction for future research is to find another application of approximate submodularity for local search. Approximate submodularity for local search is a general property of set functions that guarantees any local optimal to be an approximation to a global optimal. There are many real problems where a local search approach works well in practice but no theoretical guarantee is known. If an objective function of these problems satisfies approximate submodularity for local search, we can devise an efficient approximation algorithms similarly to feature selection.

6 Fast Greedy Algorithms for Dictionary Selection

In this chapter, we propose fast greedy algorithms for dictionary selection with generalized sparsity constraints. This chapter is organized as follows. Section 6.1 states the background and overview of this chapter. Section 6.2 provides the basic concepts and definitions. Section 6.3 formally defines the problem setting. Section 6.4 provides the definition of p-replacement sparsity families and shows existing sparsity constraints are handled under this class. Section 6.5 presents our algorithm, Replacement OMP. Section 6.6 describes the extension to the online setting. The experimental results are presented in Section 6.7. Section 6.8 provides a summary and future work of this chapter.

6.1 Background and Overview

Learning sparse representations of data and signals has been extensively studied for the past decades in machine learning and signal processing [Foucart and Rauhut, 2013]. In these methods, a specific set of basis signals (atoms), called a *dictionary*, is required and used to approximate a given signal in a sparse representation. The design of a dictionary is highly non-trivial, and many studies have been devoted to the construction of a good dictionary for each signal domain, such as natural images and sounds. Recently, approaches to construct a dictionary from data have shown the state-of-the-art results in various domains. The standard approach is called *dictionary learning* [Arora et al., 2014, Zhou et al., 2009, Agarwal et al., 2016]. Although many studies have been devoted to dictionary learning, it is usually difficult to solve, requiring a non-convex optimization problem that often suffers from local minima. Also, standard dictionary learning methods (e.g., MOD [Engan et al., 1999] or k-SVD [Aharon et al., 2006]) require a heavy time complexity.

Krause and Cevher [2010] proposed a combinatorial analog of dictionary learning, called *dictionary selection*. In dictionary selection, given a finite set of candidate atoms, a dictionary is constructed by selecting a few atoms from the set. Dictionary selection could be faster than dictionary learning due to its discrete nature. Another advantage of dictionary selection is that the approximation guarantees hold even in agnostic settings, i.e., we do not need stochastic generating models of the data. Furthermore, dictionary selection algorithms can be used for *media summarization*, in which the atoms must be selected from given data points [Cong et al., 2012, 2017].

The basic setting of dictionary selection is formulated as follows. Let $V = [n]$ be a finite set of candidate atoms. We represent the candidate atoms as a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ whose columns are the atoms in V. For each $t \in [T]$, let $\mathbf{y}_t \in \mathbb{R}^d$ be a data point, for which we want to provide a sparse representation by learning a dictionary, where T is the number of data points. In the basic setting, Krause and Cevher [2010] defined a utility function $u_t \colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ based on the variance reduction metric as $u_t(\mathbf{w}_t) = \|\mathbf{y}_t\|_2^2 - \|\mathbf{y}_t - \mathbf{A}\mathbf{w}_t\|_2^2$. We define $Z_t \subseteq V$ to be the set of atoms used in a sparse representation of data point $\mathbf{y}_t$. Let $f_t(Z_t) = \max_{\mathbf{w} \colon \mathrm{supp}(\mathbf{w}) \subseteq Z_t} u_t(\mathbf{w})$ be a set function that represents the quality of the sparse representation of data point $\mathbf{y}_t$ obtained by using atoms Z_t for each $t \in [T]$. The dictionary selection can be written as the problem of finding a dictionary $X \subseteq V$ of size k that

maximizes

$$h(X) = \sum_{t=1}^{T} \max_{Z_t \subseteq X \colon |Z_t| \leq s} f_t(Z_t).$$

For each data point $\mathbf{y}_t$, the best set of atoms $Z_t \subseteq X$ of size s is used for a sparse representation. We can regard this as a two-stage procedure that selects a dictionary $X \subseteq V$ in the first stage and selects $Z_t \subseteq X$ for the sparse representation of each data point. Since the problem of maximizing $f_t(Z_t)$ is NP-hard even in the case of variance reduction metric [Natarajan, 1995], not only the maximization but also the evaluation of the objective function h are NP-hard.

Our main contribution is a novel and efficient algorithm called Replacement OMP for dictionary selection. This algorithm is based on Replacement Greedy [Stan et al., 2017] for *two-stage submodular maximization*, which is a similar problem to dictionary selection. We extend their approach to dictionary selection with an additional improvement that exploits techniques in orthogonal matching pursuit, and obtain Replacement OMP. Replacement Greedy and Replacement OMP can be viewed as algorithms that unify the local search algorithm developed in Chapter 5 and the greedy algorithm. We compare our method with the previous methods in Table 6.1. Replacement OMP has a smaller running time than $\mathrm{SDS}_{\mathrm{OMP}}$ [Das and Kempe, 2011] and Replacement Greedy. The only exception is $\mathrm{SDS}_{\mathrm{MA}}$ [Das and Kempe, 2011], which intuitively ignores any correlation of the atoms. In our experiment, we demonstrate that Replacement OMP outperforms $\mathrm{SDS}_{\mathrm{MA}}$ in terms of test residual variance. We note that the constant approximation ratios of $\mathrm{SDS}_{\mathrm{MA}}$, Replacement Greedy, and Replacement OMP are incomparable in general. In addition, we demonstrate that Replacement OMP achieves a competitive performance with dictionary learning algorithms in a smaller running time, in numerical experiments.

Incorporating further prior knowledge on the data domain often improves the quality of dictionaries [Rubinstein et al., 2010, Rusu et al., 2014, Dumitrescu and Irofti, 2018]. For example, if the data points are patches of a natural image, most patches are a simple background, and therefore the number of the total size of the supports must be small. Cevher and Krause [2011] proposed a sparsity constraint called the *average sparsity*, in which they add a global constraint $\sum_{t=1}^{T} |Z_t| \leq s'$. Intuitively, the average sparsity constraint requires that the most data points can be represented by a small number of atoms. The average sparsity has been also intensively studied in dictionary learning [Dumitrescu and Irofti, 2018]. To deal with these generalized sparsities in a unified manner, we propose a novel class of sparsity constraints, namely *p-replacement sparsity families*. We prove that Replacement OMP can be applied for the generalized sparsity constraint with a slightly worse approximation ratio. In contrast, Replacement Greedy cannot be extended to the average sparsity setting because it can only handle local constraints on Z_t.

We also consider the *online setting*, in which data points arrive sequentially and we cannot store all of them. We show that Replacement OMP can be extended to the online setting, with a sublinear approximate regret.

6.1.1 Related Work

Related work on dictionary selection. Krause and Cevher [2010] first introduced dictionary selection as a combinatorial analog of dictionary learning. They proposed $\mathrm{SDS}_{\mathrm{MA}}$ and $\mathrm{SDS}_{\mathrm{OMP}}$, and analyzed the approximation ratio using the *coherence* of the matrix $\mathbf{A}$. Balkanski et al. [2016] studied two-stage submodular maximization, which is a problem obtained by replacing f_t in the basic setting of dictionary selection with a monotone submodular function for every $t \in [T]$. Stan et al. [2017] proposed Replacement Greedy for two-stage submodular maximization. Yaghoobi et al. [2014] applied dictionary learning algorithms to dictionary selection.

Table 6.1: Comparison of known methods with Replacement OMP. The constants m_s, M_s, and $M_{s,2}$ are the restricted concavity and smoothness constants of u_t for each $t \in [T]$. The running time is for the case of the variance reduction metric and the individual sparsity constraint. The methods proposed by Krause and Cevher [2010] are indicted by †. The results by Das and Kempe [2011] are indicted by ‡. The method proposed by Stan et al. [2017] is indicted by §.

Method	Approximation ratio	Running time	Generalized sparsity
$\text{SDS}_{\text{MA}}{}^{\dagger}$	$\frac{m_1 m_s}{M_1 M_s}(1-1/\mathrm{e})^{\ddagger}$	$\mathrm{O}((k+d)nT)$	No
$\text{SDS}_{\text{OMP}}{}^{\dagger}$	$\mathrm{O}(1/k)^{\ddagger}$	$\mathrm{O}((s+k)sdknT)$	No
Replacement Greedy§	$\left(\frac{m_{2s}}{M_{s,2}}\right)^2\left(1-\exp\left(-\frac{M_{s,2}}{m_{2s}}\right)\right)$	$\mathrm{O}(s^2dknT)$	No
Replacement OMP	$\left(\frac{m_{2s}}{M_{s,2}}\right)^2\left(1-\exp\left(-\frac{M_{s,2}}{m_{2s}}\right)\right)$	$\mathrm{O}((n+ds)kT)$	Yes

Related work on online dictionary selection. To the best of our knowledge, there is no existing research in the literature that addresses online dictionary selection. For a related problem in sparse optimization, namely *online linear regression*, Kale et al. [2017] proposed an algorithm based on *supermodular minimization* [Liberty and Sviridenko, 2017] with a sublinear approximate regret guarantee. Chen et al. [2018b] dealt with online maximization of weakly DR-submodular functions. All these studies dealt with single-stage versions, which are different from our two-stage setting.

Related work on multi-task feature selection. Many approaches for multi-task feature selection have been proposed from various perspectives. Kolar and Xing [2010] is the first study that considered multi-task feature selection to be a two-stage procedure similar to ours. Their proposed method first screens out irrelevant features by simultaneous orthogonal matching pursuit [Tropp et al., 2006], which does not utilize the two-stage structure of the problem, then selects a set of features for each task out of the remaining features by Adaptive Lasso. Kolar and Xing [2010] also provided a recovery guarantee of their proposed method under the assumption that the maximum and minimum eigenvalues of the covariance matrix are bounded. Also, there are many studies that elaborated Lasso-based regularization terms for multi-task feature selection [Obozinski et al., 2006, Argyriou et al., 2008, Lozano and Swirszcz, 2012].

6.2 Preliminaries

The following lemma is often useful for proving an approximate ratio.

Lemma 100. *Suppose that $\Delta_i, r_i \geq 0$ $(i = 1, 2, \dots)$ satisfies*

$$\Delta_i \geq C\left(v^* - \sum_{j=1}^{i-1}\Delta_j\right) - r_i, \tag{6.1}$$

for $i = 1, 2, \dots$, for some constants $C \in [0, 1]$ and $v^* \geq 0$. Then

$$\sum_{i=1}^{l}\Delta_i \geq \left[1-(1-C)^l\right]v^* - \sum_{i=1}^{l} r_i \geq (1-\exp(-Cl))v^* - \sum_{i=1}^{l} r_i \tag{6.2}$$

for any non-negative integer l.

Proof. We show

$$v^* - \sum_{i=1}^{l} \Delta_i \leq (1-C)^l v^* + \sum_{i=1}^{l} r_i \tag{6.3}$$

for $l = 0, 1, 2, \dots$ by the induction on l. For $l = 0$, (6.3) is trivial. For $l \geq 1$, we have

$$\begin{aligned} v^* - \sum_{i=1}^{l} \Delta_i &= v^* - \sum_{i=1}^{l-1} \Delta_i - \Delta_l \\ &\leq v^* - \sum_{i=1}^{l-1} \Delta_i - C \left(v^* - \sum_{j=1}^{l-1} \Delta_j \right) + r_l \\ &= (1-C) \left(v^* - \sum_{i=1}^{l-1} \Delta_i \right) + r_l. \end{aligned}$$

Now (6.3) follows from the induction and $1 - C \in [0, 1]$. □

6.3 Problem Setting

In this section, we formulate the generalized version of the problem setting of dictionary selection, which we deal with in this chapter. This problem can be viewed as a two-stage combinatorial optimization problem. The first stage is to select $X \subseteq V$ as a dictionary and the second stage is to select $Z_t \subseteq X$ as the atoms for the tth data point for each $t \in [T]$. In this generalized setting, we impose a global constraint on the supports $(Z_t)_{t \in [T]}$, that is, the supports cannot be selected independently for each $t \in [T]$. We formally write such constraints as a down-closed [1] family $\mathcal{I} \subseteq \prod_{t=1}^{T} 2^V$. Therefore, we aim to find $X \subseteq V$ with $|X| \leq k$ maximizing

$$h(X) = \max_{Z_1, \dots, Z_t \subseteq X \colon (Z_1, \dots, Z_t) \in \mathcal{I}} \sum_{t=1}^{T} f_t(Z_t). \tag{6.4}$$

We can see that the original setting of dictionary selection is a special case where $\mathcal{I} = \{(Z_1, \cdots, Z_T) \in \prod_{t=1}^{T} 2^V \mid \forall t \in [T],\ |Z_t| \leq s\}$.

We assume that the atoms are unit vectors in $\mathbb{R}^d$ without loss of generality. Let $\mathbf{w}_t^{(Z_t)}$ denote the maximizer of $u_t(\mathbf{w})$ subject to $\mathrm{supp}(\mathbf{w}) \subseteq Z_t$.

6.3.1 Multi-task Feature Selection

Here we introduce another application of our proposed framework, *multi-task feature selection problem*, which boils down to the same optimization problem as dictionary selection. Multi-task feature selection is the problem of selecting features for different tasks simultaneously. Here a task represents a single machine learning problem instance such as learning a linear classifier that judges whether each patient has a disease or not from health checkup data. Suppose we are given multiple tasks that have similar properties each other, such as disease prediction of similar diseases from the same health checkup data. Multi-task feature selection is the problem of selecting a set of features for each task by utilizing the similarity among the tasks.

[1] A set family $\mathcal{I}$ is said to be down-closed if $X \in \mathcal{I}$ and $Y \subseteq X$ then $Y \in \mathcal{I}$.

We formulate multi-task feature selection as follows. Let $\mathbf{A}_t \in \mathbb{R}^{d\times n}$ and $\mathbf{y}_t \in \mathbb{R}^d$ be the feature matrix and the response vector for the tth task for each $t \in [T]$. Assume that the indices $V = [n]$ of $\mathbf{A}_t$s correspond to each other, that is, the ith columns of all $\mathbf{A}_t$s represent the same feature for all $i \in [n]$. We aim at selecting a set $Z_t \subseteq V$ of features for the tth task for each $t \in [T]$ that leads to a small objective value $f_t(Z_t) = \|\mathbf{y}_t\|_2^2 - \min_{\mathbf{w}_t\colon \mathrm{supp}(\mathbf{w}_t)\subseteq Z_t} \|\mathbf{y}_t - \mathbf{A}_t\mathbf{w}_t\|_2^2$ while taking the similarity among the tasks into account. One natural assumption that represents the similarity is that only a few features appear in $\bigcup_{t\in[T]} Z_t$. Specifically, we assume that there exists a small set $X \subseteq V$ that contains all Z_t, i.e., $Z_t \subseteq X$. By setting a sparsity constraint $\mathcal{I} \in \prod_{t=1}^T 2^V$ on $Z_1, \cdots, Z_T$, this problem coincides with optimization problem (6.4).

