Parallel Quasi-concave set optimization: A new frontier that scales without needing submodularity

Praneeth Vepakomma [1] Yulia Kempner [2] Ramesh Raskar [1]

Abstract

Classes of set functions along with a choice of ground set are a bedrock to determine and develop corresponding variants of greedy algorithms to obtain efficient solutions for combinatorial optimization problems. The class of approximate constrained submodular optimization has seen huge advances at the intersection of good computational efficiency, versatility and approximation guarantees while exact solutions for unconstrained submodular optimization are NP-hard. What is an alternative to situations when submodularity does not hold? Can efficient and globally exact solutions be obtained? We introduce one such new frontier: The class of quasi-concave set functions induced as a dual class to monotone linkage functions. We provide a parallel algorithm with a time complexity over n processors of $\mathcal{O}(n^2 g) + \mathcal{O}(\log \log n)$ where n is the cardinality of the ground set and g is the complexity to compute the monotone linkage function that induces a corresponding quasi-concave set function via a duality. The complexity reduces to $\mathcal{O}(gn\log(n))$ on n^2 processors and to $\mathcal{O}(gn)$ on n^3 processors. Our algorithm provides a globally optimal solution to a maxi-min problem as opposed to submodular optimization which is approximate. We show a potential for widespread applications via an example of diverse feature subset selection with exact global maxi-min guarantees upon showing that a statistical dependency measure called distance correlation can be used to induce a quasi-concave set function.

1. Introduction

The rich structure of some set function classes allows for development of efficient algorithms for combinatorial optimization problem. To be formal, a set system $(F, \mathcal{Z})$ is a collection F of subsets of a ground set $\mathcal{Z}$. For example F could be subsets of the power set of $\mathcal{Z}$ or could be subsets that satisfy the structure of a greedoid (Korte et al., 2012), semi-lattice (Chajda et al., 2007), independence systems(Conforti & Laurent, 1989) or an antimatroid(Dietrich, 1989; Kempner & Levit, 2003; Algaba et al., 2004) and so forth.

Popular set function classes such as submodular functions (Lovász, 1983; Edmonds, 2003; Nemhauser et al., 1978; Fujishige, 2005; Feige et al., 2011; Krause & Golovin, 2014; Iyer & Bilmes, 2013) have resulted in a wide array of powerful algorithms for several tasks across different fields.

Under lack of submodularity, relaxations that characterize approximate submodularity, (Bian et al., 2017; Bogunovic et al., 2018; Horel & Singer, 2016; Chierichetti et al., 2020; Das & Kempe, 2018) have been introduced to develop combinatorial algorithms with approximation guarantees. Other set function classes beyond submodularity include those of subadditive functions, quasi-submodular functions and the lesser known class of induced quasi-concave set functions that is relevant to this paper.

This paper introduces a parallel algorithm for optimizing quasi-concave set functions with global optimality guarantees as opposed to submodular optimization that provides approximate solutions. Algorithms for optimizing general quasi-concave set functions do not exist, while a specific sub class of quasi-concave set functions that can be written in terms of monotone linkage functions can be optimized to obtain globally optimal solutions. As an example, we show that certain monotone linkage functions of distance covariance induce a corresponding quasi-concave set function. We use our algorithm to find an optimally diverse set of features based on distance covariance.

1.1. Preliminaries

We now list the definition of *quasi-concave set functions* and state the *induced quasi-concave set function optimization problem* which are central to the focus of this paper.

2. Quasi-concave set functions

Definition 2.1 (**Quasi-Concave Set Function** (Mullat, 1976; Kuznecov et al., 1985; Zaks & Muchnik, 1989; Vepakomma & Kempner, 2019)**).** A function $F : \mathcal{F} \mapsto \mathbb{R}$ defined on a set system $(\mathbf{X}, \mathcal{F})$ is quasi-concave if for each

$\mathbf{S}, \mathbf{T} \in \mathcal{F}$,

$$F(\mathbf{S} \cap \mathbf{T}) \geq \min\{F(\mathbf{S}), F(\mathbf{T})\} \quad (1)$$

Connection: We would like to note its notational similarity to its continuous counter-part of strictly quasi-concave functions which are those real-valued functions defined on any convex subset of real-valued vector spaces such that $f(\lambda x + (1-\lambda)y) \geq \min\{f(x), f(y)\}$ for all $x \neq y$ and $\lambda \in (0,1)$.

We denote the set $2^{\mathbf{X}} \setminus \{\phi, \mathbf{X}\}$ by $\mathcal{P}^{-X}$ and we use i indexed subsets like S_i to indicate a singleton (unit cardinality) element of $\mathbf{S}$ labeled by i.

Definition 2.2 (**Monotone Linkage Function** (Mullat, 1976)). A function $\pi(X_i, \mathbf{Z})$ defined on $\mathbf{Z} \in \mathcal{P}^{-X}, X_i \in \mathbf{X} \setminus \mathbf{Z}$ is called a monotone linkage function if

$$\pi(X_i, \mathbf{S}) \geq \pi(X_i, \mathbf{T}), \mathbf{S} \subseteq \mathbf{T} \in \mathcal{F}, \forall X_i \in \mathbf{X} \setminus T \quad (2)$$

We would like to note for the clarity of the reader that X_i is an element while $\mathbf{S}, \mathbf{T}$ are sets. Therefore, to make this distinction clear we denote sets in bold-faced font and elements otherwise.

Monotone linkage functions have been introduced and used for clustering in (Kempner et al., 1997; Kempner & Muchnik, 2003). A recent work (Seiffarth et al., 2021) uses these functions to find maximum margin separations in finite closure systems.

