A

Max-Sum Diversification, Monotone Submodular Functions and Dynamic Updates

Allan Borodin, University of Toronto
Aadhar Jain, Under Armour Connected Fitness
Hyun Chul Lee, Under Armour Connected Fitness
Yuli Ye, Wish

Result diversification is an important aspect in web-based search, document summarization, facility location, portfolio management and other applications. Given a set of ranked results for a set of objects (e.g. web documents, facilities, etc.) with a distance between any pair, the goal is to select a subset S satisfying the following three criteria: (a) the subset S satisfies some constraint (e.g. bounded cardinality); (b) the subset contains results of high "quality"; and (c) the subset contains results that are "diverse" relative to the distance measure. The goal of result diversification is to produce a diversified subset while maintaining high quality as much as possible. We study a broad class of problems where the distances are a metric, where the constraint is given by independence in a matroid, where quality is determined by a monotone submodular function, and diversity is defined as the sum of distances between objects in S. Our problem is a generalization of the *max sum diversification* problem studied in [Gollapudi and Sharma 2009] which in turn is a generalization of the *max sum p-dispersion problem* studied extensively in location theory. It is NP-hard even with the triangle inequality. We propose two simple and natural algorithms: a greedy algorithm for a cardinality constraint and a local search algorithm for an arbitrary matroid constraint. We prove that both algorithms achieve constant approximation ratios.

CCS Concepts:• **Information systems** → **Information retrieval diversity;• Theory of computation** → **Design and analysis of algorithms; Approximation algorithms analysis;**

Additional Key Words and Phrases: diversification, dispersion, submodular functions, greedy algorithms, local search

1. INTRODUCTION

Suppose a search engine wishes to provide a more nuanced service to users who wish to be provided with a diversity of web page responses to a query. Without the diversity requirement, any set S of web pages achieves some quality value $f(S)$ relative to the query as determined by the search engine. It is reasonable to assume that this quality function is a monotone submodular function; that is (informally), additional pages will not lessen the value but any increase in value will be at a decreasing rate. Now the user may wish to balance that quality score with a diversity requirement. For example, the user may want to limit the number of retured pages that primarily fall within different topics and as well limit the total number of web pages being returned. This requirement can be enforced by a matroid constraint. The user may also want the results to represent different styles of writing as say represented by a vector of indicative word occurrences in the web doucments being returned. This kind of diversity

Author's addresses: A. Borodin, Department of Computer Science, University of Toronto, 10 Kings College Rd., Toronto, Ontario, Canada M5S3G4; email: bor@cs.toronto.edu; A. Jain, Under Armour Connected Fitness, Mountain View, California; email: ajain@myfitnesspal.com; H.C.Lee, Under Armour Connected Fitness, Mountain View, California; email: clee@myfitnesspal.com; Y. Ye, Wish Toronto Office, Unit 609, 15 Wertheim Court, Richmond Hill, Ontario, Canada L4B3H7; email: yuli.ye@gmail.com.

DOI: http://dx.doi.org/10.1145/0000000.0000000

can be modeled by a distance function d between the documents and it is reasonable to assume that this distance function is a metric. This is an example of the general result diversification problem we consider in this paper.

Result diversification has many important applications in databases, operations research, information retrieval, and finance. In this paper, we study and extend a particular version of result diversification, known as max-sum diversification. More specifically, we consider the setting where we are given a set of elements in a metric space and a set valuation function f defined on every subset. For any given subset S, the overall objective is a linear combination of $f(S)$ and the sum of the distances induced by S. The goal is to find a subset S satisfying some constraints that maximizes the overall objective.

This diversification problem was first studied by Gollapudi and Sharma [2009] for modular (i.e. linear) set functions and for sets satisfying a cardinality constraint (i.e. a uniform matroid). The max-sum p-dispersion problem seeks to find a subset S of cardinality p so as to maximize $\sum_{x,y \in S} d(x,y)$. The diversification problem is then a linear combination of a quality function $f()$ and the max-sum dispersion function. Gollapudi and Sharma give a greedy 2 approximation algorithm for some metrical distance diversification problems by reducing to the analogous dispersion problem. More specifically for max-sum diversification they use the greedy algorithm of Hassin, Rubsenstein and Tamir [1997]. Hassin et al. give a non greedy algorithm for a more general problem where the goal is to construct k subsets each having p elements. (We will restrict attention to the case $k = 1$.) Their non greedy algorithm obtains the ratio $2 - \frac{1}{\lceil p/2 \rceil}$ and hence the same approximation holds for the Gollapudi and Sharma diversification problem.

The first part of our paper considers an extension of the modular case to the monotone submodular case, for which the algorithm in [Gollapudi and Sharma 2009] no longer applies. We are able to maintain the same 2-approximation using a natural, but different greedy algorithm. We then further extend the problem by considering any matroid constraint and show that a natural single swap local search algorithm provides a 2-approximation in this more general setting. This extends the Nemhauser, Wolsey and Fisher [1978] approximation result for the problem of submodular function maximization subject to a matroid constraint (without the distance function component). We note that the dispersion function is a supermodular function [1] and hence the Nemhauser et al. result does not immediately extend to our diversification problem.

Submodular functions have been extensively considered since they model many natural phenomena. For example, in terms of keyword based search in database systems, it is well understood that users begin to gradually (or sometimes abruptly) lose interest the more results they have to consider [Vieira et al. 2011a; 2011b]. But on the other hand, as long as a user continues to gain some benefit, additional query results can improve the overall quality but at a decreasing rate. In a related application, Lin and Bilmes [2011] argue that monotone submodular functions are an ideal class of functions for text summarization. Following and extending the results in [Gollapudi and Sharma 2009], we consider the case of maximizing a linear combination of a submodular quality function $f(S)$ and the max-sum dispersion subject to a cardinality constraint (i.e., $|S| \leq p$ for some given p). We present a greedy algorithm that is somewhat unusual in that it does not try to optimize the objective in each iteration but rather op-

[1] Motivated by the analysis in this paper, Borodin et al. [2014] introduce the class of *weakly submodular functions* and show that the max-sum dispersion measure as well as all monotone submodular functions are weakly submodular. Furthermore, it is shown that the problem of maximizing such functions subject to cardinality (resp. general matroid) constraints can be polynomial time approximated within a constant factor by a greedy (resp. local search) algorithm.

timizes a closely related potential function. We show that our greedy approach matches the greedy 2-approximation [2] in [Gollapudi and Sharma 2009] obtained for diversification with a modular quality function. We note that the greedy algorithm in [Gollapudi and Sharma 2009] utilizes the max dispersion algorithm of Hassin, Rubinstein and Tamir [1997] which greedily adds edges whereas our algorithm greedily adds vertices.

Our next result continues with the submodular case but now we go beyond a cardinality constraint (i.e., the uniform matroid) on S and allow the constraint to be that S is independent in a given matroid. This allows a substantial increase in generality. In a partition matroid, the universe U is partitioned into sets $S_1, \ldots, S_m$ and the independent sets S satisfy $S = \cup_{1 \leq i \leq m} S_i$ with $|S_i| \leq p_i$ for some given bounds p_i on each block of the partition. The cardinality constraint is then a special case of a partition matroid with $m = 1$. While diversity might be represented by the distance between retrieved database tuples under a given criterion (for instance, a kernel based diversity measure called *answer tree kernel* is used in [Zhao et al. 2011]), we could use a partition matroid to insure that (for example) the retrieved database tuples come from a variety of different sources. That is, we may wish to have p_i tuples from a specific database field i. This is, of course, another form of diversity but one orthogonal to diversity based on the given criterion. Similarly in the stock portfolio example, we might wish to have a balance of stocks in terms of say risk and profit profiles (using some statistical measure of distance) while using a submodular quality function to reflect a users submodular utility for profit and using a partition matroid to insure that different sectors of the economy are well represented. Another important class of matroids (relevant to the above applications) is that of transversal matroids. In a transversal matroid, the universe U is a union of (possibly) intersecting sets $\mathcal{C} = C_1, \ldots, C_m$ and a set $S = \{s_1, \ldots s_r\} \subseteq U$ is an independent set in the traversal matroid induced by the collection if there is an injective function ϕ from S into $\mathcal{C}$ with say $\phi(s_i) = C_i$ and $\phi(s_i) \in C_i$. That is, S forms a set of representatives for each set C_i or equivalently there is a matching between S and $\mathcal{C}$. (Note that a given s_i could occur in other sets C_j.) In a database application, our goal might be to derive a set S such that the database tuples in S form a set of representatives for the collection. We also note that Schrijver et al [2003] show that the intersection of any matroid with a uniform matroid is still a matroid so that in the above examples, we could further impose the constraint that the set S has at most p elements.

Our final theoretical result concerns dynamic updates. Here we restrict attention to a modular set function $f(S)$; that is, we now have weights on the elements and $f(S) = \sum_{u \in S} w(u)$ where $w(u)$ is the weight of element u. This allows us to consider changes to the weight of a single element as well as changes to the distance function.

The rest of the paper is organized as follows. In Section 2, we discuss related work in dispersion and result diversification. In Section 3, we formulate the problem as a combinatorial optimization problem and discuss the complexity of the problem. In Section 4, we consider max-sum diversification with monotone submodular set quality functions subject to a cardinality constraint and give a conceptually simple greedy algorithm that achieves a 2-approximation. We extend the problem to the matroid case in Section 5 and discuss dynamic updates in Section 6. Section 7 carries out a number of experiments. In particular, we compare our greedy algorithm with the greedy algorithm of Gollapudi and Sharma. Section 8 concludes the paper.

[2] Clearly, in the modular case for p constant, a brute force trial of all subsets of size p is an optimum, albeit inefficient, algorithm.

2. RELATED WORK

With the proliferation of today's social media, database and web content, ranking becomes an important problem as it decides what gets selected and what does not, what is to be displayed first and what is to be displayed last. Many early ranking algorithms, for example in web search, are based on the notion of "relevance", i.e., the closeness of the object to the search query. However, there has been a rising interest to incorporate some notion of "diversity" into measures of quality.

One early work in this direction is the notion of "Maximal Marginal Relevance" (MMR) introduced by Carbonell and Goldstein in [Carbonell and Goldstein 1998]. More specifically, MMR is defined as follows:

$$\mathrm{MMR} = \max_{D_i \in R \setminus S} [\lambda \cdot sim_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} sim_2(D_i, D_j)],$$

where Q is a query; R is the ranked list of documents retrieved; S is the subset of documents in R already selected; sim_1 is the similarity measure between a document and a query, and sim_2 is the similarity measure between two documents. The parameter λ controls the trade-off between novelty (a notion of diversity) and relevance. The MMR algorithm iteratively selects the next document with respect to the MMR objective function until a given cardinality condition is met. The MMR heuristic has been widely used, but to the best of our knowledge, it has not been theoretically justified. Our paper provides some theoretical evidence why MMR is a legitimate approach for diversification. The greedy algorithm we propose in this paper can be viewed as a natural extension of MMR.

There is extensive research on how to diversify returned ranking results to satisfy multiple users. Namely, the result diversity issue occurs when many facets of queries are discovered and a set of multiple users expect to find their desired facets in the first page of the results. Thus, the challenge is to find the best strategy for ordering the results such that many users would find their relevant pages in the top few slots.