6.4 p-Replacement Sparsity Families

In this section, we define a novel class of sparsity families, we call *p-replacement sparsity families*, and show that existing sparsity families are a special case of this class. First, we define the set of *feasible replacements* for the current support $Z_1, \cdots, Z_T$ and an atom a as

$$\mathcal{F}_a(Z_1, \cdots, Z_T) = \left\{(Z_1', \cdots, Z_T') \in \mathcal{I}\colon Z_t' \subseteq Z_t + a,\ |Z_t \setminus Z_t'| \leq 1\ (\forall t \in [T])\right\}. \tag{6.5}$$

That is, the set of members in $\mathcal{I}$ obtained by adding a and removing at most one element from each Z_t. Let $\mathcal{F}(Z_1, \cdots, Z_T) = \bigcup_{a\in V} \mathcal{F}_a(Z_1, \cdots, Z_T)$. If $Z_1, \ldots, Z_T$ are clear from the context, we simply write it as $\mathcal{F}_a$. The p-replacement sparsity families can be defined as a class where any pair of feasible supports can be characterized by a set of feasible replacements as follows.

Definition 101 (p-replacement sparsity)**.** A sparsity constraint $\mathcal{I} \subseteq \prod_{t=1}^T 2^V$ is *p-replacement sparse* if for any $(Z_1, \ldots, Z_T), (Z_1^*, \ldots, Z_T^*) \in \mathcal{I}$, there is a sequence of p feasible replacements $(Z_1^{p'}, \ldots, Z_T^{p'}) \in \mathcal{F}(Z_1, \ldots, Z_T)$ ($p' \in [p]$) such that each element in $Z_t^* \setminus Z_t$ appears at least once in the sequence $(Z_t^{p'} \setminus Z_t)_{p'=1}^p$ and each element in $Z_t \setminus Z_t^*$ appears at most once in the sequence $(Z_t \setminus Z_t^{p'})_{p'=1}^p$.

In the following, We provide examples of p-replacement sparsity families and bound their replacement sparsity parameter.

6.4.1 Individual Matroids

First, we introduce the standard sparsity constraints, which we call *individual sparsity*.

Example 102 (individual sparsity)**.** The sparsity constraint for the standard dictionary selection can be written as $\mathcal{I} = \{(Z_1, \cdots, Z_T) \mid |Z_t| \leq s\ (\forall t \in [T])\}$. We call it *the individual sparsity constraint*.

This individual sparsity constraint is a special case of an *individual matroid constraint*, described below.

Example 103 (individual matroids)**.** This was proposed by Stan et al. [2017] as a sparsity constraint for two-stage submodular maximization. An *individual matroid constraint* can be written as $\mathcal{I} = \{(Z_1, \cdots, Z_T) \mid Z_t \in \mathcal{I}_t\ (\forall t \in [T])\}$ where $(V, \mathcal{I}_t)$ is a matroid[2] for each $t \in [T]$. An individual sparsity constraint is a special case of an individual matroid constraint where $(V, \mathcal{I}_t)$ is the uniform matroid for all t.

Proposition 104. *An individual matroid constraint is k-replacement sparse.*

Proof. Let $(Z_1, \cdots, Z_T), (Z_1^*, \cdots, Z_T^*) \in \mathcal{I}$ be arbitrary sparse subsets. First, we consider the case where Z_t and Z_t^* are both bases[3] of the matroid for all $t \in [T]$. For such Z_t and Z_t^*, we can make

[2] A *matroid* is a pair of a finite ground set V and a non-empty down-closed family $\mathcal{I} \subseteq 2^V$ that satisfy that for all $Z, Z' \in \mathcal{I}$ with $|Z| < |Z'|$, there is an element $a \in Z' \setminus Z$ such that $Z \cup \{a\} \in \mathcal{I}$

[3] For any matroid $(V, \mathcal{I})$, a set $X \in \mathcal{I}$ is called *a base* if it is maximal in $\mathcal{I}$.

k replacements as follows. For each $t \in [T]$, there exists a bijection $\pi_t \colon Z_t^* \to Z_t$ by the exchange property of matroids. For each atom $a^* \in \bigcup_{t=1}^T Z_t^*$, we make a replacement that adds a^* to and removes $\pi_t(a^*)$ from Z_t for all $t \in [T]$ such that $a^* \in Z_t^*$.

If Z_t or Z_t^* is not a base of the matroid, we can add arbitrary atoms to Z_t and Z_t^* until they are both bases, and make k replacements for them in the same way as described above. Removing the atoms that do not exist in Z_t and Z_t^* from these k replacements, we obtain replacements for the original Z_t and Z_t^*. □

6.4.2 Block Sparsity

Example 105 (block sparsity)**.** Block sparsity was proposed by Krause and Cevher [2010]. This sparsity requires that the support must be sparse within each prespecified block. That is, disjoint blocks $B_1, \cdots, B_b \subseteq [T]$ of data points are given in advance, and an only small subset of atoms can be used in each block. Formally, $\mathcal{I} = \{(Z_1, \cdots, Z_T) \mid |\bigcup_{t \in B_{b'}} Z_t| \leq s_{b'} \ (\forall b' \in [b])\}$ where $s_{b'} \in \mathbb{Z}_{\geq 0}$ for each $b' \in [b]$ are sparsity parameters.

Proposition 106. *A block sparsity constraint is k-replacement sparse.*

Proof. Let $(Z_1, \cdots, Z_T), (Z_1^*, \cdots, Z_T^*) \in \mathcal{I}$ be arbitrary sparse subsets. We can make k replacements as follows. Let $Z_{b'} = \bigcup_{t \in B_{b'}} Z_t$ and $Z_{b'}^* = \bigcup_{t \in B_{b'}} Z_t^*$. If $|Z_{b'}| < s_{b'}$ or $|Z_{b'}| < s_{b'}$, we can add arbitrary atoms until these inequalities are tight. For each block $b' \in [b]$, we can make a bijection $\pi_t \colon Z_b^* \to Z_b$. For each atom $a^* \in \bigcup_{t=1}^T Z_t^*$, we make one replacement that adds a^* for all $t \in [T]$ such that $a^* \in Z_t^*$ and removes $\pi_t(a^*)$ from all blocks such that $a^* \in \bigcup_{t \in B_{b'}} Z_t^*$. □

We can show the common generalization of an individual matroid sparsity and block sparsity is also k-replacement sparse by combining the proofs.

6.4.3 Average Sparsity

Example 107 (Average sparsity [Cevher and Krause, 2011])**.** This sparsity imposes a constraint on the average number of used atoms among all data points. The number of atoms used for each data point is also restricted. Formally, $\mathcal{I} = \{(Z_1, \cdots, Z_T) \mid |Z_t| \leq s_t, \sum_{t=1}^T |Z_t| \leq s'\}$ where $s_t \in \mathbb{Z}_{\geq 0}$ for each $t \in [T]$ and $s' \in \mathbb{Z}_{\geq 0}$ are sparsity parameters.

First, we consider an easier case with only a total number constraint, that is, $\mathcal{I} = \{(Z_1, \cdots, Z_T) \mid \sum_{t=1}^T |Z_t| \leq s'\}$. We call it an average sparsity constraint without individual sparsity.

Proposition 108. *An average sparsity constraint without individual sparsity is $(2k-1)$-replacement sparse.*

Proof. Let $(Z_1, \cdots, Z_T), (Z_1^*, \cdots, Z_T^*) \in \mathcal{I}$ be arbitrary feasible sparse subsets. We assume $(Z_1, \cdots, Z_T)$ and $(Z_1^*, \cdots, Z_T^*)$ are maximal in $\mathcal{I}$, but we can deal with non-maximal ones by filling them with dummy elements in the same way as the proof of Proposition 104. Here we show it is possible to greedily make a sequence of $2k-1$ feasible replacements $(Z_1^{r'}, \cdots, Z_T^{r'})_{r'=1}^{2k-1}$ such that each atom in $Z_t^* \setminus Z_t$ appears at least once in the sequence $(Z_t^{r'} \setminus Z_t)_{r'=1}^{2k-1}$ and each atom in $Z_t \setminus Z_t^*$ appears at most once in the sequence $(Z_t \setminus Z_t^{r'})_{r'=1}^{2k-1}$.

Let X and X^* be the sets of atoms appearing in $(Z_1, \cdots, Z_T)$ and $(Z_1^*, \cdots, Z_T^*)$, respectively. We arrange the atoms in each of X and X^* in an arbitrary order and consider them one by one in parallel. Let us suppose we currently consider $a \in X$ and $a^* \in X^*$. We make a replacement that adds a^* for several $t \in [T]$ and removes a for the other several $t \in [T]$ in the following way. Let τ be the number of

$Z_t \setminus Z_t^*$ that contains a, i.e., $\tau = |\{t \in [T] \mid a \in Z_t \setminus Z_t^*\}|$ and τ^* the number of $Z_t^* \setminus Z_t$ that contains a^*, i.e., $\tau = |\{t \in [T] \mid a^* \in Z_t^* \setminus Z_t\}|$. If $\tau > \tau^*$, we can let this replacement add a^* for all $t \in [T]$ such that $a^* \in Z_t^* \setminus Z_t$ and remove a for any subset of $\{t \in [T] \mid a \in Z_t \setminus Z_t^*\}$ with size τ^*. Conversely, if $\tau \leq \tau^*$, we can let this replacement add a^* for an arbitrary subset of $\{t \in [T] \mid a^* \in Z_t^* \setminus Z_t\}$ of size τ and remove a for all $t \in [T]$ such that $a \in Z_t \setminus Z_t^*$. We proceed to a next replacement after removing a^* from Z_t^* for all $t \in [T]$ such that a^* is added in this replacement, and a from Z_t for all $t \in [T]$ such that a is removed in this replacement. If $a \notin Z_t \setminus Z_t^*$ for all $t \in [T]$, we move the focus from a to the next atom. Similarly, if $a^* \notin Z_t^* \setminus Z_t$ for all $t \in [T]$, we move the focus from a^* to the next atom.

This procedure ends after at most $2k - 1$ iterations. This is because at each iteration we move the focus from a to the next atom in X or from a^* to the next atom in X^*, and we have $|X| \leq k$ and $|X^*| \leq k$. □

Here we show this bound is tight for an average sparsity constraint without individual sparsity by giving an example.

Example 109. Assume $T \geq k^2$. For simplicity, we further assume T is a multiple of k. Let us consider the case of $s' = T$, i.e., $\mathcal{I} = \{(Z_1, \cdots, Z_T) \mid \sum_{t=1}^{T} |Z_t| \leq T\}$. Let $V = \{v_1, \cdots, v_{2k}\}$ be the ground set. Here we show the replacement sparsity parameter of this sparsity constraint is at least $2k - 1$ by giving $(Z_1, \cdots, Z_T)$ and $(Z_1^*, \cdots, Z_T^*)$ that require $2k - 1$ replacements. Suppose $Z_t = \{v_1, \cdots, v_k\}$ for $1 \leq t \leq T/k$ and $Z_t = \varnothing$ for other t. Let $Z_t^* = \{v_{k+1}\}$ for $1 \leq t \leq T-k+1$ and $Z_{T-k+i}^* = \{v_{k+i}\}$ for each $i = 2, \cdots, k$. .

It can be seen that we must use $k-1$ different replacements for $Z_{T-k+2}^*, \cdots, Z_T^*$. In each replacement, an added element is restricted to a single atom, but $Z_{T-k+2}^*, \cdots, Z_T^*$ are all singleton sets of different atoms. Then elements in $Z_{T-k+2}^*, \cdots, Z_T^*$ must be dealt with by different replacements, and $k - 1$ replacements are needed.

In addition, we must use k other replacements for $Z_1^*, \cdots, Z_{T-k+1}^*$. Since $(Z_1, \cdots, Z_T)$ is maximal in $\mathcal{I}$, the total number of added atoms of each replacement must be at most the total number of removed atoms of this replacement. However, in each replacement, the number of atoms removed from each Z_t is at most one, and only $Z_1, \cdots, Z_{T/k}$ are non-empty, hence at most T/k elements can be removed in each replacement. Therefore, we must use k different replacements for $Z_1^*, \cdots, Z_{T-k+1}^*$ because there are $T - k + 1$ singleton sets $Z_1^*, \cdots, Z_{T-k+1}^*$ and $T \geq k^2$.

In conclusion, the replacement sparsity parameter of this sparsity constraint is at least $2k - 1$.

We bound the replacement sparsity parameter of an average sparsity constraint based on the analysis on average sparsity without individual sparsity.

Proposition 110. *An average sparsity constraint is $(3k - 1)$-replacement sparse.*

Proof. Here we give a sequence of $3k - 1$ replacements that satisfies the conditions for replacement sparsity.

First, we use k replacements for dealing with the individual sparsity constraints. Let $S \subseteq [T]$ be the set of indices such that $|Z_t| = s_t$. For each $a^* \in X^*$, we make a replacement that adds a^* for all $t \in S$ such that $a^* \in Z_t^* \setminus Z_t$ and possibly removes an atom in $Z_t \setminus Z_t^*$ for all $t \in S$. By selecting the removed atoms so that they do not overlap, we can define these k replacements such that, for all $t \in S$, each atom in $Z_t^* \setminus Z_t$ is added once and each atom in $Z_t \setminus Z_t^*$ is removed once.

For the rest of the elements, we need not consider the individual sparsity constraints, therefore the rest elements can be dealt with $2k-1$ replacements in the same way as the proof of Proposition 108. □

6.5 Algorithms

In this section, we present Replacement Greedy [Stan et al., 2017] and Replacement OMP for dictionary selection with generalized sparsity constraints.

6.5.1 Replacement Greedy

Replacement Greedy was first proposed as an algorithm for a different problem, *two-stage submodular maximization* [Balkanski et al., 2016]. In two-stage submodular maximization, the goal is to maximize

$$h(X) = \sum_{t=1}^{T} \max_{Z_t \subseteq X \colon Z_t \in \mathcal{I}_t} f_t(Z_t), \tag{6.6}$$

where f_t is a non-negative monotone submodular function ($t \in [T]$) and $\mathcal{I}_t$ is a matroid. Despite the similarity of the formulation, in dictionary selection, the functions f_t are not necessarily submodular, but come from the continuous function u_t. Furthermore, in two-stage submodular maximization, the constraints on Z_t are individual for each $t \in [T]$, while we pose a global constraint $\mathcal{I}$. In the following, we present an adaptation of Replacement Greedy to dictionary selection with generalized sparsity constraints.

Replacement Greedy stores the current dictionary X and supports $Z_t \subseteq X$ such that $(Z_1, \dots, Z_T) \in \mathcal{I}$, which are initialized as $X = \varnothing$ and $Z_t = \varnothing$ ($t \in [T]$). At each step, the algorithm considers the gain of adding an element $a \in V$ to X with respect to each function f_t, i.e., the algorithm selects a that maximizes $\max_{(Z'_1,\dots,Z'_T)\in\mathcal{F}_a} \sum_{t=1}^{T} \{f_t(Z'_t) - f(Z_t)\}$. See Algorithm 15 for a pseudocode description. Note that for the individual matroid constraint $\mathcal{I}$, the algorithm coincides with the original Replacement Greedy [Stan et al., 2017].