Induced quasi-concave set function optimization This is stated as the problem of maximizing a quasi-concave set function $M_\pi(\mathbf{T})$ over the modified power set $\mathcal{P}^{-X}$:

$$\arg\max_{\mathbf{T} \subset \mathcal{P}^{-\mathbf{X}}} M_\pi(\mathbf{T}) = \arg\max_{\mathbf{T} \subset \mathcal{P}^{-\mathbf{X}}} \min_{X_i \in \mathbf{X} \setminus \mathbf{T}} \pi(X_i, \mathbf{T}) \quad (3)$$

where $\pi(X_i, \mathbf{Z})$ is a monotone linkage function.

3. Contributions

1. We provide a parallel algorithm to find all the subsets that globally optimize the induced quasi-concave set function optimization problem in (3).

2. The proposed parallel algorithm has a time complexity over n processors of $\mathcal{O}(n^2 g) + \mathcal{O}(\log \log n)$ where n is the cardinality of the ground set and g is the complexity to compute the monotone linkage function that induces a corresponding quasi-concave set function via a duality. The complexity reduces to $\mathcal{O}(gn \log(n))$ on n^2 processors and to $\mathcal{O}(gn)$ on n^3 processors. The parallel approach reduces the currently existing cubic computational complexity of the non parallel version which is $\mathcal{O}(n^3 g) + \mathcal{O}(n)$.

3. As an example, we show that some functions of distance covariance (a measure of statistical dependence) are quasi-concave set functions. This lets us optimize them to obtain globally optimal maxi-min solutions for the most diverse subset of features.

3.1. Quasi-concave set function optimization under various set systems

A greedy-type algorithm for finding maximizers of induced quasi-concave set functions was constructed in (Mullat, 1976; Kuznecov et al., 1985; Zaks & Muchnik, 1989). Inspired by this work, extensions of these algorithms were developed for the setting of multipartite graphs in (Vashist, 2006). Similarly, quasi-concave set functions of distance covariance were derived in (Vepakomma & Kempner, 2019) and their optimization resulted in a solution for a diverse feature selection problem with guarantees. Furthermore, quasi-concave set functions were extended to various set systems including antimatroids (Levit & Kempner, 2004) and meet-semilattices in (Kempner & Muchnik, 2008).

4. Related work: Comparing quasi-concave set functions with submodularity

Given the seminal impact of submodular optimization, we would like to compare the definitions of quasi-concave set functions with submodular functions and their relaxations. We state some connections inline that we find accordingly.

1. **Submodular optimization** (Fujishige, 2005) Let $\mathbf{V}$ be a ground set with cardinality $|\mathbf{V}| = n$, and let $f : 2^{\mathbf{V}} \rightarrow \mathbb{R}_{\geq 0}$ be a set function defined on $\mathbf{V}$. The function f is said to be submodular if for any sets $\mathbf{X} \subseteq \mathbf{Y} \subseteq \mathbf{V}$ and any element $e \in V \backslash Y$, it holds that the discrete derivative

$$f(\mathbf{X} \cup \{e\}) - f(\mathbf{X}) \geq f(\mathbf{Y} \cup \{e\}) - f(\mathbf{Y})$$

is non-increasing in $\mathbf{X}$. That is, the incremental gain of adding an element to a subset is $\geq$ (is not smaller) the incremental gain of adding it to a superset. An equivalent definition is that for every **S**,**T** $\subseteq$ **V** we have that

$$f(\mathbf{S}) + f(\mathbf{T}) \geq f(\mathbf{S} \cup \mathbf{T}) + f(\mathbf{S} \cap \mathbf{T}) \quad (4)$$

The problem of maximizing a normalized monotone submodular function subject to a cardinality constraint has been studied extensively. A celebrated result of (Nemhauser et al., 1978) shows that a simple greedy algorithm that starts with an empty set and then iteratively adds elements with highest marginal gains provides a $(1 - 1/e)$-approximation.
Connection: Upon defining a linkage function to be

Type	Induced Quasi-concave set function (Parallel: **Ours**)	Induced Quasi-concave set function	Quasi-concave set function (General purpose)	Unconstrained Submodular	Robust submodular	Unconstrained Quasi submodular	Quasi semistrictly submodular M-/L-convex	$SSQM^{\neq}$ under M-convex domain
Complexity	On n processors, $\mathcal{O}(n^2 g) + \mathcal{O}(\log\log n)$. For n^2, n^3 processors, check Table 2.	$\mathcal{O}(n^3 g) + \mathcal{O}(n)$	Unknown	NP-Hard	$\mathcal{O}(nk)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2 \log L) + \mathcal{O}(n^2)$	$\mathcal{O}(n^4 (\log L)^2)$
Solution	Globally optimal	Globally optimal	Unknown	Unknown	Approximate	Approximate	Approximate	Approximate

Table 1. We show the computational complexity of our parallel algorithm and contrast it with that of its non-parallel version (cubic complexity), settings of submodular optimization and its relaxations. n is the size of the ground set, k is the cardinality of the returned set = $\max\{|x(v) - y(v)| | x, y \in dom\ f, v \in V\}$ where $f : Z^V \mapsto \mathbb{R} \cup \{+\infty\}$ and g is the complexity to compute the monotone linkage function.