Rafiei et al. [Rafiei et al. 2010] modeled this as a continuous optimization problem. They introduce a weight vector W for the search results, where the total weight sums to one. They define the portfolio variance to be W^TCW, where C is the co-variance matrix of the result set. The goal then is to minimize the portfolio variance while the expected relevance is fixed at a certain level. They report that their proposed algorithm can improve upon Google in terms of the diversity on random queries, retrieving 14% to 38% more aspects of queries in top five, while maintaining a precision very close to Google.

Bansal et al. [Bansal et al. 2010] considered the setting in which various types of users exist and each is interested in a subset of the search results. They use a performance measure based on *discounted cumulative gain*, which defines the usefulness (gain) of a document as its position in the resulting list. Based on this measure, they suggest a general approach for developing approximation algorithms for ranking search results that captures different aspects of users' intents. They also take into account that the relevance of one document cannot be treated independent of the relevance of other documents in a collection returned by a search engine. They consider both the scenario where users are interested in only a single search result (e.g., navigational queries) and the scenario where users have different requirements on the number of search results, and develop good approximation solutions for them.

The database community has studied the query diversification problem, which is mainly for keyword search in databases [Liu et al. 2009; Yu et al. 2009; Drosou and Pitoura 2009; Vieira et al. 2011b; Zhao et al. 2011; Vieira et al. 2011a; Demidova et al. 2010]. Given a very large database, an exploratory query can easily lead to a vast answer set. Typically, an answer's relevance to the user query is based on *top-k* or *tf-idf*.

As a way of increasing user satisfaction, different query diversification techniques have been proposed including some system based ones taking into account query parameters, evaluation algorithms, and dataset properties. For many of these, a max-sum type objective function is usually used.

Other than those discussed above, there are many papers studying result diversification in different settings, via different approaches and through different perspectives, for example [Zhai et al. 2003; Chen and Karger 2006; Zhu et al. 2007; Yue and Joachims 2008; Radlinski et al. 2008; Agrawal et al. 2009; Brandt et al. 2011; Santos et al. 2011; Dou et al. 2011; Slivkins et al. 2010]. The reader is referred to [Agrawal et al. 2009; Drosou and Pitoura 2010] for a good summary of the field. Most relevant to our work is the paper by Gollapudi and Sharma [Gollapudi and Sharma 2009], where they develop an axiomatic approach to characterize and design diversification systems. Furthermore, they consider three different diversification objectives and using earlier results in facility dispersion, they are able to give algorithms with good worst case approximation guarantees. This paper is a continuation of research along this line.

Minack et al. [Minack et al. 2011] have studied the problem of incremental diversification for very large data sets. Instead of viewing the input of the problem as a set, they consider the input as a stream, and use a simple online algorithm to process each element in an incremental fashion, maintaining a near-optimal diverse set at any point in the stream. Although their results are largely experimental, this approach significantly reduces CPU and memory consumption, and hence is applicable to large data sets. Our dynamic update algorithm deals with a problem of a similar nature, but in addition to our experimental results, we are also able to prove theoretical guarantees. To the best of our knowledge, our work is the first of its kind to obtain a near-optimality condition for result diversification in a dynamically changing environment.

Independent of our conference paper [Borodin et al. 2012], Abbassi, Mirrokni and Thakus [Abbassi et al. 2013] have also shown that the (Hamming distance 1) local search algorithm provides a 2-approximation for the max-sum dispersion problem subject to a matroid constraint. Their version of the dispersion problem is somewhat more general in that they additionally consider that the points are chosen from different clusters. They indirectly consider a quality measure by first restricting the universe of objects to high quality objects and then apply dispersion. They provide a number of interesting experimental results.

3. PROBLEM FORMULATION

Although the notion of “diversity” naturally arises in the context of databases, social media and web search, the underlying mathematical object is not new. As presented in [Gollapudi and Sharma 2009], there is a rich and long line of research in location theory dealing with a similar concept; in particular, one objective is the placement of facilities on a network to maximize some function of the distances between facilities. The situation arises when proximity of facilities is undesirable, for example, the distribution of business franchises in a city. Such location problems are often referred to as *dispersion* problems; for more motivation and early work, see [Erkut 1990; Erkut and Neuman 1989; Kuby 1987].

Analytical models for the dispersion problem assume that the given network is represented by a set $V = \{v_1, v_2, \dots, v_n\}$ of n vertices along with a distance function between every pair of vertices. The objective is to locate p facilities ($p \le n$) among the n vertices, with at most one facility per vertex, such that some function of distances between facilities is maximized. Different objective functions are considered for the dispersion problems in the literature including: the max-sum criterion (maximize the total distances between all pairs of facilities) in [Wang and Kuo 1988; Erkut 1990; Ravi et al. 1994], the max-min criterion (maximize the minimum distance between a pair of facilities) in [Kuby 1987; Erkut 1990; Ravi et al. 1994], the max-mst (maximize the minimum spanning tree among all facilities) and many other related criteria in [Halldórsson et al. 1995; Chandra and Halldórsson 2001]. When the distances are arbitrary, the max-sum problem is a weighted generalization of the densest subgraph problem which is a known difficult problem not admitting a PTAS ([Khot 2006] and not known to have a constant approximation algorithm. Sometimes the problem is studied for specific metric distances (e.g as in Fekete and Meijer [Fekete and Meijer 2003]) or for restricted classes of weights (e.g. as in Czygrinow [Czygrinow 2000]) where there can be a PTAS. Our diversification problem is a generalization of the max sum p-dispersion problem assuming arbitrary metric distances. For the max-sum criteria and for most of the objective criteria, the dispersion problem is NP-hard, and approximation algorithms have been developed and studied; see [Chandra and Halldórsson 2001] for a summary of known results. Our diversification problem is a generalization of the following max sum p-dispersion problem for arbitrary metric distances. Most relevant to this paper is the max-sum dispersion problem with metric distances.

PROBLEM 1. `Max-Sum` p `Dispersion`

Let U be the underlying ground set, and let $d(\cdot,\cdot)$ be a metric distance function on U. Given a fixed integer p, the goal of the problem is to find a subset $S \subseteq U$ that:

$$\begin{array}{ll}\text{maximizes} & \sum_{\{u,v\}:u,v\in S} d(u,v) \\ \text{subject to} & |S| = p,\end{array}$$

The problem is known to be NP-hard by an easy reduction from Max-Clique, and as noted by Alon [Alon 2014], there is evidence that the problem is hard to compute in polynomial time with approximation $2-\epsilon$ for any $\epsilon > 0$ when $p = n^r$ for $1/3 \leq r < 1$ (for suffiently large n). Namely, based on the assumption that the planted clique problem is hard [3], Alon et al [Alon et al. 2011] show that it is hard to distinguish between a graph having a large planted clique of size p and one in which the densest sub-graph of size p is of density at most an arbitrarily small constant δ. Considering the complement of a random graph G in $\mathcal{G}(n, 1/2)$, their result says that it is hard to distinguish between a graph having an independent set of size p and one in which the density of edges in any size p-sub-graph is at least $(1-\delta)$. Adding another node to the complement graph that is connected to all nodes in G, the graph distance metric is now the $\{1, 2\}$ metric formed by the transitive closure so that adjacent nodes have distance 1 and non adjacent nodes have distance 2. So we therefore cannot distinguish between graphs where there exists a set of nodes S of size p (for p as above) where $\sum_{(u,v)\in S} d(u,v) = \binom{p}{2} * 2$ and one where in every set of size p, we have $\sum_{(u,v)\in S} d(u,v) \leq \binom{p}{2}[(1-\delta)+2\delta]$.

In [Ravi et al. 1994], Ravi, Rosenkrantz and Tayi give a greedy algorithm (greedily choosing vertices) that is shown to have approximation ratio no worse than 4 and no better than $\frac{2}{1+2/p(p-1)}$. Hassin, Rubenstein and Tamir [Hassin et al. 1997] improve upon the Ravi et al result by an algorithm that greedily chooses *edges* yielding an approximation ratio of 2. Hassin et al also give an algorithm based on maximum matching that provides a $2 - \frac{1}{\lceil p/2 \rceil}$ approximation for a more general problem; namely, the algorithm must find a subset U' which is partitioned into k disjoint subsets, each of size p so as to maximize the pairwise sum of all pairs of vertices in U'. The more general (p, k) problem is similar to a partition matroid constraint but in a partition matroid, the partition is given as part of the definition of the matroid and each block of the partition has its own cardinality constraint.

[3] See [Meka et al. 2015] for the latest evidence with regard to the hardness of the planted clique problem.

Answering an open problem stated in Hassin et al., Birnbaum and Goldman [Birnbaum and Goldman 2009] give an improved analysis proving that the Ravi et al. greedy algorithm results in a $\frac{2p-2}{p-1}$ approximation for the max-sum p dispersion problem. This then shows that a 2-approximation is a tight bound (as p grows) for the Ravi et al. greedy algorithm. More generally, Birnbaum and Goldman show that greedily choosing a set of d nodes provides a $\frac{2p-2}{p+d-2}$ approximation. Our analysis in Section 4 yields an alternative proof that the Ravi et al. greedy algorithm approximation ratio is no worse than 2 even when extended to the max-sum p diversification problem (with a monotone submodular value function) considered in Section 4.

PROBLEM 2. `Max-Sum` p `Diversification`

Let U be the underlying ground set, and let $d(\cdot,\cdot)$ be a metric distance function on U. For any subset of U, let $f(\cdot)$ be a non-negative set function measuring the value of a subset. Given a fixed integer p, the goal of the problem is to find a subset $S \subseteq U$ that:

$$\begin{array}{ll} \text{maximizes} & f(S) + \lambda \sum_{\{u,v\}:u,v\in S} d(u,v) \\ \text{subject to} & |S| = p, \end{array}$$

where λ is a parameter specifying a desired trade-off between the two objectives.

The max-sum diversification problem is first proposed and studied in the context of result diversification in [Gollapudi and Sharma 2009] [4], where the function $f(\cdot)$ is modular. In their paper, the value of $f(S)$ measures the relevance of a given subset to a search query, and the value $\sum_{\{u,v\}:u,v\in S} d(u,v)$ gives a diversity measure on S. The parameter λ specifies a desired trade-off between diversity and relevance. They reduce the problem to the max-sum dispersion problem, and using an algorithm in [Hassin et al. 1997], they obtain an approximation ratio of 2.

In this paper, we first study the problem with more general valuation functions; namely, normalized, monotone submodular set functions. For notational convenience, for any two sets S, T and an element e, we write $S \cup \{e\}$ as $S + e$, $S \setminus \{e\}$ as $S - e$, $S \cup T$ as $S + T$, and $S \setminus T$ as $S - T$. A set function f is *normalized* if $f(\emptyset) = 0$. The function is *monotone* if for any $S, T \subseteq U$ and $S \subseteq T$,

$$f(S) \leq f(T).$$

It is *submodular* if for any $S, T \subseteq U$, $S \subseteq T$ with $u \in U$,

$$f(T + u) - f(T) \leq f(S + u) - f(S).$$

In the remainder of paper, all functions considered are normalized and monotone. We proceed to our first contribution, a greedy algorithm (different than the one in [Gollapudi and Sharma 2009]) that obtains a 2-approximation for monotone submodular set functions.