Algorithm 15 Replacement Greedy & Replacement OMP

1: Initialize $X \leftarrow \varnothing$ and $Z_t \leftarrow \varnothing$ for $t = 1, \dots, T$.
2: **for** $i = 1, \dots, k$ **do**
3: Pick $a^* \in V$ that maximizes

$$\begin{cases} \max\limits_{(Z'_1,\cdots,Z'_T)\in\mathcal{F}_{a^*}} \sum\limits_{t=1}^{T} \left\{ f_t(Z'_t) - f_t(Z_t) \right\} & \text{(Replacement Greedy)} \\ \max\limits_{(Z'_1,\cdots,Z'_T)\in\mathcal{F}_{a^*}} \left\{ \dfrac{1}{M_{s,2}} \sum\limits_{t=1}^{T} \left\| \nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z'_t \setminus Z_t} \right\|^2 - M_{s,2} \sum\limits_{t=1}^{T} \left\| (\mathbf{w}_t^{(Z_t)})_{Z_t \setminus Z'_t} \right\|^2 \right\} & \text{(Replacement OMP)} \end{cases}$$

and let $(Z'_1, \cdots, Z'_T)$ be a replacement achieving a maximum.
4: Set $X \leftarrow X + a^*$ and $Z_t \leftarrow Z'_t$ for each $t \in [T]$.
5: **return** X.

Stan et al. [2017] showed that Replacement Greedy achieves an $((1 - 1/\sqrt{e})/2)$-approximation when f_t are monotone submodular. We extend their analysis to our non-submodular setting. First, we show the following key lemma.

Lemma 111. *Assume $\mathcal{I}$ is p-replacement sparse. Suppose that at some step, the solution is updated from $(Z_1, \cdots, Z_T)$ to $(Z'_1, \cdots, Z'_T)$ by Replacement Greedy. Let $(Z_1^*, \cdots, Z_T^*) \in \operatorname{argmax}_{(Z_1,\cdots,Z_T)\in\mathcal{I} \colon Z_t \subseteq X^*} f_t(Z)$ where X^* is an optimal solution for dictionary selection. Then, the marginal gain of Replacement Greedy is bounded from below as follows.*

$$\sum_{t=1}^{T} f_t(Z'_t) - \sum_{t=1}^{T} f_t(Z_t) \geq \frac{1}{p} \left\{ \frac{m_{2s}}{M_{s,2}} \sum_{t=1}^{T} f_t(Z_t^*) - \frac{M_{s,2}}{m_{2s}} \sum_{t=1}^{T} f_t(Z_t) \right\}$$

where $s = \max_{(Z_t)_{t=1}^T \in \mathcal{I}} \max_{t\in[T]} |Z_t|$.

Proof. Note that from the condition on feasible replacements, we have $|Z_t \triangle Z_t'| \leq 2$. Since u_t is $M_{s,2}$-smooth on $\Omega_{s,2}$, it holds that for any $\mathbf{z} \in \mathbb{R}^n$ with $\mathrm{supp}(\mathbf{z}) \subseteq Z_t' \setminus Z_t$,

$$\begin{aligned}
f_t(Z_t') - f_t(Z_t) &= u_t(\mathbf{w}^{(Z_t')}) - u_t(\mathbf{w}^{(Z_t)}) \\
&\geq u_t((\mathbf{w}^{(Z_t)})_{Z_t \cap Z_t'} + \mathbf{z}) - u_t(\mathbf{w}^{(Z_t)}) \\
&\geq \left\langle \nabla u_t(\mathbf{w}^{(Z_t)}), \mathbf{z} - (\mathbf{w}^{(Z_t)})_{Z_t \setminus Z_t'} \right\rangle - \frac{M_{s,2}}{2} \|\mathbf{z} - (\mathbf{w}^{(Z_t)})_{Z_t \setminus Z_t'}\|^2
\end{aligned}$$

Since this inequality holds for every $\mathbf{z}$ with $\mathrm{supp}(\mathbf{z}) \subseteq Z_t' \setminus Z_t$, by optimizing it for $\mathbf{z}$, we obtain

$$f_t(Z_t') - f_t(Z_t) \geq \frac{1}{2M_{s,2}} \|\nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z_t' \setminus Z_t}\|^2 - \frac{M_{s,2}}{2} \|(\mathbf{w}_t^{(Z_t)})_{Z_t \setminus Z_t'}\|^2. \tag{6.7}$$

In addition, due to the strong concavity of u_t, we have

$$\begin{aligned}
f_t(Z_t^*) - f_t(Z_t) &= u_t(\mathbf{w}^{(Z_t^*)}) - u_t(\mathbf{w}^{(Z_t)}) \\
&\leq \left\langle \nabla u_t(\mathbf{w}_t^{(Z_t)}), \mathbf{w}_t^{(Z_t^*)} - \mathbf{w}_t^{(Z_t)} \right\rangle - \frac{m_{2s}}{2} \left\| \mathbf{w}_t^{(Z_t^*)} - \mathbf{w}_t^{(Z_t)} \right\|^2 \\
&\leq \max_{\mathbf{z}\colon \mathrm{supp}(\mathbf{z}) \subseteq Z_t^*} \left\{ \left\langle \nabla u_t(\mathbf{w}_t^{(Z_t)}), \mathbf{z} - \mathbf{w}_t^{(Z_t)} \right\rangle - \frac{m_{2s}}{2} \left\| \mathbf{z} - \mathbf{w}_t^{(Z_t)} \right\|^2 \right\} \\
&= \frac{1}{2m_{2s}} \left\| (\nabla u_t(\mathbf{w}^{(Z_t)}))_{Z_t^* \setminus Z_t} \right\|^2 - \frac{m_{2s}}{2} \left\| (\mathbf{w}^{(Z_t)})_{Z_t \setminus Z_t^*} \right\|^2.
\end{aligned} \tag{6.8}$$

Similarly, due to the strong concavity of u_t, we have

$$\begin{aligned}
-f_t(Z_t) &= u_t(\mathbf{0}) - u_t(\mathbf{w}_t^{(Z_t)}) \\
&\leq \left\langle \nabla u_t(\mathbf{w}_t^{(Z_t)}), -\mathbf{w}_t^{(Z_t)} \right\rangle - \frac{m_{2s}}{2} \left\| \mathbf{w}_t^{(Z_t)} \right\|^2 \\
&= -\frac{m_{2s}}{2} \left\| \mathbf{w}_t^{(Z_t)} \right\|^2 \\
&\leq -\frac{m_{2s}}{2} \left\| (\mathbf{w}_t^{(Z_t)})_{Z_t \setminus Z_t^*} \right\|^2
\end{aligned} \tag{6.9}$$

Since $\mathcal{I}$ is p-replacement sparse, we can take a sequence of p replacements $(Z_1^{p'}, \cdots, Z_T^{p'})_{p'=1}^p$ such that

- $(Z_1^{p'}, \cdots, Z_T^{p'}) \in \mathcal{F}(Z_1, \cdots, Z_T)$,
- each element in $Z_t^* \setminus Z_t$ appears at least once in sequence $(Z_t^{p'} \setminus Z_t)_{p'=1}^p$ for each $t \in [T]$,
- each element in $Z_t \setminus Z_t^*$ appears at most once in sequence $(Z_t \setminus Z_t^{p'})_{p'=1}^p$ for each $t \in [T]$.

Now we prove the lemma by utilizing these properties.

$$\begin{aligned}
&\sum_{t=1}^T f_t(Z_t') - \sum_{t=1}^T f_t(Z_t) \\
&\geq \frac{1}{p} \sum_{p'=1}^p \left\{ \sum_{t=1}^T f_t(Z_t^{p'}) - \sum_{t=1}^T f_t(Z_t) \right\} \\
&\qquad\qquad \text{(by the choice of } (Z_1', \ldots, Z_T') \text{ and the feasibility of } (Z_1^{p'}, \cdots, Z_T^{p'}))
\end{aligned}$$

$$
\begin{aligned}
&\geq \frac{1}{p}\sum_{p'=1}^{p}\sum_{t=1}^{T}\left\{\frac{1}{2M_{s,2}}\left\|\nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z_t^{p'}\setminus Z_t}\right\|^2 - \frac{M_{s,2}}{2}\left\|\left(\mathbf{w}_t^{(Z_t)}\right)_{Z_t\setminus Z_t^{p'}}\right\|^2\right\} && \text{(by (6.7))}\\
&\geq \frac{1}{p}\sum_{t=1}^{T}\left\{\frac{1}{2M_{s,2}}\left\|\nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z_t^*\setminus Z_t}\right\|^2 - \frac{M_{s,2}}{2}\left\|\left(\mathbf{w}_t^{(Z_t)}\right)_{Z_t\setminus Z_t^*}\right\|^2\right\} && \text{(by the property of } (Z_t^{p'})_{p'=1}^{p})\\
&\geq \frac{1}{p}\sum_{t=1}^{T}\left\{\frac{m_{2s}}{M_{s,2}}\left(f_t(Z_t^*) - f_t(Z_t)\right) - \left(\frac{M_{s,2}}{m_{2s}} - \frac{m_{2s}}{M_{s,2}}\right)f_t(Z_t)\right\} && \text{(by (6.8) and (6.9))}\\
&= \frac{1}{p}\sum_{t=1}^{T}\left\{\frac{m_{2s}}{M_{s,2}}f_t(Z_t^*) - \frac{M_{s,2}}{m_{2s}}f_t(Z_t)\right\}.
\end{aligned}
$$

□

Finally, we obtain the following theorem.

Theorem 112. *Assume that u_t is m_{2s}-strongly concave on Ω_{2s} and $M_{s,2}$-smooth on $\Omega_{s,2}$ for $t \in [T]$ and that the sparsity constraint $\mathcal{I}$ is p-replacement sparse. Let $(Z_1^*, \cdots, Z_T^*) \in \mathcal{I}$ be optimal supports of an optimal dictionary X^*. Then the solution $(Z_1, \cdots, Z_T) \in \mathcal{I}$ of Replacement Greedy after k' steps satisfies*

$$
\sum_{t=1}^{T} f_t(Z_t) \geq \frac{m_{2s}^2}{M_{s,2}^2}\left(1 - \exp\left(-\frac{k'}{p}\frac{M_{s,2}}{m_{2s}}\right)\right)\sum_{t=1}^{T} f_t(Z_t^*).
$$

Proof. By combining Lemma 111 with Lemma 100, we obtain the statement. □

6.5.2 Replacement OMP

Now we propose our algorithm, Replacement OMP. A down-side of Replacement Greedy is its heavy computation: in each greedy step, we need to evaluate $\sum_{t=1}^{T} f_t(Z_t')$ for each $(Z_1', \ldots, Z_t') \in \mathcal{F}_a(Z_1, \ldots, Z_t)$, which amounts to solving linear regression problems snT times if u is the variance reduction metric. To avoid heavy computation, we propose a proxy of this quantity by borrowing an idea from orthogonal matching pursuit. Replacement OMP selects an atom $a \in V$ that maximizes

$$
\max_{(Z_1', \cdots, Z_T') \in \mathcal{F}_a(Z_1, \cdots, Z_T)} \left\{\frac{1}{M_{s,2}}\sum_{t=1}^{T}\|\nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z_t'\setminus Z_t}\|^2 - M_{s,2}\sum_{t=1}^{T}\|(\mathbf{w}_t^{(Z_t)})_{Z_t\setminus Z_t'}\|^2\right\}. \tag{6.10}
$$

This algorithm requires the smoothness parameter $M_{s,2}$ before the execution. Computing $M_{s,2}$ is generally difficult, but this parameter for the variance reduction metric can be bounded by $\lambda_{\max}(\mathbf{A}, 2)$. This value can be computed in $O(n^2 d)$ time.

Lemma 113. *Assume $\mathcal{I}$ is p-replacement sparse. Suppose at some step, the solution is updated from $(Z_1, \cdots, Z_T)$ to $(Z_1', \cdots, Z_T')$ by Replacement OMP. Let $(Z_1^*, \cdots, Z_T^*) \in \operatorname{argmax}_{(Z_1, \cdots, Z_T) \in \mathcal{I}:\ Z_t \subseteq X^*} f_t(Z)$ where X^* is an optimal solution for dictionary selection. Then, the marginal gain of Replacement OMP is bounded from below as follows.*

$$
\sum_{t=1}^{T} f_t(Z_t') - \sum_{t=1}^{T} f_t(Z_t) \geq \frac{1}{p}\left\{\frac{m_{2s}}{M_{s,2}}\sum_{t=1}^{T} f_t(Z_t^*) - \frac{M_{s,2}}{m_{2s}}\sum_{t=1}^{T} f_t(Z_t)\right\},
$$

where $s = \max_{(Z_t)_{t=1}^T \in \mathcal{I}} \max_{t \in [T]} |Z_t|$.

Proof. We can obtain the following inequalities from the restricted strong concavity and smoothness of u_t in the same way as the above proof of Lemma 111.

$$f_t(Z'_t) - f_t(Z_t) \geq \frac{1}{2M_{s,2}} \left\| \nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z'_t \setminus Z_t} \right\|^2 - \frac{M_{s,2}}{2} \left\| (\mathbf{w}_t^{(Z_t)})_{Z_t \setminus Z'_t} \right\|^2. \tag{6.11}$$

$$f_t(Z^*_t) - f_t(Z_t) \leq \frac{1}{2m_{2s}} \left\| (\nabla u_t(\mathbf{w}^{(Z_t)}))_{Z^*_t \setminus Z_t} \right\|^2 - \frac{m_{2s}}{2} \left\| (\mathbf{w}^{(Z_t)})_{Z_t \setminus Z^*_t} \right\|^2. \tag{6.12}$$

$$-f_t(Z_t) \leq -\frac{m_{2s}}{2} \left\| (\mathbf{w}_t^{(Z_t)})_{Z_t \setminus Z^*_t} \right\|^2. \tag{6.13}$$

Since $\mathcal{I}$ is p-replacement sparse, we can take a sequence of p replacements $(Z_1^{p'}, \cdots, Z_T^{p'})_{p'=1}^p$ that satisfies the properties mentioned in the proof of Lemma 111. With these properties, we obtain

$$\begin{aligned}
&\sum_{t=1}^T f_t(Z'_t) - \sum_{t=1}^T f_t(Z_t) \\
&\geq \sum_{t=1}^T \left\{ \frac{1}{2M_{s,2}} \left\| \nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z'_t \setminus Z_t} \right\|^2 - \frac{M_{s,2}}{2} \left\| (\mathbf{w}_t^{(Z_t)})_{Z_t \setminus Z'_t} \right\|^2 \right\} && \text{(by (6.11))} \\
&\geq \frac{1}{p} \sum_{p'=1}^p \sum_{t=1}^T \left\{ \frac{1}{2M_{s,2}} \left\| \nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z_t^{p'} \setminus Z_t} \right\|^2 - \frac{M_{s,2}}{2} \left\| \left(\mathbf{w}_t^{(Z_t)}\right)_{Z_t \setminus Z_t^{p'}} \right\|^2 \right\} \\
& && \text{(by the choice of } (Z'_1, \ldots, Z'_T) \text{ and the feasibility of } (Z_1^{p'}, \cdots, Z_T^{p'})) \\
&\geq \frac{1}{p} \sum_{t=1}^T \left\{ \frac{1}{2M_{s,2}} \left\| \nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z^*_t \setminus Z_t} \right\|^2 - \frac{M_{s,2}}{2} \left\| \left(\mathbf{w}_t^{(Z_t)}\right)_{Z_t \setminus Z^*_t} \right\|^2 \right\} && \text{(by the property of } (Z_t^{p'})_{p'=1}^p) \\
&\geq \frac{1}{p} \sum_{t=1}^T \left\{ \frac{m_{2s}}{M_{s,2}} \left(f_t(Z^*_t) - f_t(Z_t)\right) - \left(\frac{M_{s,2}}{m_{2s}} - \frac{m_{2s}}{M_{s,2}} \right) f_t(Z_t) \right\} && \text{(by (6.12) and (6.13))} \\
&= \frac{1}{p} \sum_{t=1}^T \left\{ \frac{m_{2s}}{M_{s,2}} f_t(Z^*_t) - \frac{M_{s,2}}{m_{2s}} f_t(Z_t) \right\}.
\end{aligned}$$

□

Finally, we obtain the following bound on the approximation ratio of Replacement OMP.