# of processors	**Time Complexity**
n (Ours)	$\mathcal{O}(n^2 g)$
n^2 (Ours)	$\mathcal{O}(gn \log n)$
n^3 (Ours)	$\mathcal{O}(gn)$
Non-parallel	$\mathcal{O}(n^3 g) + \mathcal{O}(n)$

Table 2. In this table, we show the complexity of our proposed parallel algorithm with respect to increasing number of processors $n, n^2 \& n^3$. Here, n is also chosen to be around the order of size of the ground set. We show that the running times can be drastically reduced from the cubic complexities in the non-parallel version.

equal to a discrete derivative of a submodular function as

$$\pi(e, \mathbf{X}) = f(\mathbf{X} \cup \{e\}) - f(\mathbf{X})$$

it can be seen that the derivative of a submodular function is a monotone linkage function. However, not every monotone linkage function is a derivative of some submodular function (Muchnik & Shvartser, 1987a;b). Combining equations (3) and (4), we can say that the functions that are both submodular and quasi-concave set functions would satisfy $f(\mathbf{S}) + f(\mathbf{T}) >= f(\mathbf{S} \cup \mathbf{T}) + f(\mathbf{S} \cap \mathbf{T}) >= f(\mathbf{S} \cup \mathbf{T}) + \min\{f(\mathbf{S}), f(\mathbf{T})\}$.

2. **Robust submodular optimization** Robust versions of submodular optimization problem were introduced in (Krause et al., 2008; Mirzasoleiman et al., 2017; Bogunovic et al., 2017; Kazemi et al., 2018; Iyer, 2019; Avdiukhin et al., 2019; Powers et al., 2016). An earlier variant is of the form introduced in (Krause et al., 2008) as

$$\max_{\mathbf{S} \subseteq \mathbf{V}, |\mathbf{S}| \leq k} \min_{\mathbf{Z} \subseteq \mathbf{S}, |\mathbf{Z}| \leq \tau} f(\mathbf{S} \backslash \mathbf{Z})$$

The τ refers to a robustness parameter, representing the size of the subset $\mathbf{Z}$ that is removed from the selected set $\mathbf{S}$. The goal is to find a set $\mathbf{S}$ such that it is robust upon the worst possible removal of τ elements, i.e., after the removal, the objective value should remain as large as possible. For $\tau = 0$, the problem reduces to standard submodular optimization. The greedy algorithm, which is near-optimal for standard submodular optimization can perform arbitrarily badly for the robust version of the problem.
Connection: Note that our statement of induced quasi-concave set function optimization problem naturally has a robustness component that is similar to the max-min constraints used in the literature on robust submodular optimization.

3. **Quasi submodular and semi-strictly submodular functions** (Mei et al., 2015) A set function $F : 2^N \mapsto \mathbb{R}$ is quasi-submodular function if $\forall \mathbf{X}, \mathbf{Y} \subseteq \mathbf{N}$, *both* of the following conditions are satisfied

$$F(\mathbf{X} \cap \mathbf{Y}) \geq F(\mathbf{X}) \Rightarrow F(\mathbf{Y}) \geq F(\mathbf{X} \cup \mathbf{Y})$$
$$F(\mathbf{X} \cap \mathbf{Y}) > F(\mathbf{X}) \Rightarrow F(\mathbf{Y}) > F(\mathbf{X} \cup \mathbf{Y})$$

On a similar note, a rich family of semistrictly submodular, discrete Quasi L-convex and discrete M-convex functions were introduced in (Murota, 1998; 2009).

5. Algorithm and proof of optimality

We now introduce required definitions and corresponding theory to derive the algorithm. This includes definitions for π-series and π-clusters

Definition 5.1 (**π-series**)**.** We refer to a series $s_\pi = (X_{i_1}, \ldots, X_{i_N})$ as a π-series if

$$\pi(X_{i_{k+1}}, \overline{\mathbf{S}}_{\mathbf{k}}) = \min_{\mathbf{X_i} \in \mathbf{X} \backslash \overline{\mathbf{S}}_{\mathbf{k}}} \pi(\mathrm{X_i}, \overline{\mathbf{S}}_{\mathbf{k}}) \tag{5}$$

for any starting set $\overline{\mathbf{S}}_{\mathbf{k}} = \{\mathbf{X_{i_1}}, \ldots, \mathbf{X_{i_k}}\}, \mathbf{k} = \mathbf{1}, \ldots, \mathbf{N-1}$.

Therefore, it is a way of greedily populating a series that can start with any first element $\mathbf{X_{i_1}}$ being the current series, but the subsequent element to be added to the series, must be the element that minimizes the element to current series function of $\pi(\mathbf{X_{i_{k+1}}}, \overline{\mathbf{S}}_{\mathbf{k}})$ where $\mathbf{X_{i_{k+1}}}$ is the next element added and $\overline{\mathbf{S}}_{\mathbf{k}}$ is the current series.

Definition 5.2 (**π-cluster**)**.** A subset $\mathbf{S} \in \mathcal{P}^{-\mathbf{X}}$ will be referred to as a π-cluster if there exists a π-series, $s_\pi = (X_{i_1}, \ldots, X_{i_N})$, such that $\mathbf{S}$ is a maximizer of $M_\pi(\overline{\mathbf{S}}_{\mathbf{k}})$ over all starting sets $\overline{\mathbf{S}}_{\mathbf{k}}$ of s_π.