4. SUBMODULAR FUNCTIONS

Submodular set functions can be characterized by the property of a decreasing marginal gain as the size of the set increases. As such, submodular functions are well-studied objects in economics, game theory and combinatorial optimization. Submodular functions have also attracted attention in different fields of computer science. For example, Kempe et al. [2003] study the problem of selecting a set of most influential nodes to maximize the total information spread in a social network. They have

[4] In fact, they have a slightly different but equivalent formulation.

shown that under two basic stochastic diffusion models, the expected influence of an initially chosen set is submodular, hence the problem admits a good approximation algorithm. In natural language processing, Lin and Bilmes [Lin et al. 2009; Lin and Bilmes 2010; 2011] have studied a class of submodular functions for document summarization. These functions each combine two terms, one which encourages the summary to be representative of the corpus, and the other which positively rewards diversity. Their experimental results show that a greedy algorithm with the objective of maximizing these submodular functions outperforms the existing state-of-art results in both generic and query-focused document summarization.

Both of the above mentioned results are based on the fundamental work of Nemhauser, Wolsey and Fisher [1978], which gave an $\frac{e}{e-1}$-approximation for maximizing monotone submodular set functions over a uniform matroid. This bound is now known to be tight even for a general matroid [2011] whereas the greedy algorithm provides a 2-approximation for an arbitrary matroid (and a $k+1$-approximation for the intersection of k matroids) as shown in [Fisher et al. 1978]. Our max-sum diversification problem with monotone submodular set functions can be viewed as an extension of that problem: the objective function now not only contains a submodular part, but also has a super-modular part: the sum of distances.

Since the max-sum diversification problem with modular set functions studied in [Gollapudi and Sharma 2009] admits a 2-approximation algorithm, it is natural to ask what approximation ratio is obtainable for the same problem with monotone submodular set functions. The Gollapudi and Sharma algorithm is based on the observation that the diversity function with modular set functions can be reduced to the max-sum p dispersion problem by changing the metric. Namely, the reduction defines the metric $d'(u,v) = w(u) + w(v) + 2\lambda d(u,v)$. It is clear then that this reduction and the algorithm in [Gollapudi and Sharma 2009] does not apply to the submodular case where elements do not have weights but rather only marginal weights. While this suggests that a greedy algorithm using marginal weights might apply (as we will show), this still requires a proof and in general one cannot expect the same approximation ratio. In what follows we assume (as is standard when considering submodular functions) access to an oracle for finding an element $u \in U - S$ that maximizes $f(S+u) - f(S)$. When f is modular, this simply means accessing the element $u \in U - S$ having maximum weight.

THEOREM 4.1. *There is a simple linear time greedy algorithm that achieves a 2-approximation for the max-sum diversification problem with monotone submodular set functions satisfying a cardinality constraint.*

Before giving the proof [5] ”of Theorem 4.1, we introduce some additional notation. We extend the notion of distance function to sets. For disjoint subsets $S, T \subseteq U$, we let $d(S) = \sum_{\{u,v\}:u,v \in S} d(u,v)$, and $d(S,T) = \sum_{\{u,v\}:u \in S, v \in T} d(u,v)$.

Now we define various types of marginal gain. For any given subset $S \subseteq U$ and an element $u \in U - S$: let $\phi(S)$ be the value of the objective function, $d_u(S) = \sum_{v \in S} d(u,v)$ be the marginal gain on the distance, $f_u(S) = f(S+u) - f(S)$ be the marginal gain on the weight, and $\phi_u(S) = f_u(S) + \lambda d_u(S)$ be the total marginal gain on the objective

[5] While greedy algorithms are conceptually simple to state and understand operationally, it can be the case that the analysis of an approximation ratio is not at all simple. For example, the Birnbaum and Goldman proof that the greedy algorithm is a 2-approximation for the cardinality constrained metric sum dispersion problem is such a proof. Their proof answered an explicit 12 year old conjecture by Hassin et al [1997] following the 4-approximation by Ravi et al [1994]. The earlier Ravi et al paper gave an example showing that the greedy algorithm was no better than a 2-approximation motivating the conjecture by Hassin et al.

function. Let $f'_u(S) = \frac{1}{2} f_u(S)$, and $\phi'_u(S) = f'_u(S) + \lambda d_u(S)$. We consider the following simple greedy algorithm:

```
GREEDY ALGORITHM
$S = \emptyset$
while $|S| < p$
    find $u \in U - S$ maximizing $\phi'_u(S)$
    $S = S + u$
end while
return $S$
```

Note that the above greedy algorithm is "non-oblivious" (in the sense of [Khanna et al. 1998]) as it is not selecting the next element with respect to the objective function $\phi(\cdot)$. This might be of an independent interest. We utilize the following lemma in [Ravi et al. 1994].

LEMMA 4.2. *Given a metric distance function* $d(\cdot,\cdot)$, *and two disjoint sets* X *and* Y, *we have the following inequality:* $(|X| - 1)d(X,Y) \geq |Y|d(X)$.

Now we are ready to prove Theorem 4.1.

PROOF. Let O be the optimal solution, and G, the greedy solution at the end of the algorithm. Let G_i be the greedy solution at the end of step i, $i < p$; and let $A = O \cap G_i$, $B = G_i - A$ and $C = O - A$. By lemma 4.2, we have the following three inequalities:

$$(|C| - 1)d(B,C) \geq |B|d(C) \tag{1}$$

$$(|C| - 1)d(A,C) \geq |A|d(C) \tag{2}$$

$$(|A| - 1)d(A,C) \geq |C|d(A) \tag{3}$$

Furthermore, we have

$$d(A,C) + d(A) + d(C) = d(O) \tag{4}$$

Note that the algorithm clearly achieves the optimal solution if $p = 1$. If $|C| = 1$, then $i = p - 1$ and $G_i \subset O$. Let v be the element in C, and let u be the element taken by the greedy algorithm in the next step, then $\phi'_u(G_i) \geq \phi'_v(G_i)$. Therefore, $\frac{1}{2} f_u(G_i) + \lambda d_u(G_i) \geq \frac{1}{2} f_v(G_i) + \lambda d_v(G_i)$, which implies $\phi_u(G_i) = f_u(G_i) + \lambda d_u(G_i) \geq \frac{1}{2} f_u(G_i) + \lambda d_u(G_i) \geq \frac{1}{2} f_v(G_i) + \lambda d_v(G_i) \geq \frac{1}{2}\phi_v(G_i)$ and hence $\phi(G) \geq \frac{1}{2}\phi(O)$.

Now we can assume that $p > 1$ and $|C| > 1$. We apply the following non-negative multipliers to equations (1), (2), (3), (4) and add them: $(1) * \frac{1}{|C|-1} + (2) * \frac{|C|-|B|}{p(|C|-1)} + (3) * \frac{i}{p(p-1)} + (4) * \frac{i|C|}{p(p-1)}$; we then have $d(A,C) + d(B,C) - \frac{i|C|(p-|C|)}{p(p-1)(|C|-1)} d(C) \geq \frac{i|C|}{p(p-1)} d(O)$. Since $p > |C|$, $d(C, G_i) \geq \frac{i|C|}{p(p-1)} d(O)$. By submodularity and monotonicity of $f'(\cdot)$, we have $\sum_{v \in C} f'_v(G_i) \geq f'(C \cup G_i) - f'(G_i) \geq f'(O) - f'(G)$. Therefore,

$$\begin{aligned} \sum_{v \in C} \phi'_v(G_i) &= \sum_{v \in C} [f'_v(G_i) + \lambda d(\{v\}, G_i)] \\ &= \sum_{v \in C} f'_v(G_i) + \lambda d(C, G_i) \geq [f'(O) - f'(G)] + \frac{\lambda i |C|}{p(p-1)} d(O). \end{aligned}$$

Let u_{i+1} be the element taken at step $(i + 1)$, then we have $\phi'_{u_{i+1}}(G_i) \geq \frac{1}{p}[f'(O) - f'(G)] + \frac{\lambda i}{p(p-1)} d(O)$. Summing over all i from 0 to $p - 1$, we have $\phi'(G) = \sum_{i=0}^{p-1} \phi'_{u_{i+1}}(G_i) \geq [f'(O) - f'(G)] + \frac{\lambda}{2} d(O)$. Hence, $f'(G) + \lambda d(G) \geq f'(O) - f'(G) + \frac{\lambda}{2} d(O)$, and $\phi(G) = f(G) + \lambda d(G) \geq \frac{1}{2}[f(O) + \lambda d(O)] = \frac{1}{2}\phi(O)$. This completes the proof. □

The greedy algorithm runs in time proportional to p (for the p iterations) times the cost of computing $\phi'_u(S)$ for a given u and S. When f is modular, the time for updating $\phi'_u(S)$ can be bounded by $O(n)$. Namely, each iteration costs $O(n)$ time (to search over all elements u in $U \setminus S$) and update $\phi'(S)$. Updating $f'(S)$ is clearly $O(1)$ while naively updating $d_u(S)$ would take time $O(p)$. But as observed by Birnbaum and Goldman [2009], $d_u(V')$ can be maintained for all $V \setminus S$ within the same $O(n)$ needed to search V' so that updating $\phi'(S)$ only costs time $O(1)$. Hence the total time is $O(np)$, linear in n when p is a constant.

COROLLARY 4.3. *The Ravi et al. [1994] greedy algorithm for dispersion has approximation ratio no worse that 2.*

PROOF. The identically zero function f is monotone submodular and for this f, our greedy algorithm is precisely the dispersion algorithm of Ravi et al. □

We note that for the dispersion problem, Birnbaum and Goldman [2009] show that their bound for the greedy algorithm is tight. In particular, for the greedy algorithm that adds one element at a time, the precise bound is $\frac{2p-2}{p-1}$.

5. MATROIDS AND LOCAL SEARCH

Theorem 4.1 provides a 2-approximation for max-sum diversification when the set function is submodular and the set constraint is a cardinality constraint, i.e., a uniform matroid. It is natural to ask if the same approximation guarantee can be obtained for an arbitrary matroid. In this section, we show that the max-sum diversification problem with monotone submodular function admits a 2-approximation subject to a general matroid constraint.

Matroids are well studied objects in combinatorial optimization. A matroid $\mathcal{M}$ is a pair $\langle U, \mathcal{F}\rangle$, where U is a set of ground elements and $\mathcal{F}$ is a collection of subsets of U, called *independent sets*, with the following properties :

— **Hereditary:** The empty set is independent and if $S \in \mathcal{F}$ and $S' \subset S$, then $S' \in \mathcal{F}$.
— **Augmentation:** If $A, B \in \mathcal{F}$ and $|A| > |B|$, then $\exists e \in A - B$ such that $B \cup \{e\} \in \mathcal{F}$.

The maximal independent sets of a matroid are called *bases* of $\mathcal{M}$. Note that all bases have the same number of elements, and this number is called the *rank* of $\mathcal{M}$. The definition of a matroid captures the key notion of independence from linear algebra and extends that notion so as to apply to many combinatorial objects. We have already mentioned two classes of matroids relevant to our results, namely partition matroids and transversal matroids.