Theorem 114. *Assume that u_t is m_{2s}-strongly concave on Ω_{2s} and $M_{s,2}$-smooth on $\Omega_{s,2}$ for $t \in [T]$ and that the sparsity constraint $\mathcal{I}$ is p-replacement sparse. Let $(Z^*_1, \cdots, Z^*_T) \in \mathcal{I}$ be optimal supports of an optimal dictionary X^*. Then the solution $(Z_1, \cdots, Z_T) \in \mathcal{I}$ of Replacement OMP after k' steps satisfies*

$$\sum_{t=1}^T f_t(Z_t) \geq \frac{m_{2s}^2}{M_{s,2}^2} \left(1 - \exp\left(-\frac{k'}{p} \frac{M_{s,2}}{m_{2s}} \right) \right) \sum_{t=1}^T f_t(Z^*_t).$$

Proof. By combining Lemma 113 with Lemma 100, we obtain the statement. □

6.5.3 Replacement Deletion-OMP

In this section, we propose an intermediate algorithm between Replacement Greedy and Replacement OMP, which we call Replacement Deletion-OMP. Though we can apply Replacement Deletion-OMP to generalized sparsity constraints, we describe its simplest version for the individual sparsity constraints, i.e., $\mathcal{I} = \{(Z_1, \cdots, Z_T) \mid |Z_t| \leq s \ (\forall t \in [T])\}$ for some $s \in \mathbb{Z}_{\geq 0}$, for simplicity.

Replacement Deletion-OMP adds atoms one by one similarly to Replacement Greedy and Replacement Deletion-OMP, but selects the atom to be added in a different way. At each step, Replacement Deletion-OMP finds an element $a_t \in Z_t$ that minimizes $\left|(\mathbf{w}_t^{(Z_t)})_{a_t}\right|$ for each $t \in [T]$. Then Replacement Deletion-OMP selects an element $a^* \in V$ to be added by maximizing $\sum_{t=1}^{T} \max\{0, f_t(Z_t + a^* - a_t)\}$. We can say that Replacement Deletion-OMP is an intermediate algorithm in the sense that it selects an element to be removed in a similar way to Replacement OMP and an element to be added in a similar way to Replacement Greedy.

Since Replacement Deletion-OMP selects an element to be removed without evaluating the objective value, it runs faster than Replacement Greedy. Also, in comparison to Replacement OMP, Replacement Deletion-OMP has an advantage since it does not use the value of $M_{s,2}$ for evaluating the approximate marginal gain. The time required by computation of $M_{s,2}$ varies from case to case, but its dependence on n is quadratic in most cases.

As illustrated above, Replacement Deletion-OMP is suitable for the cases where n is large. Since the ground set of dictionary selection is the set of atoms taken from existing dictionaries, its size n is not expected to be so large in realistic situations. In cases where n is large such as multi-task feature selection, Replacement Deletion-OMP can be a reasonable choice.

Lemma 115. *Assume $\mathcal{I}$ is p-replacement sparse. Suppose at some step, the solution is updated from $(Z_1, \cdots, Z_T)$ to $(Z'_1, \cdots, Z'_T)$ by Replacement Deletion-OMP. Let $(Z_1^*, \cdots, Z_T^*) \in \operatorname{argmax}_{(Z_1,\cdots,Z_T)\in\mathcal{I}:\ Z_t\subseteq X^*} f_t(Z)$ where X^* is an optimal solution for dictionary selection. Then, the marginal gain of Replacement Deletion-OMP is bounded from below as follows.*

$$\sum_{t=1}^{T} f_t(Z'_t) - \sum_{t=1}^{T} f_t(Z_t) \geq \frac{1}{p}\left\{\frac{m_{2s}}{M_{s,2}}\sum_{t=1}^{T} f_t(Z_t^*) - \frac{M_{s,2}}{m_{2s}}\sum_{t=1}^{T} f_t(Z_t)\right\},$$

where $s = \max_{(Z_t)_{t=1}^T\in\mathcal{I}} \max_{t\in[T]} |Z_t|$.

Proof. In the same way as the proofs for Replacement Greedy, we obtain the following inequalities from the restricted strong concavity and smoothness of u_t.

$$f_t(Z'_t) - f_t(Z_t) \geq \frac{1}{2M_{s,2}}\left\|\nabla u_t(\mathbf{w}_t^{(Z_t)})_{Z'_t\setminus Z_t}\right\|^2 - \frac{M_{s,2}}{2}\left\|(\mathbf{w}_t^{(Z_t)})_{Z_t\setminus Z'_t}\right\|^2. \tag{6.14}$$

$$f_t(Z_t^*) - f_t(Z_t) \leq \frac{1}{2m_{2s}}\left\|(\nabla u_t(\mathbf{w}^{(Z_t)}))_{Z_t^*\setminus Z_t}\right\|^2 - \frac{m_{2s}}{2}\left\|(\mathbf{w}^{(Z_t)})_{Z_t\setminus Z_t^*}\right\|^2. \tag{6.15}$$

$$-f_t(Z_t) \leq -\frac{m_{2s}}{2}\left\|(\mathbf{w}_t^{(Z_t)})_{Z_t\setminus Z_t^*}\right\|^2. \tag{6.16}$$

Since an element to be removed a^*

$$\begin{aligned}
\sum_{t=1}^{T} f_t(Z'_t) - \sum_{t=1}^{T} f_t(Z_t) &= \sum_{t=1}^{T} \max\{0, f_t(Z_t + a^* - a_t) - f_t(Z_t)\} \\
&\geq \frac{1}{k}\sum_{x\in X^*}\sum_{t=1}^{T} \max\{0, f_t(Z_t + x - a_t) - f_t(Z_t)\} \qquad \text{(by the choice of } a^*\text{)} \\
&\geq \frac{1}{k}\sum_{t=1}^{T}\sum_{x\in Z_t^*} \{f_t(Z_t + x - a_t) - f_t(Z_t)\} \\
&\geq \frac{1}{k}\sum_{t=1}^{T}\sum_{x\in Z_t^*} \left\{\frac{1}{2M_{s,2}}\left(\nabla u_t\left(\mathbf{w}_t^{(Z_t)}\right)\right)_x^2 - \frac{M_{s,2}}{2}\left(\mathbf{w}_t^{(Z_t)}\right)_{a_t}^2\right\} \qquad \text{(by (6.14))}
\end{aligned}$$

Algorithm 16 Replacement Deletion-OMP

Initialize $X \leftarrow \varnothing$ and $Z_t \leftarrow \varnothing$ for $t = 1, \dots, T$.
for $i = 1, \dots, k$ **do**
  Let $a_t \in \operatorname{argmin}_{a \in Z_t} \left|(\mathbf{w}_t^{(Z_t)})_a\right|$ for each $t \in [T]$.
  Pick $a^* \in V$ that maximizes $\sum_{t=1}^T \max\{0, f_t(Z_t + a^* - a_t)\}$.
  Set $X \leftarrow X + a^*$ and if $f_t(Z_t + a^* - a_t) > f_t(Z_t)$, then $Z_t \leftarrow Z_t + a^* - a_t$ for each $t \in [T]$.
return X.

$$\geq \frac{1}{k} \sum_{t=1}^{T} \left\{ \frac{1}{2M_{s,2}} \left\| \left(\nabla u_t \left(\mathbf{w}_t^{(Z_t)} \right) \right)_{Z_t^*} \right\|^2 - \frac{M_{s,2}}{2} \left\| \left(\mathbf{w}_t^{(Z_t)} \right)_{Z_t} \right\|^2 \right\}$$

(by the choice of a_t)

$$\geq \frac{1}{k} \sum_{t=1}^{T} \left\{ \frac{m_{s+s^*}}{M_{s,2}} f_t(Z_t^*) - \frac{M_{s,2}}{m_{s+s^*}} f_t(Z_t) \right\} \qquad \text{(by (6.15) and (6.16))}$$

□

Here we show the same approximation ratio bound for Replacement Deletion-OMP.

Theorem 116. *Assume that u_t is m_{2s}-strongly concave on Ω_{2s} and $M_{s,2}$-smooth on $\Omega_{s,2}$ for $t \in [T]$ and that the sparsity constraint $\mathcal{I}$ is an individual sparsity constraint. Let $(Z_1^*, \cdots, Z_T^*) \in \mathcal{I}$ be optimal supports of an optimal dictionary X^*. Then the solution $(Z_1, \cdots, Z_T) \in \mathcal{I}$ of Replacement Deletion-OMP after k' steps satisfies*

$$\sum_{t=1}^{T} f_t(Z_t) \geq \frac{m_{2s}^2}{M_{s,2}^2} \left(1 - \exp\left(-\frac{k'}{k} \frac{M_{s,2}}{m_{2s}} \right) \right) \sum_{t=1}^{T} f_t(Z_t^*).$$

Proof. The theorem follows from Lemmas 100 and 115. □

6.5.4 Fast Implementation for Average Sparsity Constraints

In general, $\mathcal{F}_a$ has $O(n^T)$ members, and therefore it is difficult to compute $\mathcal{F}_a$. Nevertheless, we show that Replacement OMP can run much faster for the examples presented in Section 6.4.

In Replacement Greedy, it is difficult to find an atom with the largest gain at each step. This is because we need to maximize a nonlinear function $\sum_{t=1}^T f_t(Z_t')$. Conversely, in Replacement OMP, if we can calculate $\mathbf{w}_t^{(Z_t)}$ and $\nabla u_t(\mathbf{w}_t^{(Z_t)})$ for all $t \in [T]$, the problem of calculating gain of each atom is reduced to maximizing a linear function.

In the following, we consider the variance reduction metric and average sparsity constraint because it is the most complex constraint. A similar result holds for the other examples. In fact, we show that this task reduces to maximum weighted bipartite matching. The Hungarian method returns the maximum weight bipartite matching in $\mathrm{O}(T^3)$ time. We can further improve the running time to $O(T \log T)$ time by utilizing the structure of this problem.

Next, we consider how to find the atom with the largest gain at each step of Replacement OMP for the average sparsity constraints.

First, we show that this task reduces to weighted bipartite matching. Let us fix an atom a^* because we can simply check all the atoms in V. Let $g_t = (\nabla u_t(\mathbf{w}^{(Z_t)}))_{a^*}^2$ and $c_t = \min_{a \in Z_t} (\mathbf{w}^{(Z_t)})_a^2$ for each $t \in [T]$. Let $S = \{t \in [T] \mid |Z_t| = s_t\}$ be the set of $t \in [T]$ such that the constraint on $|Z_t|$ is tight.

For each $a^* \in V$, the problem of finding the best replacement can be formulated as follows. The goal is to maximize $\sum_{t\in A} g_t - \sum_{t\in B} c_t$ by selecting $A \subseteq [T]$ (the set of indices t such that a^* is added to Z_t) and $B \subseteq [T]$ (the set of indices t such that an atom is removed from Z_t). We have two constraints on A and B. The first constraint is $|A| - |B| \le \theta$ where $\theta = s' - \sum_{t=1}^{T} |Z_t|$, derived from the total number constraint $\sum_{t=1}^{T} |Z_t| \le s'$. The second constraint is $A \cap S \subseteq B$, derived from the individual constraint $|Z_t| \le s_t$. In summary, the formulation as an optimization problem is:

$$\begin{aligned} \max_{A,B\subseteq[T]} \quad & \sum_{t\in A} g_t - \sum_{t\in B} c_t \\ \text{subject to} \quad & |A| - |B| \le \theta \\ & A \cap S \subseteq B. \end{aligned}$$

This problem can be regarded as a special case of maximum weight bipartite matching problem. Let $U = [T]$ and $V = [T] \cup \{d_1, \cdots, d_\theta\}$ be the set of vertices where $d_1, \cdots, d_\theta$ are dummy elements with zero cost, i.e., $c_{d_i} = 0$ for all $i \in [\theta]$. Let $E = \{(t,t) \mid t \in S\} \cup (U \setminus S) \times V$ be the set of edges. The weight of each edge $(\alpha, \beta) \in E$ is defined as $w((\alpha,\beta)) = g_\alpha - c_\beta$. Then any matching $M \subseteq E$ in this graph corresponds to a solution $A = \partial M \cap U$ and $B = \partial M \cap V \setminus \{d_1, \cdots, d_\theta\}$ in the above optimization problem.

Algorithm 17 Calculation of the gain for average sparsity constraints

Input $S = \{t \in [T] \mid |Z_t| = s_t\}$ the set of indices t such that Z_t is tight, $g_t = (\nabla u_t(\mathbf{w}^{(Z_t)}))^2_{a^*}$, $c_t = \min_{a\in Z_t} (\mathbf{w}^{(Z_t)})^2_a$ for each $t \in [T]$, and $\theta = s' - \sum_{t=1}^{T} |Z_t|$.
Output $A, B \subseteq [T]$ maximizing $\sum_{t\in A} g_t - \sum_{t\in B} c_t$ subject to $A \cap S \subseteq B$ and $|A| \le |B| + \theta$.

Initialize $A_0 \leftarrow \varnothing$ and $B_0 \leftarrow \varnothing$.
Let $S = \{t \in [T] \mid |Z_t| = s_t\}$.
Sort $t \in [T] \setminus S$ according to g_t into the priority queue Q_1 in descending order.
Sort $t \in [T]$ according to c_t into the priority queue Q_2 in ascending order.
Sort $t \in S$ according to $g_t - c_t$ into the priority queue Q_3 in descending order.
for $i = 1, \cdots, T$ **do**
  Let α, β and γ be the top elements in Q_1, Q_2, and Q_3, respectively.
  if $g_\alpha - c_\beta \mathbf{1}\{|A_{i-1}| = |B_{i-1}| + \theta\} \le 0$ and $g_\gamma - c_\gamma \le 0$ **then**
    return A_{i-1} and B_{i-1}
  else
    if $g_\alpha - c_\beta \mathbf{1}\{|A_{i-1}| = |B_{i-1}| + \theta\} \ge g_\gamma - c_\gamma$ **then**
      $A_i \leftarrow A_{i-1} + \alpha$ and remove α from Q_1.
      if $|A_{i-1}| = |B_{i-1}| + \theta$ **then**
        $B_i \leftarrow B_{i-1} + \beta$ and remove β from Q_2.
        if $\beta \in S$ **then**
          Remove β from Q_3 and add β to Q_1.
    else
      $A_i \leftarrow A_{i-1} + \gamma$ and $B_i \leftarrow B_{i-1} + \gamma$.
      Remove γ from Q_3.
return A_T and B_T

Here we give a fast greedy method for calculating the gain of each atom. This algorithm can be executed in $\mathrm{O}(T \log T)$ time. The detailed description of this algorithm is given in Algorithm 17.