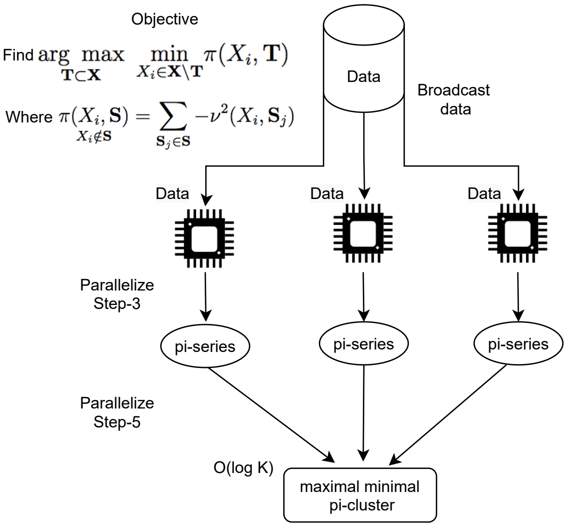
Objective
Find $\arg\max_{\mathbf{T}\subset\mathbf{X}} \min_{X_i\in\mathbf{X}\setminus\mathbf{T}} \pi(X_i,\mathbf{T})$
Where $\pi(X_i,\mathbf{S}) = \sum_{X_i\notin\mathbf{S}} \sum_{\mathbf{S}_j\in\mathbf{S}} -\nu^2(X_i,\mathbf{S}_j)$
Data
Broadcast data
Data
Data
Data
Parallelize Step-3
pi-series
pi-series
pi-series
Parallelize Step-5
O(log K)
maximal minimal pi-cluster

Figure 1. The proposed parallel algorithm consists of generating a π-series at each parallel entity over a copy of the data. The π-series at each entity starts with a different X_i. Each entity then generates a π-cluster corresponding to its generated $\pi - series$.The final step involves picking the best $\pi - cluster$. This is the only step that is not done in parallel.

Theorem 5.1. *(Kempner et al., 1997) If for a π-series $s_\pi = (X_{i_1}, X_{i_2}, \ldots, X_{i_N})$, a subset $\mathbf{S} \subset \mathbf{X}$ contains X_{i_1}, and if $X_{i_{k+1}}$ is the first element in s_π not contained in $\mathbf{S}$ (for some $k \in \{1, \ldots, N-1\}$, then $M_\pi(\overline{\mathbf{S}}_\mathbf{k}) \geq M_\pi(\mathbf{S})$*

where $\overline{\mathbf{S}}_\mathbf{k} = (X_{i_1}, \ldots, X_{i_k})$. In particular, if $\mathbf{S}$ is an inclusion-minimal maximizer of M_π (with regard to $\mathcal{P}^{-\mathbf{X}}$), then $\mathbf{S} = \overline{\mathbf{S}}_\mathbf{k}$, that is, $\mathbf{S}$ is a π-cluster.

From (Kempner et al., 1997) we have

Proposition 5.2. *If $\mathbf{S_1}, \mathbf{S_2} \subset \mathbf{X}$ are overlapping maximizers of a quasi-concave set function $M_\pi(\mathbf{S})$ over $\mathcal{P}^{-\mathbf{X}}$, then $\mathbf{S_1} \cap \mathbf{S_2}$ is also a maximizer of $M_\pi(\mathbf{S})$.*

This means that the minimal maximizers of a quasi-convex set function are not overlapping. Moreover, any nonminimal maximizer can be uniquely partitioned into a set of the minimal ones.

Theorem 5.3. *Each maximizer of a quasi-concave set function on $\mathcal{P}^{-\mathbf{X}}$ is a union of its inclusion-minimal maximizers.*

Proof. Indeed, if $\mathbf{S}^*$ is a maximizer of $M_\pi(\mathbf{S})$ over $\mathcal{P}^{-\mathbf{X}}$, then, according to Theorem 5.1, for any $X_i \in \mathbf{S}^*$, there exists a minimal maximizer included in $\mathbf{S}^*$ and containing X_i. □

Algorithm 1 Algorithm for induced quasi-convex set function optimization

```
1:  function =DIVERSEMINIMALMAXIMDCOV($\mathbf{X}$)
2:      for all $X_i \in \mathbf{X}$ do
3:      Greedily form $\pi$-series $s_\pi(x) = (X_i, X_{i_2} \ldots X_{i_N})$
        starting from $X_i$ as its first
            element.
4:          for each $\pi$-series $s_\pi(x)$ in step 3 do
5:          Find a corresponding smallest starting subset
            $\mathbf{T_x}$ with
              $M_\pi(\mathbf{T_x}) = \max_{\mathbf{1\leq k\leq N-1}} \pi(\mathbf{X_{i_{k+1}}}, \{\mathbf{X_{i_1}}, \ldots, \mathbf{X_{i_k}}\})$
6:          end for
7:      end for
8:      Among the non-coinciding minimal $\pi$-clusters $T_x$'s
        choose those that maximize
              $M_\pi(\mathbf{T_x}) = \min_{\mathbf{X_i}\in\mathbf{X}\setminus\mathbf{T_x}} \pi(\mathbf{X_i}, \mathbf{T_x})$
            all of which are the required minimal maximizers,
        and we return them as minimalMax
9:  return (minimalMax)
10: end function
```

Theorem 5.4. *The algorithm above finds all the minimal maximizers over $\mathcal{P}^{-\mathbf{X}}$.*

Proof. From Theorem 5.3 it follows that each element of minimalMax is a maximizer of $M_\pi(\mathbf{S})$ over $\mathcal{P}^{-\mathbf{X}}$. Assume that there is a minimal maximizer $\mathbf{S}$ that does not belong to minimalMax, and let $X_i \in \mathbf{S}$. Then, according to Theorem 5.1, there exist π-series starting from X_i and minimal π-cluster $T_x \subseteq \mathbf{S}$ containing X_i with $M_\pi(\mathbf{T_x}) \geq \mathbf{M}_\pi(\mathbf{S})$. Since $\mathbf{S}$ does not belong to minimalMax, and, according to Steps 5 and 8 of the algorithm, T_x or some subset of T_x belongs to minimalMax, there is a minimal maximizer strictly included in $\mathbf{S}$ which contradicts the minimality of $\mathbf{S}$. □