PROBLEM 3. `Max-Sum Diversification for Matroids`

Let U be the underlying ground set, and $\mathcal{F}$ be the set of independent subsets of U such that $\mathcal{M} = \langle U, \mathcal{F}\rangle$ is a matroid. Let $d(\cdot,\cdot)$ be a (non-negative) metric distance function measuring the distance on every pair of elements. For any subset of U, let f be a non-negative monotone submodular set function measuring the weight of the subset. The goal of the problem is to find a subset $S \in \mathcal{F}$ that:

$$\text{maximizes } f(S) + \lambda \textstyle\sum_{\{u,v\}:u,v \in S} d(u,v)$$

where again λ is a parameter specifying a desired trade-off between the two objectives. As before, we let $\phi(S)$ be the value of the objective function. Note that since the function $\phi(\cdot)$ is monotone, S is essentially a basis of the matroid $\mathcal{M}$. The greedy algorithm in Section 4 still applies, but it fails to achieve any constant approximation ratio even for a linear quality function f including the identically zero function; that is, for max-sum dispersion. (See the Appendix.) This is in contrast to the seminal result of Nemhauser, Wolsey and Fisher [1978] showing that the greedy algorithm is optimal (respectively, a 2-approximation) for linear functions (respectively, monotone submodular functions) subject to a matroid constraint. Note that the problem is trivial if the rank of the matroid is less than two. Therefore, without loss of generality, we assume the rank is at least two. Let

$$\{x, y\} = \text{argmax}_{\{x,y\}\in\mathcal{F}}[f(\{x, y\}) + \lambda d(x, y)].$$

We now consider the following oblivious local search algorithm:

```
LOCAL SEARCH ALGORITHM
let $S$ be a basis of $\mathcal{M}$ containing both $x$ and $y$
while there is an $u \in U - S$ and $v \in S$ such that $S + u - v \in \mathcal{F}$ and $\phi(S + u - v) > \phi(S)$
    $S = S + u - v$
end while
return $S$
```

THEOREM 5.1. *The local search algorithm achieves an approximation ratio of 2 for max-sum diversification with a matroid constraint.*

Note that if the rank of the matroid is two, then the algorithm is clearly optimal. From now on, we assume the rank of the matroid is greater than two. Before we prove the theorem, we first give several lemmas. All the lemmas assume the problem and the underlying matroid without explicitly mentioning it. Let O be the optimal solution, and S, the solution at the end of the local search algorithm. Let $A = O \cap S$, $B = S - A$ and $C = O - A$.

LEMMA 5.2. *For any two sets $X, Y \in \mathcal{F}$ with $|X| = |Y|$, there is a bijective mapping $g : X \to Y$ such that $X - x + g(x) \in \mathcal{F}$ for any $x \in X$.*

This is a known property of a matroid and its proof can be found in [Brualdi 1969]. Since both S and O are bases of the matroid, they have the same cardinality. Therefore, B and C have the same cardinality. By Lemma 5.2, there is a bijective mapping $g : B \to C$ such that $S - b + g(b) \in \mathcal{F}$ for any $b \in B$. Let $B = \{b_1, b_2, \ldots, b_t\}$, and let $c_i = g(b_i)$ for all i. Without loss of generality, we assume $t \geq 2$, for otherwise, the algorithm is optimal by the local optimality condition.

LEMMA 5.3. $f(S) + \sum_{i=1}^{t} f(S - b_i + c_i) \geq f(S - \sum_{i=1}^{t} b_i) + \sum_{i=1}^{t} f(S + c_i)$.

PROOF. Since f is submodular,

$$f(S) - f(S - b_1) \geq f(S + c_1) - f(S + c_1 - b_1),$$
$$f(S - b_1) - f(S - b_1 - b_2) \geq f(S + c_2) - f(S + c_2 - b_2),$$
$$\ldots$$
$$f(S - \textstyle\sum_{i=1}^{t-1} b_i) - f(S - \sum_{i=1}^{t} b_i) \geq f(S + c_t) - f(S + c_t - b_t).$$

Summing up these inequalities, we have $f(S) - f(S - \sum_{i=1}^{t} b_i) \geq \sum_{i=1}^{t} f(S + c_i) - \sum_{i=1}^{t} f(S - b_i + c_i)$, and the lemma follows. □

LEMMA 5.4. $\sum_{i=1}^{t} f(S + c_i) \geq (t - 1)f(S) + f(S + \sum_{i=1}^{t} c_i)$.

PROOF. Since f is submodular,

$$f(S + c_t) - f(S),$$
$$f(S + c_{t-1}) - f(S) \geq f(S + c_t + c_{t-1}) - f(S + c_t),$$
$$f(S + c_{t-2}) - f(S) \geq f(S + c_t + c_{t-1} + c_{t-2}) - f(S + c_t + c_{t-1})$$
$$\ldots$$
$$f(S + c_1) - f(S) \geq f(S + \textstyle\sum_{i=1}^{t} c_i) - f(S + \sum_{i=2}^{t} c_i)$$

Summing up these inequalities, we have $\sum_{i=1}^{t} f(S+c_i) - tf(S) \geq f(S+\sum_{i=1}^{t} c_i) - f(S)$, and the lemma follows. □

LEMMA 5.5. $\sum_{i=1}^{t} f(S-b_i+c_i) \geq (t-2)f(S)+f(O)$.

PROOF. Combining Lemma 5.3 and Lemma 5.4, we have

$$\begin{aligned}&f(S)+\textstyle\sum_{i=1}^{t} f(S-b_i+c_i)\\ &\geq f(S-\textstyle\sum_{i=1}^{t} b_i)+\sum_{i=1}^{t} f(S+c_i)\\ &\geq (t-1)f(S)+f(S+\textstyle\sum_{i=1}^{t} c_i) =(t-1)f(S)+f(S+C)\\ &\geq (t-1)f(S)+f(O).\end{aligned}$$

Therefore, the lemma follows. □

LEMMA 5.6. *If* $t>2$, $d(B,C)-\sum_{i=1}^{t} d(b_i,c_i) \geq d(C)$.

PROOF. For any b_i, c_j, c_k, we have $d(b_i,c_j)+d(b_i,c_k) \geq d(c_j,c_k)$. Summing up these inequalities over all i,j,k with $i \neq j$, $i \neq k$, $j \neq k$, we have each $d(b_i,c_j)$ with $i \neq j$ is counted $(t-2)$ times; and each $d(c_i,c_j)$ with $i \neq j$ is counted $(t-2)$ times. Therefore $(t-2)[d(B,C)-\sum_{i=1}^{t} d(b_i,c_i)] \geq (t-2)d(C)$, and the lemma follows. □

LEMMA 5.7. $\sum_{i=1}^{t} d(S-b_i+c_i) \geq (t-2)d(S)+d(O)$.

PROOF. $\sum_{i=1}^{t} d(S-b_i+c_i)$

$$\begin{aligned}&=\textstyle\sum_{i=1}^{t}[d(S)+d(c_i,S-b_i)-d(b_i,S-b_i)]\\ &=td(S)+\textstyle\sum_{i=1}^{t} d(c_i,S-b_i)-\sum_{i=1}^{t} d(b_i,S-b_i)\\ &=td(S)+\textstyle\sum_{i=1}^{t} d(c_i,S)-\sum_{i=1}^{t} d(c_i,b_i)-\sum_{i=1}^{t} d(b_i,S-b_i)\\ &=td(S)+d(C,S)-\textstyle\sum_{i=1}^{t} d(c_i,b_i)-d(A,B)-2d(B).\end{aligned}$$

There are two cases. Case 1: If $t>2$ then by Lemma 5.6, we have $d(C,S)-\sum_{i=1}^{t} d(c_i,b_i)=d(A,C)+d(B,C)-\sum_{i=1}^{t} d(c_i,b_i) \geq d(A,C)+d(C)$. Furthermore, since $d(S)=d(A)+d(B)+d(A,B)$, we have $2d(S)-d(A,B)-2d(B) \geq d(A)$. Therefore

$$\begin{aligned}&\textstyle\sum_{i=1}^{t} d(S-b_i+c_i)\\ &=td(S)+d(C,S)-\textstyle\sum_{i=1}^{t} d(c_i,b_i)-d(A,B)-2d(B)\\ &\geq (t-2)d(S)+d(A,C)+d(C)+d(A) \geq (t-2)d(S)+d(O).\end{aligned}$$

Case 2: If $t=2$, then since the rank of the matroid is greater than two, $A \neq \emptyset$. Let z be an element in A, then we have

$$\begin{aligned}&2d(S)+d(C,S)-\textstyle\sum_{i=1}^{t} d(c_i,b_i)-d(A,B)-2d(B)\\ &=d(A,C)+d(B,C)-\textstyle\sum_{i=1}^{t} d(c_i,b_i)+2d(A)+d(A,B)\\ &\geq d(A,C)+d(c_1,b_2)+d(c_2,b_1)+d(A)+d(z,b_1)+d(z,b_2)\\ &\geq d(A,C)+d(A)+d(c_1,c_2) \geq d(A,C)+d(A)+d(C)=d(O).\end{aligned}$$

Therefore $\sum_{i=1}^{t} d(S-b_i+c_i) = td(S)+d(C,S)-\sum_{i=1}^{t} d(c_i,b_i)-d(A,B)-2d(B) \geq (t-2)d(S)+d(O)$. This completes the proof. □

Now with the proofs of Lemma 5.5 and Lemma 5.7, we are ready to complete the proof of Theorem 5.1.

PROOF. Since S is a locally optimal solution, we have $\phi(S) \geq \phi(S-b_i+c_i)$ for all i. Therefore, for all i we have $f(S)+\lambda d(S) \geq f(S-b_i+c_i)+\lambda d(S-b_i+c_i)$.

Summing up over all i, we have $tf(S)+\lambda td(S) \geq \sum_{i=1}^{t} f(S-b_i+c_i)+\lambda\sum_{i=1}^{t} d(S-b_i+c_i)$. By Lemma 5.5, we have $tf(S)+\lambda td(S) \geq (t-2)f(S)+f(O)+\lambda\sum_{i=1}^{t} d(S-b_i+c_i)$. By Lemma 5.7, we have $tf(S)+\lambda td(S) \geq (t-2)f(S)+f(O)+\lambda[(t-2)d(S)+d(O)]$. Therefore, $2f(S)+2\lambda d(S)) \geq f(O)+\lambda d(O)$. $\phi(S) \geq \frac{1}{2}\phi(O)$. This completes the proof. □

Theorem 5.1 shows that even in the more general case of a matroid constraint, we can still achieve the approximation ratio of 2. As is standard in such local search algorithms, with a small sacrifice on the approximation ratio, the algorithm can be modified to run in polynomial time by requiring at least an ϵ-improvement at each iteration rather than just any improvement.

6. DYNAMIC UPDATES

In this section, we discuss dynamic updates for the max-sum diversification problem subject to a cardinality constraint with modular set functions. The setting is that we have initially computed a good solution (satisfying the cardinality constraint p) with some approximation guarantee. The weights are changing over time, and upon seeing a change of weight, we want to maintain the quality (the same approximation ratio) of the solution by modifying the current solution without completely recomputing it. We use the number of updates to quantify the amount of modification needed to maintain the desired approximation. An *update* is a single swap of an element in S with an element outside S, where S is the current solution. We ask the following question: ***"Can we maintain a good approximation ratio with a limited number of updates?"*** Since the best known approximation algorithm achieves approximation ratio of 2, it is natural to ask whether it is possible to maintain that ratio through local updates. And if it is possible, how many such updates it requires. To simplify the analysis, we restrict to the following oblivious update rule. Let S be the current solution, and let u be an element in S and v be an element outside S. The marginal gain v has over u with respect to S is defined to be

$$\phi_{v \to u}(S) = \phi(S \setminus \{u\} \cup \{v\}) - \phi(S).$$

OBLIVIOUS (SINGLE ELEMENT SWAP) UPDATE RULE
Find a pair of elements (u, v) with $u \in S$ and $v \notin S$ maximizing $\phi_{v\to u}(S)$. If $\phi_{v\to u}(S) \le 0$, do nothing; otherwise swap u with v.