Proposition 117. *Algorithm 17 returns an optimal solution in* $\mathrm{O}(T \log T)$ *time.*

Proof. First, we show the validity of the algorithm.

Before proving the optimality of the output, we note that the marginal gain of each step of the algorithm is largest among all the feasible updates. Let us consider the addition of α to A_{i-1}. There are three cases of updates. If $\alpha \in S \setminus B_{i-1}$ is added to A_{i-1}, we must also add α to B_{i-1}. If $\alpha \notin S \setminus B_{i-1}$ and $|A_{i-1}| = |B_{i-1}| + \theta$, adding $\beta \notin B_{i-1}$ with smallest cost c_t is the best choice. If $\alpha \notin S \setminus B_{i-1}$ and $|A_{i-1}| < |B_{i-1}| + \theta$, not changing B_{i-1} is the best choice. Algorithm 17 selects the best one from these cases.

We show (A_i, B_i) be optimal among feasible solutions such that $|A| = i$ by induction on i. It is clear that (A_0, B_0) is optimal among feasible solutions such that $|A| = 0$.

Now we assume (A_{i-1}, B_{i-1}) is optimal among feasible solutions such that $|A| = i - 1$. Let (A'_i, B'_i) be an optimal solution among feasible solutions such that $|A| = i$. If there exist $\alpha \in A'_i \setminus A_{i-1}$ and $\beta \in B'_i \setminus B_{i-1}$ such that $(A_{i-1} + \alpha, B_{i-1} + \beta)$ and $(A'_i - \alpha, B'_i - \beta)$ are both feasible, we obtain

$$\begin{aligned}
\sum_{t \in A_i} g_t - \sum_{t \in B_i} c_t &\geq \left(\sum_{t \in A_{i-1}} g_t - \sum_{t \in B_{i-1}} c_t \right) + (g_{\alpha_i} - c_{\beta_i}) \\
&\geq \left(\sum_{t \in A'_{i-1}} g_t - \sum_{t \in B'_{i-1}} c_t \right) + \left(g_{\alpha'} - c_{\beta'} \right) \\
&\geq \left(\sum_{t \in A'_i} g_t - \sum_{t \in B'_i} c_t \right),
\end{aligned}$$

which proves the optimality of (A_i, B_i). The second inequality is because the marginal gain of α_i (or possibly α_i and β_i) is largest among feasible additions. In the same way, if there exists $\alpha \in A'_i \setminus A_{i-1}$ such that $(A_{i-1} + \alpha, B_{i-1})$ and $(A'_i - \alpha, B'_i)$ are both feasible, then (A_i, B_i) is optimal.

We show the existence of such an α or pair (α, β). Since $|A'_i| > |A_{i-1}|$, we have $A'_i \setminus A_{i-1} \neq \varnothing$. Let $\alpha \in A'_i \setminus A_{i-1}$ be an arbitrary element. If $\alpha \in B'_i \setminus B_{i-1}$, the pair (α, α) satisfies the condition. If $\alpha \notin B'_i \setminus B_{i-1}$ and $|A_{i-1}| < |B_{i-1}| + \theta$, then α satisfies the condition. If $\alpha \notin B'_i \setminus B_{i-1}$ and $|A_{i-1}| = |B_{i-1}| + \theta$, we have $|B'_i| \geq |A'_i| - \theta > |B_{i-1}|$, then $B'_i \setminus B_{i-1} \neq \varnothing$. Therefore a pair of α and an arbitrary $\beta \in B'_i \setminus B_{i-1}$ satisfies the condition.

Finally we consider the running time of this algorithm. Sorting requires $\mathrm{O}(T \log T)$ time. Each iteration requires $\mathrm{O}(\log T)$ time. Thus, the total running time is $\mathrm{O}(T \log T)$. □

In summary, we obtain the following:

Theorem 118. *Assume that the assumption of Theorem 114 holds. Further assume that u is the variance reduction metric and $\mathcal{I}$ is the average sparsity constraint. Then Replacement OMP finds a solution $(Z_1, \cdots, Z_T) \in \mathcal{I}$ that satisfies*

$$\sum_{t=1}^{T} f_t(Z_t) \geq \left(\frac{\lambda_{\max}(\mathbf{A}, 2s)}{\lambda_{\min}(\mathbf{A}, 2)} \right)^2 \left(1 - \exp\left(-\frac{1}{3} \frac{\lambda_{\min}(\mathbf{A}, 2)}{\lambda_{\max}(\mathbf{A}, 2s)} \right) \right) \sum_{t=1}^{T} f_t(Z_t^*)$$

in $O(Tk(n \log T + ds))$ time.

Proof. In each iteration, we need to find an atom with the largest gain and the corresponding new supports $(Z'_1, \ldots, Z'_t)$. This can be done in $\mathrm{O}(nT \log T)$ time. Furthermore, we need to compute a new coefficient $\mathbf{w}_t^{(Z'_t)} = \mathbf{A}_{Z'_t}^{+} \mathbf{y}_t$ for the new support Z'_t ($t \in [T]$), where $\mathbf{A}^{+}$ is the pseudo inverse. This can be done efficiently via maintaining the QR-decomposition of $\mathbf{A}_{Z_t}$ under rank-two update [Golub and

Van Loan, 2012] with a cost of $O(s^2 + ds) = O(ds)$ time for each matrix. Thus each iteration requires $O(T(n \log T + ds))$ time, which proves the theorem. □

Remark 119. If finding an atom with the largest gain is computationally intractable, we can add an atom whose gain is no less than τ times the largest gain. In this case, we can bound the approximation ratio with replacing k' with $\tau k'$ in Theorems 112, 114 and 116.

6.6 Extensions to the Online Setting

Our algorithms can be extended to the following online setting. The problem is formulated as a two-player game between a player and an adversary. At each round $t = 1, \dots, T$, the player must select (possibly in a randomized manner) a dictionary $X_t \subseteq V$ with $|X_t| \leq k$. Then, the adversary reveals a data point $\mathbf{y}_t \in \mathbb{R}^d$ and the player gains with respect to the best s-sparse approximation to $\mathbf{y}_t$ with the selected dictionary X_t:

$$f_t(X_t) = \max_{Z_t \subseteq X_t \colon |Z_t| \leq s} \max_{\mathbf{w}_t \colon \operatorname{supp}(\mathbf{w}_t) \subseteq Z_t} u_t(\mathbf{w}_t),$$

where $u_t(\mathbf{w}_t)$ represents the quality of $\mathbf{w}_t$ for the approximation of $\mathbf{y}_t$. The performance measure of a player's strategy is the *expected α-regret*:

$$\operatorname{regret}_\alpha(T) = \alpha \max_{X^* : |X^*| \leq k} \sum_{t=1}^{T} f_t(X^*) - \mathbf{E}\left[\sum_{t=1}^{T} f_t(X_t)\right],$$

where $\alpha > 0$ is a constant independent from T corresponding to the offline approximation ratio, and the expectation is taken over the randomness in the player. Let $g_t(X) = \max_{Z \subseteq X \colon |Z| \leq s} f_t(Z)$ be the objective function at the tth round. In the following, we provide the online versions of algorithms for offline dictionary selection: Online SDS$_{\mathsf{MA}}$, Online Replacement Greedy, and Online Replacement OMP.

6.6.1 Online SDS$_{\mathsf{MA}}$

The first algorithm is based on SDS$_{\mathsf{MA}}$ for offline dictionary selection, which was proposed by Krause and Cevher [2010] and given an improved analysis by Das and Kempe [2011]. At each round t, we consider a function $\tilde{f}_t(Z) = \sum_{a \in Z} f_t(a|\varnothing)$, which is a modular approximation of f_t. Intuitively, the modular approximation $\tilde{f}_t$ ignores the interactions among the atoms. We define the surrogate objective $\tilde{g}_t$ as

$$\tilde{g}_t(X) = \max_{Z \subseteq X \colon |Z| \leq s} \tilde{f}_t(Z). \tag{6.17}$$

It is easy to show that $\tilde{g}_t$ is monotone submodular. Hence, we can apply the online greedy algorithm [Streeter and Golovin, 2008] to these surrogate functions.

Assuming the strong concavity and smoothness of u_t, the original objective function g_t can be bounded from lower and upper with the surrogate function $\tilde{g}_t$. A similar result is given in Elenberg et al. [2018] for offline sparse regression.

Lemma 120. *Suppose u_t is m_1-strongly concave and M_1-smooth on Ω_1, and m_s-strongly concave and M_s-smooth on Ω_s. Then,*

$$\frac{m_1}{M_s} \tilde{g}_t(X) \leq g_t(X) \leq \frac{M_1}{m_s} \tilde{g}_t(X).$$

Proof. Let $Z \subseteq V$ be an arbitrary subset such that $|Z| \leq s$. Since the submodularity ratio $\gamma_{\varnothing,s}$ of f is no less than m_s/M_1 [Elenberg et al., 2018],

$$\frac{m_s}{M_1} f_t(Z) \leq \sum_{a \in Z} \tilde{f}_t(a) = \tilde{f}_t(Z).$$

As this bound holds for any $Z \subseteq V$ of size no more than s, we have

$$g_t(X) = \max_{Z \subseteq X \colon |Z| \leq s} f_t(Z) \leq \frac{M_1}{m_s} \max_{Z \subseteq X \colon |Z| \leq s} \tilde{f}_t(Z) = \frac{M_1}{m_s} \tilde{g}_t(X).$$

Next, we prove the lower bound of $g_t(X)$. From the optimality of $\mathbf{w}^{(Z)}$, for any $\mathbf{z}$ such that $\operatorname{supp}(\mathbf{z}) \subseteq Z$,

$$\begin{aligned} f_t(Z) &= u_t(\mathbf{w}^{(Z)}) - u_t(\mathbf{0}) \\ &\geq u_t(\mathbf{z}) - u_t(\mathbf{0}) \\ &\geq \langle \nabla u_t(\mathbf{0}), \mathbf{z} \rangle - \frac{M_s}{2} \|\mathbf{z}\|^2 \end{aligned}$$

where the last inequality is due to the strong concavity of u_t. Using $\mathbf{z} = \frac{1}{M_s} (\nabla u_t(\mathbf{0}))_Z$, we obtain

$$f_t(Z) \geq \frac{1}{2M_s} \|(\nabla u_t(\mathbf{0}))_Z\|^2. \tag{6.18}$$

On the other hand, from the smoothness of u_t, we have for all $a \in Z$,

$$\begin{aligned} f_t(a) &= u_t(\mathbf{w}^{(a)}) - u_t(\mathbf{0}) \\ &\leq \langle \nabla u_t(\mathbf{0}), \mathbf{w}^{(a)} \rangle - \frac{m_1}{2} \|\mathbf{w}^{(a)}\|^2 \\ &\leq \max_{c \in \mathbb{R}} \langle \nabla u_t(\mathbf{0}), c\mathbf{e}_a \rangle - \frac{m_1}{2} \|c\mathbf{e}_a\|^2 \\ &= \frac{1}{2m_1} (\nabla u_t(\mathbf{0}))_a^2. \end{aligned}$$

Summing up for all $a \in Z$, we obtain

$$\tilde{f}_t(Z) = \sum_{a \in Z} f_t(a) \leq \frac{1}{2m_1} \|(\nabla u_t(\mathbf{0}))_Z\|^2. \tag{6.19}$$

Combining (6.18) and (6.19), we obtain the lower bound

$$f_t(Z) \geq \frac{m_1}{M_s} \tilde{f}_t(Z),$$

which proves the lower bound of $g_t(X)$ in the same way as the upper bound. □

The expected regret of this algorithm can be bounded as follows.

Theorem 121. *Let $\alpha = (1 - \frac{1}{\mathrm{e}}) \frac{m_1 m_s}{M_1 M_s}$. The expected α-regret of the modular approximation algorithm after T rounds is bounded as follows.*

$$\operatorname{regret}_\alpha(T) \leq \frac{k \Delta_{\max} m_1}{M_s} \sqrt{2T \ln n}$$

where $n = |V|$ and $\Delta_{\max} = \max_{a \in V} \max_{t \in [T]} f_t(a|\varnothing)$.

Proof. Applying the regret bound for online submodular maximization [Streeter and Golovin, 2008], we obtain

$$\left(1 - \frac{1}{\mathrm{e}}\right) \sum_{t=1}^{T} \tilde{g}_t(X^*) - \sum_{t=1}^{T} \tilde{g}_t(X_t) \leq k\Delta_{\max}\sqrt{2T \ln n}. \tag{6.20}$$

since the gains for each subroutine are bounded by $\Delta_{\max}$. From Lemma 120, we obtain the bound in the statement. □

In the case of $u_t(\mathbf{w}) = \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2$, α is equal to an approximation ratio shown in Das and Kempe [2011].

Corollary 122. *For the variance reduction metric, the expected regret of the modular approximation algorithm is*

$$\mathrm{regret}_\alpha(T) \leq \frac{k\Delta_{\max}}{\lambda_{\max}(\mathbf{A}, s)}\sqrt{2T \ln n},$$

where $\alpha = (1 - \frac{1}{\mathrm{e}})\frac{\lambda_{\min}(\mathbf{A},s)}{\lambda_{\max}(\mathbf{A},s)}$.

6.6.2 Online Replacement Greedy

In the following, we provide online adaptation of Replacement Greedy. Similarly to Streeter and Golovin [2008], we use k expert algorithms $\mathcal{A}^1, \cdots, \mathcal{A}^k$ as subroutines. At each round, online Replacement Greedy selects a set of k elements $a_t^1, \cdots, a_t^k$ according to the expert algorithms $\mathcal{A}^1, \cdots, \mathcal{A}^k$, respectively. After the target point $\mathbf{y}_t$ is revealed, the algorithm decides the feedback to the subroutines by considering how Z_t changes if $a_t^1, \cdots, a_t^k$ are added to X sequentially. As in the offline version of Replacement Greedy, we start with $Z_t^0 = \varnothing$ and consider adding a_t^i to Z_t or not with keeping $|Z_t| \leq s$ for each $i = 1, \cdots, k$. Denoting Z_t at the ith step by Z_t^i, we can write the feedback given to the subroutine $\mathcal{A}^i$ as $\Delta_t(\cdot, Z_t^{i-1})$ where

$$\Delta_t(a, Z_t^i) = \begin{cases} f_t(Z_t^i + a) - f_t(Z_t^i) & (i < s) \\ \max\left\{0, \max\limits_{a' \in Z_t^i} \left\{f_t(Z_t^i - a' + a) - f_t(Z_t^i)\right\}\right\} & (i \geq s) \end{cases}$$

is the gain obtained by adding a to Z_t^i. If $\Delta_t(a_t^i, Z_t^{i-1}) > 0$, the algorithm updates Z_t by adding a_t^i and, if $i > s$, removing a' that maximizing $f_t(Z_t^{i-1} - a' + a_t^i)$. For each $a \in V$, the value of gain $\Delta_t(a, Z_t^{i-1})$ is given to $\mathcal{A}^i$ as the feedback about a. A pseudocode description of our algorithm is shown in Algorithm 18.