6. Computational complexity

When we have n processors, then we can build each π-series (in step-3 of algorithm) in $\mathcal{O}(n^2 g)$ on one processor (including step 5), and because we build them in parallel, steps 3-5 take $\mathcal{O}(n^2 g)$ time. Finding the maximum in step 8 takes $\mathcal{O}(\log\log n)$ time on n processors, under the CRCW (concurrent-read-concurrent-write) mode (Horowitz & Sahni, 1978; Horiguchi & Miranker, 1989; Valiant, 1975; Krizanc, 1999). If we have n^2 processors, n processors are used to build each π-series. To add one element to a series we have to find min between n elements, that takes $\mathcal{O}(\log\log n)$ on n processors, so to build each pi-series takes $g * (\log 1 + \log 2 + \ldots + \log n) = \mathcal{O}(gn \log n)$, and to finish it we have to find $\max$ with n^2 processors which takes $\mathcal{O}(1)$ time. This gives us $\mathcal{O}(gnloglogn)$ complexity. If we have n^3 processors, then we can use n^2 processors to build each π-series. To add one element to a series we have to find $\min$ between n elements which takes $\mathcal{O}(1)$ on n^2 processors. So to build each π-series takes $\mathcal{O}(gn)$ time, and to finish we have to find $\max$ with n^3 processors, that takes $\mathcal{O}(1)$ time. These are summarized in Tables 1 and 2.

7. Maxi-min Diverse Variable Selection

As an illustrating example, that we derive, we aim to find all the subsets that maximize the function $M_\pi(\mathbf{T})$ which result in the solutions which are diverse features in the context of statistics/machine learning as follows

$$\arg\max_{\mathbf{T}\subset\mathbf{X}} M_\pi(\mathbf{T}) = \arg\max_{\mathbf{T}\subset\mathbf{X}} \min_{X_i\in\mathbf{X}\setminus\mathbf{T}} \pi(X_i, \mathbf{T}) \tag{6}$$

For specificity, we use distance covariance upon normalization of the data as a measure of statistical dependence to model the diversity via $\pi(\mathbf{X_i}, \mathbf{S})$ as defined in Lemma 8.1.

8. Relevant Background on Distance Covariance and Distance Correlation

In this section we introduce some preliminaries about distance correlation and distance covariance and illustrate a connection between these functions and quasi-concave set function optimization. Distance Correlation (Székely et al., 2007) is a measure of nonlinear statistical dependencies between random vectors of arbitrary dimensions. We describe below distance covariance $\nu^2(\mathbf{x}, \mathbf{y})$ between random variables $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^m$ with finite first moments is a non-negative number as

$$\nu^2(\mathbf{x}, \mathbf{y}) = \int_{\mathbb{R}^{d+m}} |f_{\mathbf{x},\mathbf{y}}(t,s) - f_{\mathbf{x}}(t) f_{\mathbf{y}}(s)|^2 w(t,s) dt ds \tag{7}$$

where $w(t,s)$ is a weight function as defined in (Székely et al., 2007), $f_{\mathbf{x}}, f_{\mathbf{y}}$ are characteristic functions of $\mathbf{x}, \mathbf{y}$ and $f_{\mathbf{x},\mathbf{y}}$ is the joint characteristic function.

The distance covariance is zero if and only if random variables $\mathbf{x}$ and $\mathbf{y}$ are independent. Using the above definition of distance covariance, we have the following expression for Distance Correlation (Székely et al., 2007):

The squared Distance Correlation between random variables $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^m$ with finite first moments is a nonnegative number is defined as

$$\rho^2(\mathbf{x}, \mathbf{y}) = \begin{cases} \frac{\nu^2(\mathbf{x},\mathbf{y})}{\sqrt{\nu^2(\mathbf{x},\mathbf{x})\nu^2(\mathbf{y},\mathbf{y})}}, & \nu^2(\mathbf{x},\mathbf{x})\nu^2(\mathbf{y},\mathbf{y}) > 0. \\ 0, & \nu^2(\mathbf{x},\mathbf{x})\nu^2(\mathbf{y},\mathbf{y}) = 0. \end{cases} \tag{8}$$

The Distance Correlation defined above has the following interesting properties.

1. $\rho^2(\mathbf{x}, \mathbf{y})$ is applicable for arbitrary dimensions d and m of $\mathbf{x}$ and $\mathbf{y}$ respectively.
2. $\rho^2(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x}$ and $\mathbf{y}$ are independent.
3. $\rho^2(\mathbf{x}, \mathbf{y})$ satisfies the relation $0 \le \rho^2(\mathbf{x}, \mathbf{y}) \le 1$.