Since the oblivious local search in Theorem 5.1 uses the same single element swap update rule, it is not hard to see that we can maintain the approximation ratio of 2. However, it is not clear how many updates are needed to maintain that ratio. We conjecture that the number of updates can be made relatively small (i.e., constant) by a non-oblivious update rule and carefully maintaining some desired configuration of the solution set. We leave this as an open question.

However, we are able to show that if we relax the requirement slightly, i.e., aiming for an approximation ratio of 3 instead of 2, and restrict slightly the magnitude of the weight-perturbation, we are able to maintain the desired ratio with a single update. Note that the weight restriction is only used for the case of a weight decrease (Theorem 6.6). We divide weight-perturbations into four types: a weight increase (decrease) which occurs on an element, and a distance increase (decrease) which occurs between two elements. We denote these four types: (I), (II),(III), (IV); and we have a corresponding theorem for each case. Before getting to the theorems, we first prove the following two lemmas. After a weight-perturbation, let S be the current solution set, and O be the optimal solution. Let S^* be the solution set after a single update using the oblivious update rule, and let $\Delta = \phi(S^*) - \phi(S)$. We let $Z = O \cap S$, $X = O \setminus Z$ and $Y = S \setminus Z$.

LEMMA 6.1. *There exists $z \in Y$ such that $\phi_z(S \setminus \{z\}) \le \frac{1}{|Y|}[f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)]$.*

PROOF. If we sum up all marginal gain $\phi_y(S \setminus \{y\})$ for all $y \in Y$, we have $\sum_{y \in Y} \phi_y(S \setminus \{y\}) = f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)$. By an averaging argument, there must exist $z \in Y$ such that $\phi_z(S \setminus \{z\}) \le \frac{1}{|Y|}[f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)]$.

Lemma 6.1 ensures the existence of an element in S such that after removing it from S, the objective function value does not decrease much. The following lemma ensures that there always exists an element outside S which can increase the objective function value substantially if we bring it in.

LEMMA 6.2. *If $\phi(S^*) < \frac{1}{3}\phi(O)$, then for all $y \in Y$, there exists $x \in X$ such that $\phi_x(S \setminus \{y\}) > \frac{1}{|X|}[2\phi(Z) + 3\phi(Y) + 3\lambda d(Z, Y) + 3\Delta]$.*

PROOF. For any $y \in Y$, and by Lemma 4.2, we have
$f(X) + \lambda d(S \setminus \{y\}, X)$
$= f(X) + \lambda d(Z, X) + \lambda d(Y \setminus \{y\}, X)$
$\geq f(X) + \lambda d(Z, X) + \lambda d(X)$.
Note that since $\phi(S^*) = \phi(S) + \Delta < \frac{1}{3}\phi(O)$, we have
$\phi(O) = \phi(Z) + f(X) + \lambda d(X) + \lambda d(Z, X)$
$> 3\phi(Z) + 3\phi(Y) + 3\lambda d(Z, Y) + 3\Delta$. Therefore,
$f(X) + \lambda d(S \setminus \{y\}, X)$
$\geq f(X) + \lambda d(Z, X) + \lambda d(X) \quad > 2\phi(Z) + 3\phi(Y) + 3\lambda d(Z, Y) + 3\Delta$.
This implies there must exist $x \in X$ such that $\phi_x(S \setminus \{y\}) > \frac{1}{|X|}[2\phi(Z) + 3\phi(Y) + 3\lambda d(Z, Y) + 3\Delta]$.

Combining Lemma 6.1 and 6.2, we can give a lower bound for Δ. We have the following corollary.

COROLLARY 6.3. *If $\phi(S^*) < \frac{1}{3}\phi(O)$, then we have $|Y| > 3$ and furthermore $\Delta > \frac{1}{|Y|-3}[2\phi(Z) + 2f(Y) + \lambda d(Y) + 2\lambda d(Z, Y)]$.*

PROOF. By Lemma 6.1, there exists $y \in Y$ such that $\phi_y(S\setminus\{y\}) \leq \frac{1}{|Y|}[f(Y)+2\lambda d(Y)+\lambda d(Z, Y)]$. Since $\phi(S^*) < \frac{1}{3}\phi(O)$, by Lemma 6.2, for this particular y, there exists $x \in X$ such that $\phi_x(S \setminus \{y\}) > \frac{1}{|X|}[2\phi(Z) + 3\phi(Y) + 3\lambda d(Z, Y) + 3\Delta]$. Since $|X| = |Y|$, we have $\Delta > \frac{1}{|Y|}[2\phi(Z) + 2f(Y) + \lambda d(Y) + 2\lambda d(Z, Y) + 3\Delta]$. If $|Y| \leq 3$, then it is a contradiction. Therefore $|Y| > 3$. Rearranging the inequality, we have $\Delta > \frac{1}{|Y|-3}[2\phi(Z) + 2f(Y) + \lambda d(Y) + 2\lambda d(Z, Y)]$.

COROLLARY 6.4. *If $p \leq 3$, then for any weight or distance perturbation, we can maintain an approximation ratio of 3 with a single update.*

PROOF. This is an immediate consequence of Corollary 6.3 since $p = |S| \geq |Y|$. □

Given Corollary 6.4, we will assume $p > 3$ for all the remaining results in this section. We first discuss weight-perturbations on elements.

THEOREM 6.5. [TYPE (I)] *For any weight increase, we can maintain an approximation ratio of 3 with a single update.*

PROOF. Suppose we increase the weight of s by δ. Since the optimal solution can increase by at most δ, if $\Delta \geq \frac{1}{3}\delta$, then we have maintained a ratio of 3. Hence we assume $\Delta < \frac{1}{3}\delta$. If $s \in S$ or $s \notin O$, then it is clear the ratio of 3 is maintained. The only interesting case is when $s \in O \setminus S$. Suppose, for the sake of contradiction, that $\phi(S^*) < \frac{1}{3}\phi(O)$, then by Corollary 6.3, we have $|Y| > 3$ and $\Delta > \frac{1}{|Y|-3}[2\phi(Z) + 2f(Y) + \lambda d(Y)+2\lambda d(Z, Y)]$. Since $\Delta < \frac{1}{3}\delta$, we have $\delta > \frac{1}{|Y|-3}[6\phi(Z)+6f(Y)+3\lambda d(Y)+6\lambda d(Z, Y)]$. On the other hand, by Lemma 6.1, there exists $y \in Y$ such that $\phi_y(S\setminus\{y\}) \leq \frac{1}{|Y|}[f(Y)+2\lambda d(Y) + \lambda d(Z, Y)]$.

Now considering a swap of s with y, the loss by removing y from S is $\phi_y(S\setminus\{y\})$, while the increase that s brings to the set $S \setminus \{y\}$ is at least δ (as s is increased by δ, and the original weight of s is non-negative). Therefore the marginal gain of the swap of s with y is $\phi_{s\to y} \geq \delta - \phi_y(S \setminus \{y\})$ and hence $\phi_{s\to y}(S) \geq \delta - \frac{1}{|Y|}[f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)]$. However, $\phi_{s\to y}(S) \leq \Delta < \frac{1}{3}\delta$. Therefore, we have $\frac{1}{3}\delta > \delta - \frac{1}{|Y|}[f(Y)+2\lambda d(Y)+\lambda d(Z, Y)]$. This implies $\delta < \frac{1}{|Y|}[\frac{3}{2}f(Y) + 3\lambda d(Y) + \frac{3\lambda}{2}d(Z, Y)]$, which is a contradiction.

THEOREM 6.6. [TYPE (II)] *For a weight decrease of magnitude δ, we can maintain an approximation ratio of 3 with $\lceil \log_{\frac{p-2}{p-3}} \frac{w}{w-\delta} \rceil$ updates, where w is the weight of the* *solution before the weight decrease. In particular, if $\delta \leq \frac{w}{p-2}$, we only need a single update.*

PROOF. Suppose we decrease the weight of s by δ. Without loss of generality, we can assume $s \in S$. Suppose, for the sake of contradiction, that $\phi(S^*) < \frac{1}{3}\phi(O)$, then by Corollary 6.3, we have $|Y| > 3$ and $\Delta > \frac{1}{|Y|-3}[2\phi(Z) + 2f(Y) + \lambda d(Y) + 2\lambda d(Z, Y)] \geq \frac{1}{p-3}\phi(S)$. Therefore $\phi(S^*) > \frac{p-2}{p-3}\phi(S)$. This implies that we can maintain the approximation ratio with $\lceil \log_{\frac{p-2}{p-3}} \frac{w}{w-\delta} \rceil$ number of updates. In particular, if $\delta \leq \frac{w}{p-2}$, we only need a single update. □

We now discuss the weight-perturbations between two elements. We assume that such perturbations preserve the metric condition. Furthermore, we assume $p > 3$ for otherwise, by Corollary 6.3, the ratio of 3 is maintained.

THEOREM 6.7. [TYPE (III)] *For any distance increase, we can maintain an approximation ratio of 3 with a single update.*

PROOF. Suppose we increase the distance of (x, y) by δ, and for the sake of contradiction, we assume that $\phi(S^*) < \frac{1}{3}\phi(O)$, then by Corollary 6.3, we have $|Y| > 3$ and $\Delta > \frac{1}{|Y|-3}[2\phi(Z) + 2f(Y) + \lambda d(Y) + 2\lambda d(Z, Y)]$. Since $\Delta < \frac{1}{3}\delta$, we have $\delta > \frac{3}{|Y|-3}[2\phi(Z) + 2f(Y) + \lambda d(Y) + 2\lambda d(Z, Y)] \geq \frac{3}{p-3}\phi(S)$.

If both x and y are in S, then it is not hard to see that the ratio of 3 is maintained. Otherwise, there are two cases:

(1) Exactly one of x and y is in S, without loss of generality, we assume $y \in S$. Considering that we swap x with any vertex $z \in S$ other than y. Since after the swap, both x and y are now in S, by the triangle inequality of the metric condition, we have $\Delta \geq (p-1)\delta - \phi(S) > (\frac{2}{3}p - 2)\delta$. Since $p > 3$, we have $\Delta > (\frac{2}{3}p - 2)\delta \geq \frac{2}{3}\delta > 2\Delta$, which is a contradiction.

(2) Both x and y are outside in S. By Lemma 6.1, there exists $z \in Y$ such that $\phi_z(S \setminus \{z\}) \leq \frac{1}{|Y|}[f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)]$. Consider the set $T = \{x, y\}$ with $S \setminus \{z\}$, by the triangle inequality of the metric condition, we have $d(T, S \setminus \{z\}) \geq (p-1)\delta$. Therefore, at least one of x and y, without loss of generality, assuming x, has the following property: $d(x, S \setminus \{z\}) \geq \frac{(p-1)\delta}{2}$. Considering that we swap x with z, we have: $\Delta \geq \frac{(p-1)}{2}\delta - \frac{1}{|Y|}[f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)]$. Since $\Delta < \frac{1}{3}\delta$, we have $\frac{1}{3}\delta > \frac{(p-1)}{2}\delta - \frac{1}{|Y|}[f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)]$. This implies that $\delta < \frac{6}{3p-5} \cdot \frac{1}{|Y|}[f(Y) + 2\lambda d(Y) + \lambda d(Z, Y)]$. Since $p > 3$, we have $\delta < \frac{1}{|Y|}[\frac{6}{7}f(Y) + \frac{12\lambda}{7}d(Y) + \frac{6\lambda}{7}d(Z, Y)]$, which is a contradiction.