Theorem 123. *Assume that u_t is m_{2s}-strongly concave on Ω_{2s} and $M_{s,2}$-smooth on $\Omega_{s,2}$ for $t \in [T]$. Then the online replacement greedy algorithm achieves the regret bound* $\mathrm{regret}_\alpha(T) \leq \sum_{i=1}^k r_i$, *where r_i is the regret of the online greedy selection subroutine $\mathcal{A}^i$ for $i \in [k]$ and*

$$\alpha = \left(\frac{m_{2s}}{M_{s,2}}\right)^2 \left(1 - \exp\left(-\frac{M_{s,2}}{m_{2s}}\right)\right).$$

In particular, if we use the hedge algorithm as the online greedy selection subroutines, we obtain $\mathrm{regret}_\alpha(T) \leq k\sqrt{2T \ln n}$.

Corollary 124. *For the variance reduction metric,*

$$\alpha \geq \left(\frac{\lambda_{\min}(\mathbf{A}, 2s)}{\lambda_{\max}(\mathbf{A}, 2)}\right)^2 \left(1 - \exp\left(-\frac{\lambda_{\max}(\mathbf{A}, 2)}{\lambda_{\min}(\mathbf{A}, 2s)}\right)\right).$$

Proof of Theorem 123. We provide a lower bound on the sum of the ith step marginal gains of the algorithm. Let Z_t^* be an optimal sparse subset of X^* for f_t, i.e., $Z_t^* \in \operatorname{argmax}_{Z \subseteq X^* \colon |Z| \leq s} f_t(Z)$. Then we have

$$\begin{aligned}
\sum_{t=1}^{T} \Delta_t(a_t^i, Z_t^{i-1}) &\geq \max_{x \in V} \sum_{t=1}^{T} \Delta_t(x, Z_t^{i-1}) - r_i \\
&\geq \frac{1}{k} \sum_{a \in X^*} \sum_{t=1}^{T} \Delta_t(a, Z_t^{i-1}) - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \sum_{a \in Z_t^*} \Delta_t(a, Z_t^{i-1}) - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \left(C_1 f_t(Z_i^*) - C_2 f_t(Z_t^{i-1}) \right) - r_i
\end{aligned} \tag{6.21}$$

where $C_1 = \frac{m_{2s}}{M_{s,2}}$ and $C_2 = \frac{M_{s,2}}{m_{2s}}$. The first inequality is due to the regret bound for the subroutine $\mathcal{A}^i$. The last inequality is due to Lemma 111. Now the theorem directly follows from Lemma 100. □

6.6.3 Online Replacement OMP

In this section, we consider an online version of Replacement OMP. This algorithm is the same as Online Replacement Greedy except the gain at each step. The gain obtained when a is added to Z_t^i is

$$\frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t^i)}) \right)_a^2$$

when $i < s$, and

$$\max \left\{ 0, \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t^i)}) \right)_a^2 - \min_{a' \in Z_t^i} \frac{M_{s,2}}{2} \left(\mathbf{w}_t^{(Z_t^i)} \right)_{a'}^2 \right\}$$

when $i \geq s$, where $\mathbf{w}_t^{(Z_t^i)} \in \operatorname{argmax}_{\mathbf{w} : \operatorname{supp}(\mathbf{w}) \subseteq Z_t^i} u_t(\mathbf{w})$.

Theorem 125. *Assume that u_t is m_{2s}-strongly concave on Ω_{2s} and $M_{s,2}$-smooth on $\Omega_{s,2}$ for $t \in [T]$. Then Online Replacement OMP algorithm achieves the regret bound $\operatorname{regret}_\alpha(T) \leq \sum_{i=1}^k r_i$, where r_i is the regret of the online greedy selection subroutine $\mathcal{A}^i$ for $i \in [k]$ and*

$$\alpha = \left(\frac{m_{2s}}{M_{s,2}} \right)^2 \left(1 - \exp\left(-\frac{M_{s,2}}{m_{2s}} \right) \right).$$

In particular, if we use the hedge algorithm as the online greedy selection subroutines, we obtain $\operatorname{regret}_\alpha(T) \leq k\sqrt{2T \ln n}$.

Proof. Since f_t is $M_{s,2}$-smooth on $\Omega_{s,2}$, it holds that for any $a, a' \in V$ and $Z_t \subseteq V$ of size at most s,

$$f_t(Z_t - a' + a) - f_t(Z_t) \geq \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_a^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}_t^{(Z_t)} \right)_{a'}^2.$$

In addition, we have

$$\frac{1}{2M_{s,2}} \| (\nabla u_t(\mathbf{w}^{(Z)}))_{X \setminus Z} \|^2 - \frac{M_{s,2}}{2} \| (\mathbf{w}^{(Z)})_{Z \setminus X} \|^2 \leq \frac{m_{2s}}{M_{s,2}} f(X) - \frac{M_{s,2}}{m_{s2}} f(Z)$$

from the proof of Lemma 111.

We provide a lower bound on the ith step marginal gain of the algorithm. Let Z_t^* be an optimal sparse subset of X^* for f_t, i.e., $Z_t^* \in \operatorname{argmax}_{Z \subseteq X^* \colon |Z| \leq s} f_t(Z)$. If $i \leq s$, then $|Z_t^{i-1}| < s$ holds for all t. Then we have

$$\begin{aligned}
\sum_{t=1}^{T} \Delta_t(a_t^i, Z_t^{i-1}) &= \sum_{t=1}^{T} \left\{ f_t(Z_t^{i-1} + a_t^i) - f_t(Z_t^{i-1}) \right\} \\
&\geq \sum_{t=1}^{T} \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a_t^i}^2 \\
&\geq \max_{a^i \in V} \sum_{t=1}^{T} \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a^i}^2 - r_i \\
&\geq \frac{1}{k} \sum_{a \in X^*} \sum_{t=1}^{T} \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a}^2 - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \sum_{a \in Z_t^* \setminus Z_t} \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a}^2 - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \frac{1}{2M_{s,2}} \left\| \nabla u_t(\mathbf{w}_t^{(Z_t)}) \right\|_{Z_t^* \setminus Z_t}^2 - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \left(\frac{m_{s,2}}{M_{s,2}} f_t(Z_i^*) - \frac{M_{s,2}}{m_{2s}} f_t(Z_t^{i-1}) \right) - r_i.
\end{aligned}$$

Otherwise, $|Z_t^{i-1}| = s$ holds for all t, therefore

$$\begin{aligned}
\sum_{t=1}^{T} \Delta_t(a_t^i, Z_t^{i-1}) &\geq \sum_{t=1}^{T} \max \left\{ 0, \max_{a_t' \in Z_t} \left\{ f_t(Z_t - a_t' + a_t^i) - f_t(Z_t) \right\} \right\} \\
&\geq \sum_{t=1}^{T} \max \left\{ 0, \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a_t^i}^2 - \min_{a_t' \in Z_t} \frac{M_{s,2}}{2} \left(\mathbf{w}_t^{(Z_t)} \right)_{a_t'}^2 \right\} \\
&\geq \max_{a^i \in V} \sum_{t=1}^{T} \max \left\{ 0, \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a^i}^2 - \min_{a_t' \in Z_t} \frac{M_{s,2}}{2} \left(\mathbf{w}_t^{(Z_t)} \right)_{a_t'}^2 \right\} - r_i \\
&\geq \frac{1}{k} \sum_{a \in X^*} \sum_{t=1}^{T} \max \left\{ 0, \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a}^2 - \min_{a_t' \in Z_t} \frac{M_{s,2}}{2} \left(\mathbf{w}_t^{(Z_t)} \right)_{a_t'}^2 \right\} - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \sum_{a \in Z_t^* \setminus Z_t} \left\{ \frac{1}{2M_{s,2}} \left(\nabla u_t(\mathbf{w}_t^{(Z_t)}) \right)_{a}^2 - \frac{M_{s,2}}{2} \left(\mathbf{w}_t^{(Z_t)} \right)_{\pi_t(a)}^2 \right\} - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \left\{ \frac{1}{2M_{s,2}} \left\| (\nabla u_t(\mathbf{w}_t^{(Z_t)}))_{Z_t^* \setminus Z_t} \right\|^2 - \frac{M_{s,2}}{2} \left\| (\mathbf{w}_t^{(Z_t)})_{Z_t \setminus Z_t^*} \right\|^2 \right\} - r_i \\
&\geq \frac{1}{k} \sum_{t=1}^{T} \left(\frac{m_{2s}}{M_{s,2}} f_t(Z_i^*) - \frac{M_{s,2}}{m_{2s}} f_t(Z_t^{i-1}) \right) - r_i. \qquad (6.22)
\end{aligned}$$

where a map $\pi_t \colon Z_t^* \setminus Z_t \to Z_t \setminus Z_t^*$ is an arbitrary bijection for each t.

Combining with Lemma 100, we obtain the theorem. □

Algorithm 18 Online Replacement Greedy & Online Replacement OMP

Initialize online greedy selection subroutines $\mathcal{A}^i$ for $i = 1, \ldots, k$.
for $t = 1, \ldots, T$ **do**
  Initialize $X_t^0 \leftarrow \varnothing$ and $Z_t^0 \leftarrow \varnothing$ for all $t \in [T]$.
  for $i = 1, \ldots, k$ **do**
    Pick $a_t^i \in V$ according to $\mathcal{A}^i$.
    Set $X_t^i \leftarrow X_t^{i-1} + a_t^i$.
  Play X_t^k and observe $\mathbf{y}_t$.
  for $i = 1, \ldots, k$ **do**
    To the subroutine $\mathcal{A}_i$, feed the gain of a defined as

- $\Delta_t(a, Z_t^{i-1})$ (Online Replacement Greedy)
- $$\begin{cases} \frac{1}{M_{s,2}} \left(\nabla u_t \left(\mathbf{w}_t^{(Z_t)} \right) \right)_a^2 & \text{if } i \leq s, \\ \max \left\{ 0, \frac{1}{M_{s,2}} \left(\nabla u_t (\mathbf{w}_t^{(Z_t^{i-1})}) \right)_a^2 - M_{s,2} \min_{a_t' \in Z_t^{i-1}} \left(\mathbf{w}_t^{(Z_t^{i-1})} \right)_{a_t'}^2 \right\} & \text{otherwise} \end{cases}$$
  (Online Replacement OMP)

    Do the optimal replacement of Z_t^{i-1} with respect to a_t^i that achieves the above gain for Replacement Greedy or Replacement OMP, and obtain Z_t^i.

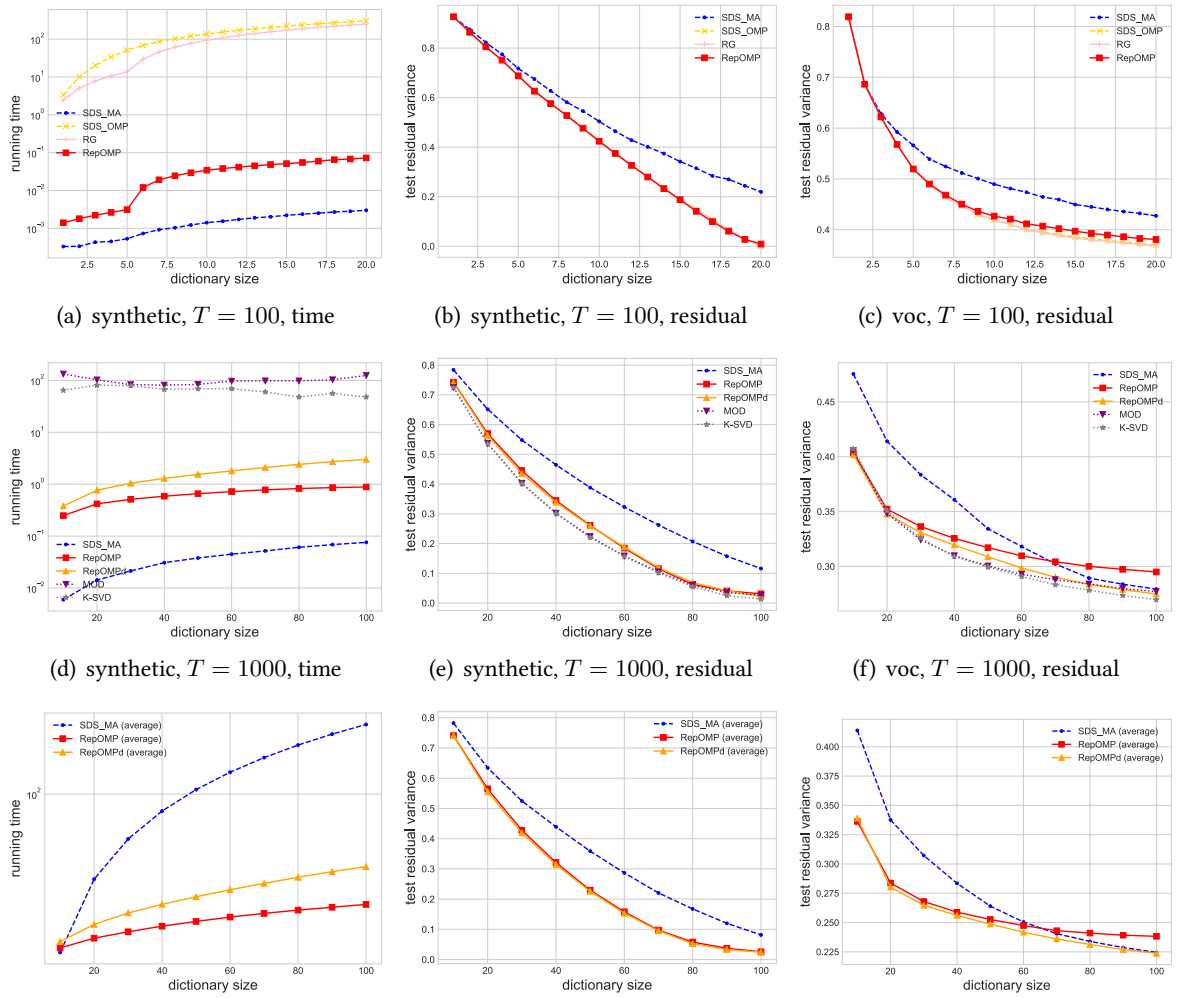
6.7 Experiments

In this section, we empirically evaluate our proposed algorithms on several dictionary selection problems with synthetic and real-world datasets. We use the variance reduction metric for all of the experiments. Since evaluating the value of the objective function is NP-hard, we plot the approximated residual variance obtained by orthogonal matching pursuit.

Ground set We use the ground set consisting of several orthonormal bases that are standard choices in signal and image processing, such as 2D discrete cosine transform and several 2D discrete wavelet transforms (Haar, Daubechies 4, and coiflet). In all of the experiments, the dimension is set to $d = 64$, which corresponds to images of size 8×8 pixels. The size of the ground set is $n = 256$.

Machine All the algorithms are implemented in Python 3.6. We conduct the experiments in a machine with Intel Xeon E3-1225 V2 (3.20 GHz and 4 cores) and 16 GB RAM.

Datasets We conduct experiments on two types of datasets. The first one is a synthetic dataset. In each trial, we randomly pick a dictionary with size k out of the ground set, and generate sparse linear combinations of the columns of this dictionary. The weights of the linear combinations are generated from the standard normal distribution. The second one is a dataset of real-world images extracted from PASCAL VOC2006 image datasets [Everingham et al., 2006]. In each trial, we randomly select an image out of 2618 images and divide it into patches of 8×8 pixels, then select T patches uniformly at random. All the patches are normalized to zero mean and unit variance. We make datasets for training and test in the same way, and use the training dataset for obtaining a dictionary and the test dataset for measuring the quality of the output dictionary.