8.1. Sample Distance Covariance and Sample Distance Correlation

We provide the definition of sample version of distance covariance given samples $\{(\mathbf{x}_k, \mathbf{y}_k) | k = 1, 2, \ldots, n\}$ sampled i.i.d. from joint distribution of random vectors $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{y} \in \mathbb{R}^m$. To do so, we define two squared Euclidean distance matrices $\mathbf{E_X}$ and $\mathbf{E_Y}$, where each entry $[\mathbf{E_X}]_{k,l} = \|\mathbf{x}_k - \mathbf{x}_l\|^2$ and $[\mathbf{E_Y}]_{k,l} = \|\mathbf{y}_k - \mathbf{y}_l\|^2$ with $k, l \in \{1, 2, \ldots, n\}$. These squared distance matrices are double-centered by making their row and column sums zero and are denoted as $\widehat{\mathbf{E}}_\mathbf{X}$, $\widehat{\mathbf{Q}}_\mathbf{X}$, respectively. So given a double-centering matrix $\mathbf{J} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, we have $\widehat{\mathbf{E}}_\mathbf{X} = \mathbf{J}\mathbf{E_X}\mathbf{J}$ and $\widehat{\mathbf{E}}_\mathbf{Y} = \mathbf{J}\mathbf{E_Y}\mathbf{J}$. The sample distance covariance and sample distance correlation can now be defined as follows.

Definition 8.1. Sample Distance Covariance (Székely et al., 2007): Given i.i.d samples $\mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_k, \mathbf{y}_k) | k = 1, 2, 3, \ldots, n\}$ and corresponding double centered Euclidean distance matrices $\widehat{\mathbf{E}}_\mathbf{X}$ and $\widehat{\mathbf{E}}_\mathbf{Y}$, the squared sample distance correlation is defined as,

$$\hat{\nu}^2(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2} \sum_{k,l=1}^{n} [\widehat{\mathbf{E}}_\mathbf{X}]_{k,l} [\widehat{\mathbf{E}}_\mathbf{Y}]_{k,l},$$

Using this, sample distance correlation is given by

$$\hat{\rho}^2(\mathbf{X}, \mathbf{Y}) = \begin{cases} \frac{\hat{\nu}^2(\mathbf{X},\mathbf{Y})}{\sqrt{\hat{\nu}^2(\mathbf{X},\mathbf{X})\hat{\nu}^2(\mathbf{Y},\mathbf{Y})}}, & \hat{\nu}^2(\mathbf{X},\mathbf{X})\hat{\nu}^2(\mathbf{Y},\mathbf{Y}) > 0. \\ 0, & \hat{\nu}^2(\mathbf{X},\mathbf{X})\hat{\nu}^2(\mathbf{Y},\mathbf{Y}) = 0. \end{cases}$$

Monotonicity of distance covariance under lack of independence: If $\mathbf{X}, \mathbf{Z} \in \mathbb{R}^p$ and $\mathbf{Y} \in \mathbb{R}^q$ and if $\mathbf{Z} \perp\!\!\!\perp (\mathbf{X}, \mathbf{Y})$ then

$$\nu^2(\mathbf{X} + \mathbf{Z}, \mathbf{Y}) \le \nu^2(\mathbf{X}, \mathbf{Y}) \tag{9}$$

Note that $\perp\!\!\!\perp$ indicates 'statistically independent' in statistical literature.

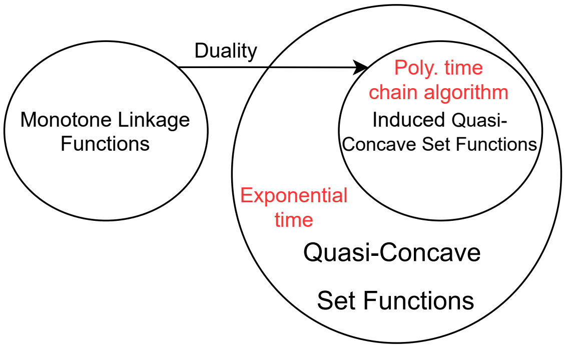
Duality
Monotone Linkage Functions
Poly. time chain algorithm
Induced Quasi-Concave Set Functions
Exponential time
Quasi-Concave Set Functions

Figure 2. This illustration refers to the duality between monotone linkage functions and quasi-concave set functions. Optimization algorithms for general ququasi-concave set functions do not exist while those that are induced via monotone linkage functions can be optimized in polynomial time.

8.2. Motivating applications for modeling diversity with quasi-concave set function optimization

A minor sampling of applications that benefit from the results in this paper do parallel traditional applications seen in submodular optimization literature. A few directions are listed below.

1. Maximally/minimally correlated marginal selection for private data synthesis (Zhang et al., 2021).
2. Modeling diversity in active learning (Wei et al., 2015), determinantal point processes (Tschiatschek et al., 2016).
3. Diverse sample selection, feature selection and data summarization in machine learning and statistics. (Prasad et al., 2014; Das et al., 2012)

8.3. A monotone linkage function of distance covariance

Lemma 8.1. *The function $\pi(X_i, \mathbf{S})$ of distance covariance defined on $X_i \notin \mathbf{S}$ as*

$$\pi(X_i, \mathbf{S})_{X_i \notin \mathbf{S}} = \sum_{\mathbf{S}_j \in \mathbf{S}} -\nu^2(X_i, \mathbf{S}_j) \tag{10}$$

is a monotone linkage function.

Proof: For $\mathbf{S} \subseteq \mathbf{T}$ we have

$$\pi(X_i, \mathbf{T})_{X_i \notin \mathbf{T}} = \sum_{\mathbf{S}_j \in \mathbf{S}} -\nu_i^2(X_i, \mathbf{S}_j) - \sum_{\mathbf{T}_j \in \mathbf{T} \setminus \mathbf{S}} \nu_i^2(X_i, \mathbf{T}_j) \tag{11}$$

$$\leq \pi(X_i, \mathbf{S})_{X_i \notin \mathbf{T}} = \sum_{\mathbf{S}_j \in \mathbf{S}} -\nu_i^2(X_i, \mathbf{S}_j) \tag{12}$$

We would also like to note that as $\nu(\cdot)$ is a non-negative function the above inequality does hold true.