Therefore, $\phi(S^*) \geq \frac{1}{3}\phi(O)$; this completes the proof. □

THEOREM 6.8. [TYPE (IV)] *For any distance decrease, we can maintain an approximation ratio of 3 with a single update.*

PROOF. Suppose we decrease the distance of (x, y) by δ. Without loss of generality, we assume both x and y are in S, for otherwise, it is not hard to see the ratio of 3 is maintained. Suppose, for the sake of contradiction, that $\phi(S^*) < \frac{1}{3}\phi(O)$, then by Corollary 6.3, we have $|Y| > 3$ and $\Delta > \frac{1}{|Y|-3}[2\phi(Z) + 2f(Y) + \lambda d(Y) + 2\lambda d(Z, Y)] \geq \frac{1}{p-3}\phi(S)$. If $\Delta \geq \delta$, then the ratio of 3 is maintained. Otherwise, $\delta > \Delta \geq \frac{1}{p-3}\phi(S)$. By the triangle inequality of the metric condition, we have $\phi(S) \geq (p-2)\delta > \frac{p-2}{p-3}\phi(S) > \phi(S)$, which is a contradiction. □

Combining Theorem 6.5, 6.6, 6.7, 6.8, we have the following corollary.

Table I: Comparison of Greedy A and Greedy B ($N = 50$, Synthetic Data)

p	OPT	$GreedyA$	$GreedyB$	LS	$AF_{GreedyA}$	$AF_{GreedyB}$	AF_{LS}	$AF_{\frac{GreedyB}{GreedyA}}$	$AF_{\frac{LS}{GreedyB}}$
3	4.888	4.295	4.658	4.856	1.138	1.049	1.006	1.084	1.042
4	7.817	7.553	7.632	7.802	1.034	1.024	1.001	1.010	1.022
5	11.274	10.353	10.943	11.255	1.088	1.030	1.001	1.056	1.028
6	15.249	14.49	14.971	15.221	1.052	1.018	1.001	1.033	1.016
7	19.563	18.385	19.211	19.501	1.064	1.018	1.003	1.044	1.015

Table II: Comparison of Greedy A, Greedy B and LS ($N = 500$, Synthetic Data Set)

p	$GreedyA$	$GreedyB$	LS	$AF_{\frac{GreedyB}{GreedyA}}$	$AF_{\frac{LS}{GreedyB}}$	$Time_{GreedyA}$	$Time_{GreedyB}$	$Time_{\frac{GreedyA}{GreedyB}}$
5	10.49	11.526	11.829	1.099	1.026	4637 ms	15 ms	309.133
10	37.63	38.67	39.135	1.028	1.012	6045 ms	46 ms	131.413
15	77.27	80.345	81.1	1.039	1.009	6919 ms	100 ms	69.19
20	133.82	138.709	139.21	1.036	1.003	9067 ms	195 ms	46.497
25	204.05	210.622	211.25	1.032	1.002	10157 ms	331 ms	30.685
30	292.12	298.2	299	1.020	1.002	11361 ms	488 ms	23.280
35	391.14	399	401.04	1.020	1.005	12247 ms	582 ms	21.042
40	507.12	516.99	518	1.019	1.001	13833 ms	751 ms	18.419
45	639.12	650.51	651.18	1.017	1.001	14591 ms	938 ms	15.555
50	779.74	797.25	798.53	1.022	1.001	16849 ms	1304 ms	12.921
55	944.90	962.24	963.04	1.018	1.000	17117 ms	1500 ms	11.411
60	1117.37	1137.18	1138.27	1.017	1.000	18399 ms	1931 ms	9.528
65	1309.96	1333.53	1334.8	1.017	1.000	18439 ms	2134 ms	8.640
70	1514.75	1537.21	1538.58	1.014	1.000	19171 ms	2441 ms	7.853
75	1733.68	1761.57	1762.854	1.016	1.000	36305 ms	4427 ms	8.200

COROLLARY 6.9. *If the initial solution achieves approximation ratio of 3, then for any weight-perturbation of* TYPE (I), (III), (IV)*; and any weight-perturbation of* TYPE (II) *that is no more than* $\frac{1}{p-2}$ *of the current solution for* $p > 3$ *and arbitrary for* $p \leq 3$*, we can maintain the ratio of 3 with a single update.*

7. EXPERIMENTS

While we emphasize that the results in this paper are mainly theoretical in nature, we present some experimental results in this section to provide additional insight about the relative performance and efficiency of our algorithms. More specifically, we wish to understand the differences between the two types of greedy algorithms (i.e. incrementally adding edges vs incrementally adding vertices) and how much local search can improve upon such greedy algorithms. To the extent that we can determine optimal solutions (i.e. for small problem instances), we want to understand how well these conceptually simple algorithms approximate optimality in a sense that goes beyond worst case analysis. All of our experiments are with respect to various cardinality constraints p.

In section 7.1, we will first consider the relative performance of two greedy algorithms and local search with respect to a synthetic data set. In section 7.2, we introduce two small algorithmic improvements (one for each of the greedy algorithms) that do not impact the approximation ratios but allow for a fairer comparison of the algorithms. This is followed in section 7.3 by experiments for a well-known dataset (LETOR) that has been actively used for different information and machine learning problems and especially for ”learn to rank” research [Qin et al. 2010]. In section 7.4, we again consider the synthetic data set as in section 7.1 and make some observations on the performance of local search for dynamically changing data.

For the synthetic data as well as the LETOR data set, we consider the max-sum diversification problem with modular set functions and a cardinality constraint p so as to be able to compare the greedy and local search algorithms as well as comparing our greedy algorithm with the algorithm of Gollapudi and Sharma [2009] whose work motivated this paper. We will refer to their diversification algorithm as Greedy A. We recall that their algorithm consists of a reduction to the max-sum p-dispersion problem and then uses the Hassin, Rubenstein and Tamir [1997] algorithm that greedily

Table III: Comparison of (Improved) Greedy A, Greedy B and LS ($N = 50$, LETOR Data Set)

p	OPT	$GreedyA$	$GreedyB$	LS	$AF_{GreedyA}$	$AF_{GreedyB}$	AF_{LS}	$AF_{\frac{GreedyB}{GreedyA}}$	$AF_{\frac{LS}{GreedyB}}$
3	7.088	6.140	7.088	7.088	1.154	1.000	1.000	1.154	1.000
4	10.020	10.020	10.000	10.020	1.000	1.002	1.000	0.998	1.002
5	12.571	12.470	12.570	12.571	1.008	1.000	1.000	1.008	1.000
6	15.315	15.060	15.060	15.315	1.017	1.017	1.000	1.000	1.016
7	18.540	17.290	17.949	18.540	1.072	1.033	1.000	1.038	1.032

Table IV: Comparison of (Improved) Greedy A, Greedy B and LS ($N = 370$, LETOR Data Set)

p	$GreedyA$	$GreedyB$	LS	$AF_{\frac{GreedyB}{GreedyA}}$	$AF_{\frac{LS}{GreedyB}}$	$Time_{GreedyA}$	$Time_{GreedyB}$	$Time_{\frac{GreedyA}{GreedyB}}$
5	13.996	13.999	13.999	1.000	1.000	2365 ms	426 ms	5.552
10	37.570	37.970	37.970	1.011	1.000	2370 ms	504 ms	4.702
15	69.590	71.600	71.600	1.029	1.000	2694 ms	421 ms	6.399
20	110.900	113.640	113.640	1.025	1.000	3280 ms	470 ms	6.979
25	154.590	162.400	162.480	1.051	1.000	3223 ms	587 ms	5.491
30	192.260	220.450	220.730	1.147	1.001	4364 ms	785 ms	5.559
35	253.790	288.490	288.970	1.137	1.002	4762 ms	758 ms	6.282
40	317.290	366.520	367.215	1.155	1.002	4599 ms	864 ms	5.323
45	397.230	454.500	455.100	1.144	1.001	6088 ms	1028 ms	5.922
50	486.440	552.500	553.150	1.136	1.001	5323 ms	1155 ms	4.609
55	584.830	660.430	661.370	1.129	1.001	7360 ms	1536 ms	4.792
60	686.970	778.140	779.220	1.133	1.001	5585 ms	1684 ms	3.317
65	805.520	905.660	906.880	1.124	1.001	7349 ms	1855 ms	3.962
70	930.600	1042.970	1044.120	1.121	1.001	5381 ms	2041 ms	2.636
75	1054.940	1189.970	1191.360	1.128	1.001	8480 ms	2212 ms	3.834

Table V: Comparison of (Improved) Greedy A, Greedy B and LS ($N = 50$, LETOR Data Set, Avg. over 5 Queries)

p	$AF_{GreedyA}$	$AF_{GreedyB}$	AF_{LS}	$AF_{\frac{GreedyB}{GreedyA}}$	$AF_{\frac{LS}{GreedyB}}$
3	1.030	1.000	1.000	1.03	1.000
4	1.009	1.004	1.000	1.004	1.004
5	1.020	1.012	1.000	1.007	1.012
6	1.059	1.018	1.000	1.04	1.018
7	1.096	1.022	1.000	1.07	1.022

chooses *edges* yielding an approximation ratio of 2. We will experimentally compare the performance and time complexity of their algorithm against our greedy by vertices algorithm which also has approximation ratio 2. We will refer to our greedy algorithm as Greedy B. We also consider how much a limited amount of local search improves the results obtained by our Greedy B algorithm. That is, we follow Greedy B by a 1-swap local search algorithm that searches for *any* improvement in each iteration. We refer to this local search algorithm as LS with the understanding that it is being initialized by Greedy B and terminated when either a local maximum is reached or when the algorithm runs for ten times the time of the Greedy B initialization. More precisely, the elapsed time is polled after each possible swap is considered and the algorithm terminates once this time exceeds 10 times the running time of the Greedy B running time.

7.1. Experiments with synthetic data sets

Our synthetic data sets are generated by uniformly at random assigning each vertex v (i.e. element of the metric space) a value $f(v) \in [0, 1]$, and for each distance $d(u, v)$ assigning a value in [1,2]. We note that the {1,2} metric is the metric relative to which the suggested hardness of approximation is derived. We construct such data sets for various values of N, the size of the universe, and for p, the cardinality constraint. In all cases, we set $\lambda = .2$, where λ is the parameter defining the relative weight between

Table VI: Comparison of (Improved) Greedy A, Greedy B and LS ($N = 370$, LETOR Data Set, Avg. over 5 Queries)

p	$AF_{\frac{GreedyB}{GreedyA}}$	$AF_{\frac{LS}{GreedyB}}$	$Time_{GreedyA}$	$Time_{GreedyB}$	$Time_{\frac{GreedyA}{GreedyB}}$
5	1.005	1	1714	303	5.657
10	1.016	1	1997	289	6.910
15	1.036	1	2387	381	6.265
20	1.056	1.002	2767	522	5.301
25	1.047	1.003	3280	574	5.714
30	1.086	1.003	2959	537	5.510
35	1.081	1.003	3387	622	5.445
40	1.105	1.003	3208	704	4.557
45	1.119	1.002	4154	837	4.963
50	1.146	1.002	4126	1035	3.986
55	1.141	1.002	5559	1298	4.283
60	1.156	1.002	5059	1411	3.585
65	1.152	1.002	5722	1534	3.730
70	1.157	1.002	4766	1691	2.818
75	1.151	1.001	7272	2180	3.336

the quality $f(S)$ of a set S and its max-sum dispersion $d(S) = \sum_{u,v \in S} d(u, v)$. For small N, we can compute the optimal value and can therefore compute and compare the experimental approximation ratios for Greedy A, Greedy B and LS.