(a) synthetic, $T = 100$, time (b) synthetic, $T = 100$, residual (c) voc, $T = 100$, residual

(d) synthetic, $T = 1000$, time (e) synthetic, $T = 1000$, residual (f) voc, $T = 1000$, residual

(g) synthetic, $T = 1000$, time (h) synthetic, $T = 1000$, residual (i) voc, $T = 1000$, residual

Figure 6.1: The experimental results for the offline setting. In all figures, the horizontal axis indicates the size of the output dictionary. (a), (b), and (c) are the results for $T = 100$. (d), (e), and (f) are the results for $T = 1000$. (g), (h), and (i) are the results for $T = 1000$ with an average sparsity constraint. For each setting, we provide the plot of the running time for the synthetic dataset, test residual variance for the synthetic dataset, and test residual variance for VOC2006 image dataset.

6.7.1 Experiments on the Offline Setting

We implement our proposed methods, Replacement Greedy (RG) and Replacement OMP (RepOMP), as well as the existing methods for dictionary selection, SDS_{MA} and SDS_{OMP}. We also implement a heuristically modified version of RepOMP, which we call RepOMPd. In RepOMPd, we replace $M_{s,2}$ with some parameter that decreases as the size of the current dictionary grows, which prevents the gains of all the atoms from being zero. Here we use $M_{s,2}/\sqrt{i}$ as the decreasing parameter where i is the number of iterations so far. In addition, we compare these methods with standard methods for dictionary learning, MOD [Engan et al., 1999] and KSVD [Aharon et al., 2006], which is set to stop when the change of the objective value becomes no more than 10^{-6} or 200 iterations are finished. Orthogonal matching pursuit is used as a subroutine in both methods.

First, we compare the methods for dictionary selection with small datasets of $T = 100$. The parameter of sparsity constraints is set to $s = 5$. The results averaged over 20 trials are shown in Figure 6.1(a), (b), and (c). The plot of the running time for VOC2006 datasets is omitted as it is much similar to that for synthetic datasets. In terms of running time, SDS_{MA} is the fastest, but the quality of the output dictionary is unsatisfactory. RepOMP is several magnitudes faster than SDS_{OMP} and RG, but its quality is almost the same with SDS_{OMP} and RG. In Figure 6.1(b), test residual variance of SDS_{OMP}, RG, and RepOMP are overlapped, and in Figure 6.1(c), test residual variance of RepOMP is slightly worse than that of SDS_{OMP} and RG. From these results, we can conclude that RepOMP is by far the most practical method for dictionary selection.

Next, we compare the dictionary selection methods with the dictionary learning methods with larger datasets of $T = 1000$. SDS_{OMP} and RG are omitted because they are too slow to be applied to datasets of this size. The results averaged over 20 trials are shown in Figure 6.1(d), (e), and (f). In terms of running time, RepOMP and RepOMPd are much faster than MOD and KSVD, but their performances are competitive with MOD and KSVD.

Finally, we conduct experiments with the average sparsity constraints. We compare RepOMP and RepOMPd with Algorithm 17 with a variant of SDS_{MA} proposed for average sparsity in Cevher and Krause [2011]. The parameters of constraints are set to $s_t = 8$ for all $t \in [T]$ and $s' = 5T$. The results averaged over 20 trials are shown in Figure 6.1(g), (h), and (i). RepOMP and RepOMPd outperform SDS_{MA} both in running time and quality of the output.

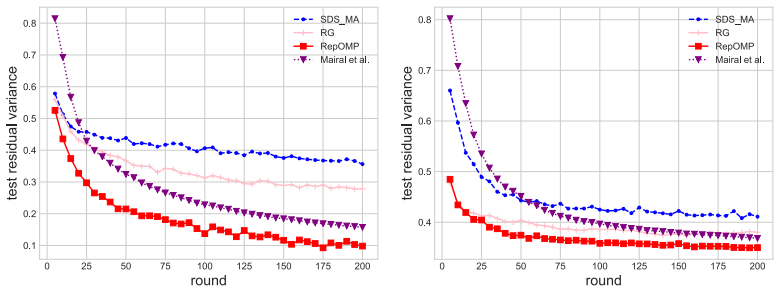
In Section 6.7.3, We provide further experimental results. There we provide examples of image restoration, in which the average sparsity works better than the standard dictionary selection.

6.7.2 Experiments on the Online Setting

Here we give the experimental results on the online setting. We implement the online version of SDS_{MA}, RG and RepOMP, as well as an online dictionary learning algorithm proposed by Mairal et al. [2010]. For all the online dictionary selection methods, the hedge algorithm is used as the subroutines. The parameters are set to $k = 20$ and $s = 5$. The results averaged over 50 trials are shown in Figure 6.2(a), (b). For both datasets, Online RepOMP shows a better performance than Online SDS_{MA}, Online RG, and the online dictionary learning algorithm.

6.7.3 Experiments on Dimensionality Reduced Data

In this section, we conduct experiments on the task called *image restoration*. In this task, we are given an incomplete image, that is, a portion of its pixels are missing. First, we divide this incomplete image into small patches of 8×8 pixels. Then we regard each of these patches as a data point $\mathbf{y}_t$, and aim to select a dictionary that yields a sparse representation of these patches. In the procedure of the algorithms, the loss is evaluated only on the given pixels. Finally, we restore the original image by replacing each patch



(a) synthetic (b) voc

Figure 6.2: The experimental results for the online setting. In both figures, the horizontal axis indicates the number of rounds. (a) is the result with synthetic datasets, and (b) is the result with VOC2006 image datasets.

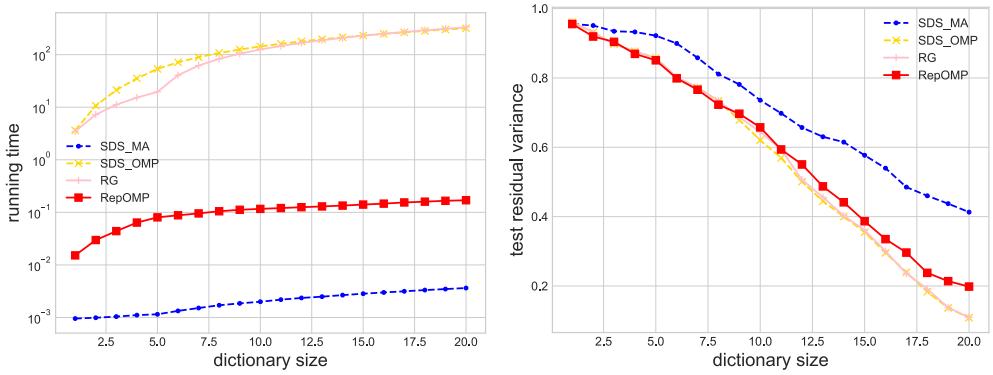
with a sparse approximation using the selected dictionaries, and the loss is evaluated on the whole pixels.

First, we conduct experiments with synthetic datasets to investigate the behavior of the algorithms. For each of the training and test datasets, we generate a bit mask such that each value takes 0 or 1 with equal probability. We give the masked training dataset to the algorithms and let them learn a dictionary. With this dictionary, we create the sparse representation of each data point in the test dataset with only unmasked elements and evaluate its residual variance with the whole elements. Figure 6.3(a) and 6.3(b) are the results for smaller datasets of $T = 100$, and Figure 6.3(c) and 6.3(d) are the results for larger datasets of $T = 1000$. In both experiments, we can see the relationship of the algorithms' performance is similar to the one in the non-masked settings, Figure 6.1(a), 6.1(b), 6.1(d), and 6.1(e).

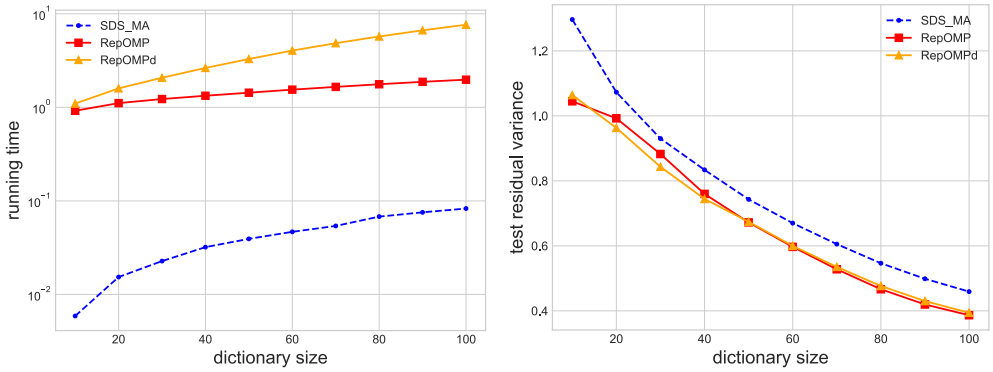
In order to illustrate the advantage of the average sparsity to ordinary dictionary learning (the individual sparsity), we give image restoration examples with real-world images. We use Replacement OMP for both the individual sparsity and the average sparsity. With setting $s_t = s$ for all $t \in [T]$, the parameters k, s, and s' are determined with the grid search. We apply Replacement OMP to incomplete images and obtain a dictionary. Then with this dictionary, we repeatedly compute the sparse representation of patches in the input image while shifting a single pixel. OMP is used for obtaining the sparse representation. When calculating the coefficients of the sparse representation of each patch, we use only the observed pixels and restore the whole pixels with these coefficients. We take the median value of all the restored patches for each pixel. In Figure 6.4, the input image, the image restored with the individual sparsity, and the image restored with the average sparsity are shown with PSNR ratios. The method with the average sparsity obtains higher PSNR ratios than one with the individual sparsity for all the images.

6.8 Summary and Future Work

In this chapter, we developed greedy algorithms for dictionary selection. We generalized the problem setting of dictionary selection by introducing p-replacement sparsity families and showed that existing sparsity constraints can be regarded as special cases of these families. Next we developed Replacement Greedy, Replacement OMP, and Replacement Deletion-OMP and gave lower bounds on their approximation ratios. We provided fast implementation of Replacement OMP for average sparsity constraints. We formulated the problem setting of online dictionary selection and proposed Online Replacement Greedy, Online Replacement OMP, and Online SDS_{MA}. Finally, we conducted experiments for synthetic and real datasets to show the practical efficiency of Replacement OMP.



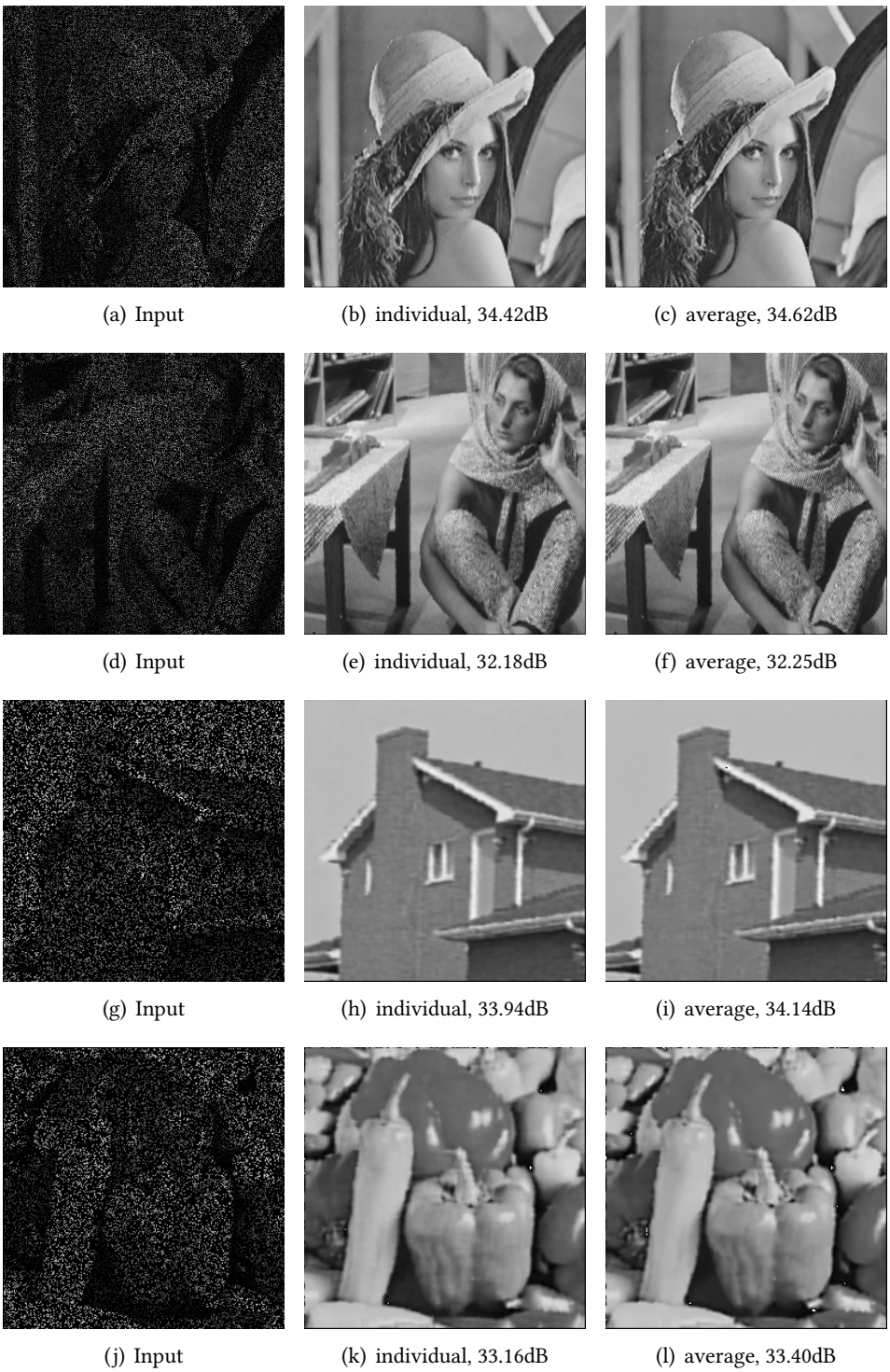
(a) $T = 100$, time (b) $T = 100$, residual



(c) $T = 1000$, time (d) $T = 1000$, residual

Figure 6.3: The experimental results for dimensionality reduced synthetic datasets. In all figures, the horizontal axis indicates the size of the output dictionary. (a) and (b) are the results for $T = 100$. (c) and (d) are the results for $T = 1000$. For each setting, we give the plot of the running time and the test residual variance.

An interesting direction for future work is to consider how to decide the ground set for dictionary selection. The ground set for dictionary selection is usually made from existing domain-specific dictionaries such as DCT or wavelets. The quality of dictionaries returned by algorithms heavily depends on how well the ground set fits into the given dataset. Hence it is vital to consider how we can decide a better ground set for the given dataset.



(a) Input (b) individual, 34.42dB (c) average, 34.62dB

(d) Input (e) individual, 32.18dB (f) average, 32.25dB

(g) Input (h) individual, 33.94dB (i) average, 34.14dB

(j) Input (k) individual, 33.16dB (l) average, 33.40dB

Figure 6.4: The results of the image restoration experiment from images with 80% of pixels missing.