By Assertion 1 from (Kempner et al., 1997), we conclude that the function $M_\pi(\mathbf{T}) = \min_{X_i \in \mathbf{X} \setminus \mathbf{T}} \pi(X_i, \mathbf{T})$ is a quasi-concave set function.

Theorem 8.2 (Quasi-Concave Distance Covariance Set Function Theorem)**.** *If we have* $\mathbf{S} \cap \mathbf{T} \neq \varnothing$ *and* $\forall \mathbf{S}, \mathbf{T}, \mathbf{Y}$ *if* $\nu^2(\mathbf{S}, \mathbf{T}) > 0 \wedge \nu^2(\mathbf{S}, \mathbf{Y}) > 0 \wedge \nu^2(\mathbf{T}, \mathbf{Y}) > 0$ *then, we have*

$$-\nu^2(\mathbf{S} \cap \mathbf{T}, \mathbf{Y}) \geq min(-\nu^2(\mathbf{S}, \mathbf{Y}), -\nu^2(\mathbf{T}, \mathbf{Y})) \tag{13}$$

Proof. (Vepakomma & Kempner, 2019)

If $\mathbf{S} \cap \mathbf{T} = \mathbf{S}$ then since $\mathbf{S} \subseteq \mathbf{T}$

the Kosorok's distance covariance inequality simplifies to give

$$-\nu^2(\mathbf{S}, \mathbf{Y}) \geq -\nu^2(\mathbf{T}, \mathbf{Y}) \tag{14}$$

Therefore, we have

$$-\nu^2(\mathbf{S} \cap \mathbf{T}, \mathbf{Y}) \geq min(-\nu^2(\mathbf{S}, \mathbf{Y}), -\nu^2(\mathbf{T}, \mathbf{Y}))$$

Similarly, if $\mathbf{S} \cap \mathbf{T} = \mathbf{T}$, then since $\mathbf{T} \subseteq \mathbf{S}$

$$-\nu^2(\mathbf{T}, \mathbf{Y}) \geq -\nu^2(\mathbf{S}, \mathbf{Y}) \tag{15}$$

and therefore,

$$-\nu^2(\mathbf{S} \cap \mathbf{T}, \mathbf{Y}) \geq min(-\nu^2(\mathbf{S}, \mathbf{Y}), -\nu^2(\mathbf{T}, \mathbf{Y})) \tag{16}$$

In the cases of $\mathbf{S} \cap \mathbf{T} \subset \mathbf{S}$ and $\mathbf{S} \cap \mathbf{T} \subset \mathbf{T}$ the Kosorok's distance covariance inequality gives

$$-\nu^2(\mathbf{S} \cap \mathbf{T}, \mathbf{Y}) > -\nu^2(\mathbf{S}, \mathbf{Y}) \tag{17}$$

and

$$-\nu^2(\mathbf{S} \cap \mathbf{T}, \mathbf{Y}) > -\nu^2(\mathbf{T}, \mathbf{Y}) \tag{18}$$

Thus,

$$-\nu^2(\mathbf{S} \cap \mathbf{T}, \mathbf{Y}) \geq min(-\nu^2(\mathbf{S}, \mathbf{Y}), -\nu^2(\mathbf{T}, \mathbf{Y})) \tag{19}$$

□

9. Conclusion

We showed that Algorithm 1 gives globally exact solutions that to the induced quasi-concave set function optimization and is highly parallelizable. This opens doors to a wide variety of real world applications that we would like to pursue as part of future work.

References

Algaba, E., Bilbao, J. M., Van den Brink, R., and Jiménez-Losada, A. Cooperative games on antimatroids. *Discrete Mathematics*, 282(1-3):1–15, 2004.

Avdiukhin, D., Mitrović, S., Yaroslavtsev, G., and Zhou, S. Adversarially robust submodular maximization under knapsack constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 148–156, 2019.

Bian, A. A., Buhmann, J. M., Krause, A., and Tschiatschek, S. Guarantees for greedy maximization of non-submodular functions with applications. In *International conference on machine learning*, pp. 498–507. PMLR, 2017.

Bogunovic, I., Mitrović, S., Scarlett, J., and Cevher, V. Robust submodular maximization: A non-uniform partitioning approach. In *International Conference on Machine Learning*, pp. 508–516. PMLR, 2017.

Bogunovic, I., Zhao, J., and Cevher, V. Robust maximization of non-submodular objectives. In *International Conference on Artificial Intelligence and Statistics*, pp. 890–899. PMLR, 2018.

Chajda, I., Halaš, R., and Kühr, J. *Semilattice structures*, volume 30. Heldermann Lemgo, 2007.

Chierichetti, F., Dasgupta, A., and Kumar, R. On additive approximate submodularity. *arXiv e-prints*, pp. arXiv–2010, 2020.

Conforti, M. and Laurent, M. On the geometric structure of independence systems. *Mathematical programming*, 45 (1):255–277, 1989.

Das, A. and Kempe, D. Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. *The Journal of Machine Learning Research*, 19(1):74–107, 2018.

Das, A., Dasgupta, A., and Kumar, R. Selecting diverse features via spectral regularization. *Advances in neural information processing systems*, 25:1583–1591, 2012.

Dietrich, B. L. Matroids and antimatroids—a survey. *Discrete Mathematics*, 78(3):223–237, 1989.