In Table 1 (resp. Table 2), we present results on the relative performance and time elapsed for Greedy A, Greedy B, and LS for $N = 50$ (resp. $N = 500$). For each setting of the N, p parameters we ran 5 trials and averaged the results. We observe these average values for each parameter setting for an algorithm ALG, and report the "observed average approximation ratio", namely $\frac{OPT-average}{ALG-average}$, denoted AF_{ALG} for the $N = 50$ data where we are able to compute the optimum value. Similarly, we denote the "relative average approximation" between two algorithms as $AF_{\frac{ALG_2}{ALG_1}}$. We also report the average time elapsed [6] for each algorithm, denoted as T_{ALG}. We make the following observations based on these trials:

- In all cases, the Greedy algorithms and LS perform quite well with regard to the optimum (when it is computed); this is not surprising as it is often the case that algorithms perform well for random or "real" data in contrast to worst case approximation ratios. More specifically, for $N = 50$ and $p \leq 7$, the approximation ratio for Greedy B ranges (roughly) between 1.02 and 1.05 while the approximation ratio for LS ranges between 1.002 and 1.007.
- As expected, the time bounds for Greedy B are substantially better than for Greedy A as Greedy B is iterating over all vertices rather than over all edges as in Greedy A.
- In all cases (for average performance), Greedy B outperforms Greedy A. For $N = 500$, the relative improvement appears generally to be decreasing as p increases, where for the largest values of $p = 70$ and 75, the relative improvement is roughly 1.5%. We observed in our experiments that the relative improvement was 2.5% if one just compared the dispersion results $d()$.
- As we would expect, local search (as defined to always stop within 10 times the running time of Greedy B) can sometimes improve upon the results of Greedy B.

[6] The time is reported in milliseconds (ms), with algorithms implemented in Java running on a Macbook Pro with 2.4 GHz Intel Core i7 processor and 8 GB 1600 MHz DDR3 memory.

Table VII: Comparison of documents being returned by (Improved) Greedy A, Greedy B and LS for the top 50 document LETOR data set

Greedy A	Greedy B	LS	OPT	N	p
4	4	4	4	50	3
29	29	29	29		
46	24	24	24		

Greedy A	Greedy B	LS	OPT	N	p
4	4	4	4	50	4
29	29	29	29		
24	24	24	24		
12	12	12	12		

Greedy A	Greedy B	LS	OPT	N	p
4	4	4	4	50	5
29	29	29	29		
24	24	24	24		
12	12	12	12		
46	49	49	49		

Greedy A	Greedy B	LS	OPT	N	p
4	4	4	4	50	6
29	29	29	29		
24	24	24	24		
12	12	12	12		
46	46	46	46		
49	49	35	**35**		

Greedy A	Greedy B	LS	OPT	N	p
4	4	4	4	50	7
29	29	29	29		
24	24	24	24		
12	12	12	12		
0	**49**	**37**	**37**		
8	46	46	46		
14	35	35	35		

For the small data set, the improvement of LS ranges (roughly) between 2% and 4%. For the large data set, as p increases, LS becomes less effective. In particular, for $p \geq 40$, LS improves upon Greedy B by at most .1%.

7.2. Improving Greedy A and Greedy B

The performance of Greedy A for odd values of p is marred by the fact, that as defined, Greedy A chooses an arbitrary last vertex rather than the best last vertex. For larger p, this does not have a significant impact but it is perhaps best to ignore small odd values of p. The performance of Greedy B is marred by the fact, that as defined, it chooses its first vertex arbitrarily rather than choosing a best pair. Our results for average performance raises the question as to whether or not Greedy B might outperform Greedy A for all inputs, that is, for all parameter settings. In order to make the comparison fair, for Greedy A we will choose the best final node rather than an arbitrary node when p is odd, and for Greedy B, we will start with the best pair of nodes rather than an arbitrary node. These minor changes do not effect the approximation ratios but can improve the observed performance of the algorithms. Using these improved greedy algorithms we found one trial (for $N = 50, p = 4$) where Greedy A outperformed Greedy B. While running the algorithms with these improvements does not alter the basic observations above, we will hereafter use the improved greedy algorithms for the LETOR data set experiments that now follow.

7.3. Experiments with the LETOR datasets

The LETOR datasets are well known datasets that have been mainly used for research on learning to rank problems. For our experiments, we use MSLR-WEB10K[7] which is a random sampling of 10,000 queries. Each item in a LETOR data set represents a document related to a query. The relevance judgments are obtained from a retired

[7] http://research.microsoft.com/en-us/projects/mslr/download.aspx

labeling set of a commercial web search engine (Microsoft Bing), which take 5 values from 0 (irrelevant) to 4 (perfectly relevant). The features are basically extracted by the provider of the LETOR dataset, and are those widely used in the research community. As such, each item u has an integral relevance score $r(u)$ (relative to the query) ranging from 0 to 5, a set of feature attributes with their respective values, and a query id. Thus, we take (as ground truth), the quality score $f(S) = \sum_{u \in S} r(u)$. We define (and take as ground truth) a metric distance $f(u, v)$ function given by the cosine similarity between the feature vectors for u and v. For Table III and Table IV, we chose one data set (chosen at random from the original LETOR dataset) and created a data set consisting of the top (by relevance score) 50 and top 370 documents. We applied the (Improved) Greedy A, B and limited local search algorithms to these two data sets for various settings of the cardinality parameter p. For the smaller 50 document data set we also computed the optimal values. We observe some qualitative differences between these “real data” experiments and the experiments for synthetic data.

- For the small data set, while the (Improved) Greedy A is slightly better for $p = 4$, (Improved) Greedy B is better in the other cases with a reasonably substantial advantage for $p = 3$ and $p = 7$.
- LS is able to find an optimal solution for *all* the small data set experiments.
- For the larger 370 document set, in contrast to the synthetic data experiments, the advantage of (Improved) Greedy B over Greedy A is more pronounced for larger values of p, the cardinality constraint. The advantage of Greedy B over Greedy A rises to about 15% and then levels off at around 12%.
- For the larger data set, the improvement due to local search never exceeds .3%.

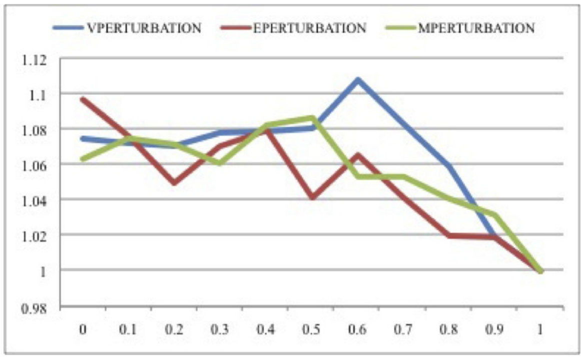
We also ran 5 different data sets (i.e. generated by 5 different queries) and averaged the results with respect to both the top 50 results and the top 370 results as shown in Table V and Table VI respectively. Note that in these tables, we are omitting the objective function values that have been previously included in other tables. We are averaging our results over different LETOR datasets (i.e. queries) and therefore reporting on the average objective function values wont be fully meaningful. These average results support what we found in Table III and Table IV, namely that (Improved) Greedy B significantly outperforms (Improved) Greedy A and that limited local search provides a very small advantage over (Improved) Greedy B. In Table VII, we present the difference in the documents being returned for the 50 document data set. Here the OPT documents are the true set of optimal documents with respect to the diversification function applied to the values of the document relevance scores and the cosine distance function. As an example, consider the results for the $N = 50, p = 7$ setting of the parameters. Here OPT and (Improved) Greedy B differ on one document while (Improved) Greedy A differs on 3 documents.

7.4. Approximation Ratio in Dynamic Updates

For dynamic updates, we use same synthetic data as in Section 7.1. We have three different dynamically changing environments:

(1) VPERTURBATION: each perturbation is a weight change on an item; that is, an item (vertex) u is randomly chosen and its value is reset uniformly at random from $[0, 1]$.
(2) EPERTURBATION: each perturbation is a distance change between two items; that is, a pair of distinct items $\{u, v\}$ is randomly chosen and the distance $d(u, v)$ is reset uniformly at random from $[1, 2]$.
(3) MPERTURBATION: each perturbation is one of the above two with equal probability.

Fig. 1: Approximation Ratio in Dynamic Updates



For each of the environments above and every value of λ, we start with our greedy solution (a 2-approximation) and run 20 steps of simulation, where each step consists of a random weight change of the stated type, followed by a single application of the oblivious update rule. We repeat this 100 times and record the worst approximation ratio occurring during these 100 updates. The results are shown in Fig. 1; the horizontal axis measures λ values, and the vertical axis measures the approximation ratio.

We have the following observations:

(1) In any dynamically changing environment, the maintained ratio is well below the provable ratio of 3. The worst observed ratio is about 1.11.
(2) The maintained ratios are decreasing to 1 for increasing $\lambda \geq 0.6$.

From the experiment, we see that the simple local search update rule seems effective for maintaining a good approximation ratio in a dynamically changing environment.

8. CONCLUSION

We study the max-sum diversification with monotone submodular set functions and give a natural 2-approximation greedy algorithm for the problem when there is a cardinality constraint. We further extend the problem to matroid constraints and give a 2-approximation local search algorithm for the problem. We examine the dynamic update setting for modular set functions, where the weights and distances are constantly changing over time and the goal is to maintain a solution with good quality with a limited number of updates. We propose a simple update rule: the oblivious (single swap) update rule, and show that if the weight-perturbation is not too large, we can maintain an approximation ratio of 3 with a single update. The diversification problem has many important applications and there are many interesting future directions. Although in this paper we restricted ourselves to the max-sum objective, there are many other well-defined notion of diversity that can be considered, see for example [Chandra and Halldórsson 1996] and [Gollapudi and Sharma 2009]. The max-sum case can be also studied for specific metrics such as the ℓ_1-norm in Euclidean space as considered by Fekete and Meijer [2003] who provide a linear time optimal algorithm for constant p and a PTAS when p is part of the input. Their PTAS algorithm also provides a $(2+\epsilon)$-approximation for the ℓ_2-norm. Their algorithms exploit the geometric nature of the metric space. Other specific metric spaces are also of interest.

In the general matroid case, the greedy algorithm given in Section 4 fails to achieve any constant approximation ratio, but one can also consider other “greedy-like algorithms” such as the partial enumeration greedy method used (for example) successfully for monotone submodular maximization subject to a knapsack constraint in Sviridenko

[2004]? Can such a technique also be used to provide an approximation for our diversification problem? Can our results be extended to provide a constant approximation for the diversification problem subject to a knapsack constraint? In a dynamic update setting, we only considered the diversification problem for a modular set function subject to a cardinality constraint. Here, we used the oblivious single swap update rule to maintain a 3-approximation. It is interesting to see if it is possible to maintain a better ratio than 3 with a limited number of updates, by larger cardinality swaps, and/or by a non-oblivious update rule. We leave this as an open question. It is also an open question as to whether we can maintain a good solution subject to an arbitrary matroid constraint. The approximation ratio and application of diversification maximization in a distributed setting is pursued in a recent paper by Abbasi-Zadeh et al [2017].