Conclusion

In this dissertation, we have proposed two new notions of approximate submodularity and developed efficient algorithms for various machine learning problems based on them.

The first notion is approximate submodularity for adaptive optimization. In Chapter 3, we defined the adaptive submodularity ratio, which represents the closeness of an objective function to the adaptive submodular functions. We showed that the adaptive submodularity ratio is bounded for adaptive influence maximization in bipartite graphs and adaptive feature selection. As we showed, if the adaptive submodularity ratio is bounded, we can provide a theoretical guarantee for the adaptive greedy algorithm and a lower bound of the adaptivity gap. In Chapter 4, we extended the result on the adaptive submodularity ratio to the batch-mode setting, through which we required to refine the existing framework for the batch-mode setting. We defined set-adaptive submodularity, which is a strictly stronger property than adaptive submodularity, and showed that this property is satisfied in many important applications. We also dealt with the batch-mode setting where the set of feasible batches at each step is determined under a combinatorial constraint.

The second notion is approximate submodularity for local search. In Chapter 5, we showed that the objective function of feature selection satisfies an approximate submodularity that can be utilized for analyzing local search algorithms. We proposed several variants of local search algorithms for a matroid constraint, p-matroid intersection constraint, or p-exchange system constraint. We conduct experiments on sparse linear regression and structure learning of graphical models to validate the practical effectiveness of the proposed algorithms. In Chapter 6, we studied the two-stage version of feature selection with strong motivation for applying it to dictionary selection. We proposed Replacement OMP, which is obtained by accelerating Replacement Greedy. We also proposed the class of p-replacement sparsity constraints, which enables us to utilize prior knowledge about relationships among variables.

Through this dissertation, we have seen that the approach based on approximate submodularity is effective for analyzing and devising practical algorithms for machine learning problems. Applications of the proposed frameworks range broadly from viral marketing to dictionary selection, but we believe there are still many possible applications of this approach.

A possible direction of future research is to develop a framework for minimization of approximately submodular functions. Submodular minimization has been applied to various machine learning problems such as regularization [Bach, 2013], clustering [Nagano et al., 2010], and image segmentation [Jegelka and Bilmes, 2011]. While maximization of approximately submodular functions has been studied extensively as described in this dissertation, there are only a few studies on the minimization side. Minimizing approximately submodular functions might yield novel efficient algorithms for regularization that does not satisfy submodularity.

Another direction is to apply the approach of approximate submodularity to problems in algorithmic game theory. It is known that utility functions about substitutable goods, such as tomatoes from different farms, are naturally expressed as submodular functions [Lehmann et al., 2006]. If there exist complementary goods, such as left and right shoes, submodularity does not hold. The frameworks of approximate submodularity developed for machine learning problems might be applied to such a game-theoretic scenario where both substitutability and complementarity exist.

Bibliography

Yahoo! webscope dataset: G1 - Yahoo! Search Marketing Advertiser-Phrase Bipartite Graph, Version 1.0. URL https://webscope.sandbox.yahoo.com/.

Alekh Agarwal, Animashree Anandkumar, Prateek Jain, and Praneeth Netrapalli. Learning sparsely used overcomplete dictionaries via alternating minimization. *SIAM Journal on Optimization*, 26(4): 2775–2799, 2016.

Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

Noga Alon, Iftah Gamzu, and Moshe Tennenholtz. Optimizing budget allocation among channels and influencers. In *Proceedings of the 21st World Wide Web Conference (WWW)*, pages 381–388, 2012.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Sanjeev Arora, Rong Ge, and Ankur Moitra. New algorithms for learning incoherent and overcomplete dictionaries. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 779–806, 2014.

Francis R. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

Francis R. Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 671–680, 2014.

Sohail Bahmani, Bhiksha Raj, and Petros T. Boufounos. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(1):807–841, 2013.

Eric Balkanski, Baharan Mirzasoleiman, Andreas Krause, and Yaron Singer. Learning sparse combinatorial representations via two-stage submodular maximization. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 2207–2216, 2016.

MohammadHossein Bateni, Mohammad Taghi Hajiaghayi, and Morteza Zadimoghaddam. Submodular secretary problem and extensions. *ACM Transactions on Algorithms*, 9(4):32:1–32:23, 2013.

Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D*, 24 (3):179–195, 1975.

Andrew An Bian, Joachim M. Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 498–507, 2017.

Ilija Bogunovic, Junyao Zhao, and Volkan Cevher. Robust maximization of non-submodular objectives. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 890–899, 2018.

Guy Bresler. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC)*, pages 771–782, 2015.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.

Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal Computing*, 40(6):1740–1766, 2011.

Emmanuel J. Candes and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Volkan Cevher and Andreas Krause. Greedy dictionary selection for sparse representation. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):979–988, 2011.

Lin Chen, Moran Feldman, and Amin Karbasi. Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy? In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 803–812, 2018a.

Lin Chen, Hamed Hassani, and Amin Karbasi. Online continuous submodular maximization. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1896–1905, 2018b.

Ning Chen, Nicole Immorlica, Anna R. Karlin, Mohammad Mahdian, and Atri Rudra. Approximating matches made in heaven. In *Proceedings of the 36th International Colloquium of Automata, Languages and Programming (ICALP)*, pages 266–278, 2009.

Yuxin Chen and Andreas Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *Proceedings of the Thirtieth International Conference on Machine Learning (ICML)*, pages 160–168, 2013.

Yuxin Chen, Hiroaki Shioi, Cesar Fuentes Montesinos, Lian Pin Koh, Serge Wich, and Andreas Krause. Active detection via adaptive submodularity. In *Proceedings of the 31th International Conference on Machine Learning, (ICML)*, pages 55–63, 2014.

Yuxin Chen, S. Hamed Hassani, Amin Karbasi, and Andreas Krause. Sequential information maximization: When is greedy near-optimal? In *Proceedings of the 28th Conference on Learning Theory (COLT)*, pages 338–363, 2015.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia*, 14(1):66–75, 2012.

Yang Cong, Ji Liu, Gan Sun, Quanzeng You, Yuncheng Li, and Jiebo Luo. Adaptive greedy dictionary selection for web media summarization. *IEEE Transactions on Image Processing*, 26(1):185–195, 2017.

Gerard Cornuejols, Marshall L. Fisher, and George L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1057–1064, 2011.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Advances in Neural Information Processing Systems (NIPS) 17*, pages 337–344, 2004.

Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.

Bogdan Dumitrescu and Paul Irofti. *Dictionary Learning Algorithms and Applications*. Springer, 2018.

Ethan R. Elenberg, Rajiv Khanna, Alexandros G. Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *Annals of Statistics*, 46(6B):3539–3568, 2018.

Kjersti Engan, Sven O. Aase, and John Hakon Husoy. Method of optimal directions for frame design. In *Proceedings of the IEEE International Conference on the Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2443–2446, 1999.

Mark Everingham, Andrew Zisserman, Chris K. I. Williams, and Luc Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf, 2006.

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

Uriel Feige and Rani Izsak. Welfare maximization and the supermodular degree. In *Innovations in Theoretical Computer Science (ITCS)*, pages 247–256, 2013.

Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

Moran Feldman. *Maximization Problems with Submodular Objective Functions*. PhD thesis, Computer Science Department, Technion - Israel Institute of Technology, 2013.

Moran Feldman and Rani Izsak. Constrained monotone function maximization and the supermodular degree. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 160–175, 2014.

Moran Feldman, Joseph Naor, Roy Schwartz, and Justin Ward. Improved approximations for k-exchange systems (extended abstract). In *Proceedings of the 19th Annual European Symposium on Algorithms (ESA)*, pages 784–798, 2011.

Alan Fern, Robby Goetschalckx, Mandana Hamidi-Haines, and Prasad Tadepalli. Adaptive submodularity with varying query sets: An application to active multi-label learning. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory (ALT)*, pages 577–592, 2017.

Yuval Filmus and Justin Ward. Monotone submodular maximization over a matroid via non-oblivious local search. *SIAM Journal on Computing*, 43(2):514–542, 2014.

Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. *An analysis of approximations for maximizing submodular set functions—II*, pages 73–87. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013.

Kaito Fujii and Hisashi Kashima. Budgeted stream-based active learning via adaptive submodular maximization. In *Advances in Neural Information Processing Systems (NIPS) 29*, pages 514–522, 2016.

Kaito Fujii and Shinsaku Sakaue. Beyond adaptive submodularity: Approximation guarantees of greedy policy with adaptive submodularity ratio. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2042–2051, 2019.

Kaito Fujii and Tasuku Soma. Fast greedy algorithms for dictionary selection with generalized sparsity constraints. In *Advances in Neural Information Processing Systems (NeurIPS) 31*, pages 4749–4758, 2018.

Satoru Fujishige. *Submodular Functions and Optimization*. Elsevier, 2nd edition, 2005.

Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Advances in Neural Information Processing Systems (NIPS) 26*, pages 2697–2705, 2013.

Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Large-scale optimistic adaptive submodularity. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1816–1823, 2014.

Daniel Golovin and Andreas Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. *CoRR*, abs/1003.3967, 2010.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011a.

Daniel Golovin and Andreas Krause. Adaptive submodular optimization under matroid constraints. *CoRR*, abs/1101.4450, 2011b.

Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems (NIPS) 23*, pages 766–774, 2010.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

Alon Gonen, Sivan Sabato, and Shai Shalev-Shwartz. Efficient active learning of halfspaces: An aggressive approach. *Journal of Machine Learning Research*, 14:2583–2615, 2013.

Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *Proceedings of the 16th International Conference of Integer Programming and Combinatorial Optimization (IPCO)*, pages 205–216, 2013.

Daisuke Hatano, Takuro Fukunaga, and Ken-ichi Kawarabayashi. Adaptive budget allocation for maximizing influence of advertisements. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3600–3608, 2016.

Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference of Machine Learning (ICML)*, pages 417–424, 2006.

Thibaut Horel and Yaron Singer. Maximization of approximately submodular functions. In *Advances in Neural Information Processing Systems (NIPS) 29*, pages 3045–3053, 2016.

Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *Journal of Machine Learning Research*, 12:3371–3412, 2009.

Ali Jalali, Christopher C. Johnson, and Pradeep Ravikumar. On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems (NIPS) 24*, pages 1935–1943, 2011.

Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, Drew Bagnell, and Siddhartha S. Srinivasa. Near optimal bayesian active learning for decision making. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 430–438, 2014.

Stefanie Jegelka and Jeff A. Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1897–1904, 2011.

T. A. Jenkyns. The efficacy of the "greedy" algorithm. In *Proceedings of the 7th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 341–350, 1976.

Satyen Kale, Zohar Karnin, Tengyuan Liang, and Dávid Pál. Adaptive feature selection: Computationally efficient online sparse linear regression under RIP. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1780–1788, 2017.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 137–146, 2003.

David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

Rajiv Khanna, Ethan Elenberg, Alexandros Dimakis, Joydeep Ghosh, and Sahand Negahban. On approximation guarantees for greedy low rank optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1837–1846, 2017.

Adam R. Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354, 2017.

Mladen Kolar and Eric Xing. Ultra-high dimensional multiple output learning with simultaneous orthogonal matching pursuit: Screening approach. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 413–420, 2010.

Andreas Krause and Volkan Cevher. Submodular dictionary selection for sparse representation. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 567–574, 2010.

Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

Matt J Kusner. Approximately adaptive submodular maximization. In *NIPS Workshop on Discrete and Combinatorial Problems in Machine Learning*, 2014.

Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research*, 35(4):795–806, 2010.

Jon Lee, Maxim Sviridenko, and Jan Vondrák. Matroid matching: The power of local search. *SIAM Journal on Computing*, 42(1):357–379, 2013.

Benny Lehmann, Daniel J. Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 420–429, 2007a.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M. VanBriesen, and Natalie S. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 420–429, 2007b.

Edo Liberty and Maxim Sviridenko. Greedy minimization of weakly supermodular set functions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 19:1–19:11, 2017.

Hui Lin and Jeff A. 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 (ACL)*, pages 510–520, 2011.

Aurélie C. Lozano and Grzegorz Swirszcz. Multi-level lasso for sparse multi-task regression. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML)*, pages 595–602, 2012.

Takanori Maehara, Yasushi Kawase, Hanna Sumita, Katsuya Tono, and Ken-ichi Kawarabayashi. Optimal pricing for submodular valuations with bounded curvature. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pages 622–628, 2017.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. *Journal of Machine Learning Research*, 17:238:1–238:44, 2016.

Kiyohito Nagano, Yoshinobu Kawahara, and Satoru Iwata. Minimum average cost clustering. In *Advances in Neural Information Processing Systems (NIPS) 23*, pages 1759–1767, 2010.

Feng Nan and Venkatesh Saligrama. Comments on the proof of adaptive stochastic set cover based on adaptive submodularity and its implications for the group identification problem in "group-based active query selection for rapid diagnosis in time-critical situations". *IEEE Transactions on Information Theory*, 63(11):7612–7614, 2017.

Balas K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2): 227–234, 1995.

Deanna Needell and Joel A. Tropp. Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, 2010.

Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4): 538–557, 2012.

George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.

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.

Guillaume Obozinski, Ben Taskar, and Michael Jordan. Multi-task feature selection. In *In the workshop of structural Knowledge Transfer for Machine Learning in the 23rd International Conference on Machine Learning (ICML)*, 2006.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1135–1144, 2016.

Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Double sparsity: learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564, 2010.

Cristian Rusu, Bogdan Dumitrescu, and Sotirios A. Tsaftaris. Explicit shift-invariant dictionary learning. *IEEE Signal Processing Letters*, 21(1):6–9, 2014.

Shinsaku Sakaue. On maximization of weakly modular functions: Guarantees of multi-stage algorithms, tractability, and hardness. *CoRR*, abs/1805.11251, 2019.

Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Berlin, 2003.

Burr Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

Tasuku Soma, Naonori Kakimura, Kazuhiro Inaba, and Ken-ichi Kawarabayashi. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 351–359, 2014.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1015–1022, 2010.

Serban Stan, Morteza Zadimoghaddam, Andreas Krause, and Amin Karbasi. Probabilistic submodular maximization in sub-linear time. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3241–3250, 2017.

Matthew J. Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems (NIPS) 21*, pages 1577–1584, 2008.

Matthew J. Streeter, Daniel Golovin, and Andreas Krause. Online learning of assignments. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 1794–1802, 2009.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 75–86, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

Joel A. Tropp, Anna C. Gilbert, and Martin J. Strauss. Algorithms for simultaneous sparse approximation. part I: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.

Justin Ward. A (k+3)/2-approximation algorithm for monotone submodular k-set packing and general k-exchange systems. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 42–53, 2012.

Mehrdad Yaghoobi, Laurent Daudet, and Michael E. Davies. Dictionary subselection using an overcomplete joint sparsity model. *IEEE Transactions on Signal Processing*, 62(17):4547–4556, 2014.

Sze Zheng Yong, Lingyun Gao, and N. Ozay. Weak adaptive submodularity and group-based active diagnosis with applications to state estimation with persistent sensor faults. In *2017 American Control Conference (ACC)*, pages 2574–2581, 2017.

Tong Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE Transactions on Information Theory*, 57(7):4689–4708, 2011.

Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John W Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *Advances in Neural Information Processing Systems (NIPS) 22*, pages 2295–2303. 2009.