Edmonds, J. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization—Eureka, You Shrink!*, pp. 11–26. Springer, 2003.

Feige, U., Mirrokni, V. S., and Vondrák, J. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

Fujishige, S. *Submodular functions and optimization*. Elsevier, 2005.

Horel, T. and Singer, Y. Maximization of approximately submodular functions. In *NIPS*, volume 16, pp. 3045–3053, 2016.

Horiguchi, S. and Miranker, W. L. A parallel algorithm for finding the maximum value. *Parallel computing*, 10(1): 101–108, 1989.

Horowitz, E. and Sahni, S. Fundamentals of computer algorithms. 1978.

Iyer, R. A unified framework of robust submodular optimization. *arXiv preprint arXiv:1906.06393*, 2019.

Iyer, R. and Bilmes, J. Submodular optimization with submodular cover and submodular knapsack constraints. *arXiv preprint arXiv:1311.2106*, 2013.

Kazemi, E., Zadimoghaddam, M., and Karbasi, A. Scalable deletion-robust submodular maximization: Data summarization with privacy and fairness constraints. In *International conference on machine learning*, pp. 2544–2553. PMLR, 2018.

Kempner, Y. and Levit, V. E. Correspondence between two antimatroid algorithmic characterizations. *The Electronic Journal of Combinatorics, 10, 2003*, 2003.

Kempner, Y. and Muchnik, I. Clustering on antimatroids and convex geometries. *WSEAS Transactions on Mathematics*, 2(1):54–59, 2003.

Kempner, Y. and Muchnik, I. Quasi-concave functions on meet-semilattices. *Discrete applied mathematics*, 156(4): 492–499, 2008.

Kempner, Y., Mirkin, B., and Muchnik, I. Monotone linkage clustering and quasi-concave set functions. *Applied Mathematics Letters*, 10(4):19–24, 1997.

Korte, B., Lovász, L., and Schrader, R. *Greedoids*, volume 4. Springer Science & Business Media, 2012.

Krause, A. and Golovin, D. Submodular function maximization. *Tractability*, 3:71–104, 2014.

Krause, A., McMahan, H. B., Guestrin, C., and Gupta, A. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(12), 2008.

Krizanc, D. A survey of randomness and parallism in comparison problems. In *Advances in Randomized Parallel Computing*, pp. 25–39. Springer, 1999.

Kuznecov, E., Muchnik, I., and Shvartzer, L. Monotonic systems and their properties. 1985.

Levit, V. E. and Kempner, Y. Quasi-concave functions on antimatroids. *arXiv preprint math/0408365*, 2004.

Lovász, L. Submodular functions and convexity. In *Mathematical programming the state of the art*, pp. 235–257. Springer, 1983.

Mei, J., Zhao, K., and Lu, B.-L. On unconstrained quasi-submodular function optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Mirzasoleiman, B., Karbasi, A., and Krause, A. Deletion-robust submodular maximization: Data summarization with "the right to be forgotten". In *International Conference on Machine Learning*, pp. 2449–2458. PMLR, 2017.

Muchnik, I. and Shvartser, L. Submodular set functions and monotone systems in aggregation, i. *Automation and Remote Control 1987*, (5), 1987a.

Muchnik, I. and Shvartser, L. Submodular set functions and monotone systems in aggregation, ii. *Automation and Remote Control 1987*, (5), 1987b.

Mullat, I. Extremal subsystems of monotonic systems. 1. *Automation and Remote Control*, 37(5):758–766, 1976.

Murota, K. Discrete convex analysis. *Mathematical Programming*, 83(1):313–371, 1998.

Murota, K. Recent developments in discrete convex analysis. In *Research trends in combinatorial optimization*, pp. 219–260. Springer, 2009.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Powers, T., Bilmes, J., Wisdom, S., Krout, D. W., and Atlas, L. Constrained robust submodular optimization. In *NIPS OPT2016 workshop*, 2016.

Prasad, A., Jegelka, S., and Batra, D. Submodular meets structured: Finding diverse subsets in exponentially-large structured item sets. *arXiv preprint arXiv:1411.1752*, 2014.

Seiffarth, F., Horváth, T., and Wrobel, S. Maximum margin separations in finite closure systems. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part I*, pp. 3–18. Springer International Publishing, 2021.

Székely, G. J., Rizzo, M. L., Bakirov, N. K., et al. Measuring and testing dependence by correlation of distances. *The annals of statistics*, 35(6):2769–2794, 2007.

Tschiatschek, S., Djolonga, J., and Krause, A. Learning probabilistic submodular diversity models via noise contrastive estimation. In *Artificial Intelligence and Statistics*, pp. 770–779. PMLR, 2016.

Valiant, L. G. Parallelism in comparison problems. *SIAM Journal on Computing*, 4(3):348–355, 1975.

Vashist, A. K. *PhD Thesis: Multipartite graph clustering for structured datasets and automating ortholog extraction*, volume 68. 2006.

Vepakomma, P. and Kempner, Y. Diverse data selection via combinatorial quasi-concavity of distance covariance: A polynomial time global minimax algorithm. *Discrete Applied Mathematics*, 265:182–191, 2019.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pp. 1954–1963. PMLR, 2015.

Zaks, Y. M. and Muchnik, I. Incomplete classifications of a finite set of objects using monotone systems. *Automation and Remote Control*, 50:553–560, 1989.

Zhang, Z., Wang, T., Li, N., Honorio, J., Backes, M., He, S., Chen, J., and Zhang, Y. Privsyn: Differentially private data synthesis. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.