Finally, a crucial property used throughout our results is the triangle inequality. In our conference paper [Borodin et al. 2012], we asked the question as to whether we can relate the approximation ratio to the parameter of a relaxed triangle inequality? Sydow [2014] provides a partial answer to this question showing that the matching based algorithm of Hassin et al [1997] can be applied to an $\alpha \geq 1$ relaxed metric (where $d(x, y) + d(y, z) \geq \alpha d(x, z)$) resulting in a (tight) $\frac{2}{\alpha}$ approximation ratio for the cardinality constrained max-sum dispersion problem. Independently, Abbasi-Zadeh and Ghadiri [2015] obtain the $\frac{2}{\alpha}$ approximation ratio for the cardinality constraint, and a $\frac{2}{\alpha^2}$ approximation ratio for an arbitrary matroid constraint.

REFERENCES

Zeinab Abbassi, Vahab S. Mirrokni, and Mayur Thakur. 2013. Diversity maximization under matroid constraints. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. 32–40.

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *WSDM*. 5–14.

Noga Alon. 2014. (2014). Personal communication.

Noga Alon, Sanjeev Arora, Rajsekar Manoikaran, Dana Moshkovitz, and Omri Weinstein. 2011. Inapproximability of Densest κ-Subgraph from Aveage Case Harsdness. (2011). Unpublished manuscript.

Nikhil Bansal, Kamal Jain, Anna Kazeykina, and Joseph Naor. 2010. Approximation Algorithms for Diversified Search Ranking. In *ICALP (2)*. 273–284.

Benjamin E. Birnbaum and Kenneth J. Goldman. 2009. An Improved Analysis for a Greedy Remote-Clique Algorithm Using Factor-Revealing LPs. *Algorithmica* 55, 1 (2009), 42–59.

Alan Borodin, Dai Le, and Yuli Ye. 2014. Weakly Submodular Functions. *CoRR* abs/1401.6697 (2014).

Allan Borodin, Hyun Chul Lee, and Yuli Ye. 2012. Max-Sum diversification, monotone submodular functions and dynamic updates. In *ACM Symposium on Principles of Database Systems (PODS)*. 155–166.

Christina Brandt, Thorsten Joachims, Yisong Yue, and Jacob Bank. 2011. Dynamic ranked retrieval. In *WSDM*. 247–256.

Richard A. Brualdi. 1969. Comments on Bases in Dependence Structures. *Bulletin of the Australian Mathematical Society* 1, 02 (1969), 161–167.

Gruia Călinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. Maximizing a Monotone Submodular Function Subject to a Matroid Constraint. *SIAM J. Comput.* 40, 6 (2011), 1740–1766.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '98)*. ACM, New York, NY, USA, 335–336.

Barun Chandra and Magnús M. Halldórsson. 1996. Facility Dispersion and Remote Subgraphs. In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory*. Springer-Verlag, London, UK, 53–65. http://portal.acm.org/citation.cfm?id=645898.756652

Barun Chandra and Magnús M. Halldórsson. 2001. Approximation Algorithms for Dispersion Problems. *J. Algorithms* 38, 2 (2001), 438–465.

Harr Chen and David R. Karger. 2006. Less is more: probabilistic models for retrieving fewer relevant documents. In *SIGIR*. 429–436.

Andrzej Czygrinow. 2000. Maximum dispersion problem in dense graphs. *Oper. Res. Lett.* 27, 5 (2000), 223–227.

Elena Demidova, Peter Fankhauser, Xuan Zhou, and Wolfgang Nejdl. 2010. DivQ: diversification for keyword search over structured databases. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10)*. ACM, 331–338.

Zhicheng Dou, Sha Hu, Kun Chen, Ruihua Song, and Ji-Rong Wen. 2011. Multi-dimensional search result diversification. In *WSDM*. 475–484.

Marina Drosou and Evaggelia Pitoura. 2009. Diversity over Continuous Data. *IEEE Data Eng. Bull.* 32, 4 (2009), 49–56.

Marina Drosou and Evaggelia Pitoura. 2010. Search result diversification. *SIGMOD Record* 39, 1 (2010), 41–47.

Erhan Erkut. 1990. The discrete p-dispersion problem. *European Journal of Operational Research* 46, 1 (May 1990), 48–60.

Erhan Erkut and Susan Neuman. 1989. Analytical models for locating undesirable facilities. *European Journal of Operational Research* 40, 3 (June 1989), 275–291.

Sándor P. Fekete and Henk Meijer. 2003. Maximum Dispersion and Geometric Maximum Weight Cliques. *Algorithmica* 38, 3 (2003), 501–511.

M. Fisher, G. Nemhauser, and L. Wolsey. 1978. An analysis of the approximations for maximizing submodular set functions-II. *Mathematical Programming Studies* 8 (1978), 73–87.

Sreenivas Gollapudi and Aneesh Sharma. 2009. An axiomatic approach for result diversification. In *World Wide Web Conference Series*. 381–390.

Magnús M. Halldórsson, Kazuo Iwano, Naoki Katoh, and Takeshi Tokuyama. 1995. Finding subsets maximizing minimum structures. In *Symposium on Discrete Algorithms*. 150–159. DOI:http://dx.doi.org/10.1145/313651.313683

Refael Hassin, Shlomi Rubinstein, and Arie Tamir. 1997. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.* 21, 3 (1997), 133–137.

David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03)*. ACM, New York, NY, USA, 137–146.

Sanjeev Khanna, Rajeev Motwani, Madhu Sudan, and Umesh V. Vazirani. 1998. On Syntactic versus Computational Views of Approximability. *SIAM J. Comput.* 28, 1 (1998), 164–191.

Subhash Khot. 2006. Ruling Out PTAS for Graph Min-Bisection, Dense k-Subgraph, and Bipartite Clique. *SIAM J. Comput.* 36, 4 (2006), 1025–1071.

Michael J. Kuby. 1987. Programming Models for Facility Dispersion: The p-Dispersion and Maxisum Dispersion Problems. *Geographical Analysis* 19, 4 (1987), 315–329.

Hui Lin and Jeff Bilmes. 2010. Multi-document Summarization via Budgeted Maximization of Submodular Functions. In *HLT-NAACL*. 912–920.

Hui Lin and Jeff Bilmes. 2011. A Class of Submodular Functions for Document Summarization. In *North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2011)*. Portland, OR. (long paper).

Hui Lin, Jeff Bilmes, and Shasha Xie. 2009. Graph-based Submodular Selection for Extractive Summarization. In *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU)*. Merano, Italy.

Ziyang Liu, Peng Sun, and Yi Chen. 2009. Structured Search Result Differentiation. *PVLDB* 2, 1 (2009), 313–324.

Raghu Meka, Aaron Potechin, and Avi Wigderson. 2015. Sum-of-squares Lower Bounds for Planted Clique. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*. 87–96.

Enrico Minack, Wolf Siberski, and Wolfgang Nejdl. 2011. Incremental diversification for very large sets: a streaming-based approach. In *SIGIR*. 585–594.

G. Nemhauser, L. Wolsey, and M. Fisher. 1978. An analysis of the approximations for maximizing submodular set functions-I. *Mathematical Programming* 14 (1978), 265–294.

Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. 2010. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.* 13, 4 (2010), 346–374.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *ICML*. 784–791.

Davood Rafiei, Krishna Bharat, and Anand Shukla. 2010. Diversifying web search results. In *WWW*. 781–790.

S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. 1994. Heuristic and Special Case Algorithms for Dispersion Problems. *Operations Research* 42, 2 (March-April 1994), 299–310.

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2011. Intent-aware search result diversification. In *SIGIR*. 595–604.

Alexander Schrijver. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer.

Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi. 2010. Learning optimally diverse rankings over large document collections. In *ICML*. 983–990.

Maxim Sviridenko. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.* 32, 1 (2004), 41–43.

Marcin Sydow. 2014. Improved Approximation Guarantee for Max Sum Diversification with Parameterised Triangle Inequality. In *Foundations of Intelligent Systems - 21st International Symposium, ISMIS 2014, Roskilde, Denmark, June 25-27, 2014. Proceedings*. 554–559.

Marcos R. Vieira, Humberto Luiz Razente, Maria Camila Nardini Barioni, Marios Hadjieleftheriou, Divesh Srivastava, Caetano Traina Jr., and Vassilis J. Tsotras. 2011a. DivDB: A System for Diversifying Query Results. *PVLDB* 4, 12 (2011), 1395–1398.

Marcos R. Vieira, Humberto Luiz Razente, Maria Camila Nardini Barioni, Marios Hadjieleftheriou, Divesh Srivastava, Caetano Traina Jr., and Vassilis J. Tsotras. 2011b. On query result diversification. In *ICDE*. 1163–1174.

D. W. Wang and Yue-Sun Kuo. 1988. A study on two geometric location problems. *Inf. Process. Lett.* 28 (August 1988), 281–286. Issue 6.

Cong Yu, Laks Lakshmanan, and Sihem Amer-Yahia. 2009. It takes variety to make a world: diversification in recommender systems. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT '09)*. 368–378.

Yisong Yue and Thorsten Joachims. 2008. Predicting diverse subsets using structural SVMs. In *ICML*. 1224–1231.

Sepehr Abbasi Zadeh and Mehrdad Ghadiri. 2015. Max-Sum Diversification, Monotone Submodular Functions and Semi-metric Spaces. *CoRR* abs/1511.02402 (2015).

Sepehr Abbasi Zadeh, Mehrdad Ghadiri, Vahab S. Mirrokni, and Morteza Zadimoghaddam. 2017. Scalable Feature Selection via Distributed Diversity Maximization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.* 2876–2883.

ChengXiang Zhai, William W. Cohen, and John D. Lafferty. 2003. Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In *SIGIR*. 10–17.

Feng Zhao, Xiaolong Zhang, Anthony K. H. Tung, and Gang Chen. 2011. BROAD: Diversified Keyword Search in Databases. *PVLDB* 4, 12 (2011), 1355–1358.

Xiaojin Zhu, Andrew B. Goldberg, Jurgen Van Gael, and David Andrzejewski. 2007. Improving Diversity in Ranking using Absorbing Random Walks. In *HLT-NAACL*. 97–104.

A. THE GREEDY ALGORITHM APPLIED TO DIVERSIFICATION WITH A MATROID CONSTRAINT

We observe that for the more general matroid constraint diversification problem, the greedy algorithm in section 4 no longer achieves any constant approximation ratio. More specifically, consider the max-sum diversifciation problem as in Gallopudi and Sharma [Gollapudi and Sharma 2009] (that is, for a modular quality function $f()$) but now subject to a partition matroid constraint. Partition the universe into $A = \{a, b\}$ with cardinality constraint 1 and $C = \{c_1, c_2, \ldots c_r\}$ with no cardinality constraint. Let the objective be $f(S) = \sum_{u \in S} q_u + \sum_{u,v \in S} d(u, v)$ where the quality and distance functions are defined as follows: $q(a) = \ell + \epsilon$, $q(x) = 0$ for all $x \neq a$, and for all x, $d(b, x) = \ell$, $d(u, x) = \epsilon$ for all $u \neq b$. The greedy algorithm (starting with a or with the best pair (a, b) will yield $f(S) = f(C \cup \{a\}) = \ell + \epsilon + \epsilon \cdot \binom{r}{2} + r\epsilon$ while the optimal solution will be $f(C \cup \{b\}) = r \cdot \ell + \epsilon \cdot \binom{r}{2}$. Hence the approximation can be made arbitrarily bad by choosing $\epsilon = \frac{1}{\binom{r}{2}}$ and taking r sufficiently large.

By the reduction in [Gollapudi and Sharma 2009] to the metric dispersion problem, the above example shows that the greedy algorithm will also suffer the same unbounded approximation ratio for the metric dispersion problem.