On Bisubmodular Maximization

Ajit P. Singh
University of Washington, Seattle
ajit@ee.washington.edu

Andrew Guillory
University of Washington, Seattle
guillory@cs.washington.edu

Jeff Bilmes
University of Washington, Seattle
bilmes@ee.washington.edu

Abstract

Bisubmodularity extends the concept of submodularity to set functions with two arguments. We show how bisubmodular maximization leads to richer value-of-information problems, using examples in sensor placement and feature selection. We present the first constant-factor approximation algorithm for a wide class of bisubmodular maximizations.

1 Introduction

Central to decision making is the problem of selecting informative observations, subject to constraints on the cost of data acquisition. For example:

Sensor Placement: One is given a set of n locations, V, and a set of $k \ll n$ sensors, each of which can cover a fixed area around it. Which subset $X \subset V$ should be instrumented, to maximize the total area covered?

Feature Selection: Given a regression model on n features, V, and an objective which measures feature quality, $f : 2^V \to \mathbb{R}$, select the best k features.

Both examples are selection problems, for which a near-optimal approximation can be found efficiently when the objective is *submodular* (see Definition 1, below). Many selection problems involve a submodular objective: e.g., sensor placement [16, 15], document summarization [20], influence maximization in social networks [12], and feature selection [13]. All can be framed as submodular function maximization:

$$\max_{S \subseteq V} \quad f(S) \quad \text{subject to} \quad S \in \mathcal{I}_i,\ \forall i \in \{1 \dots r\}. \tag{1}$$

The constraint sets $\mathcal{I}_i$ are independent sets of matroids, which include knapsack or cardinality constraints. Fully polynomial-time approximation algorithms exist when the matroid constraints consist only of a cardinality constraint [22], or under multiple matroid constraints [6].

Our interest lies in enriching the scope of selection problems which can be efficiently solved. We discuss a richer class of properties on the objective, known as *bisubmodularity* [24, 1] to describe biset optimizations:

$$\max_{A,B \subseteq V} \quad f(A, B) \quad \text{subject to} \quad (A, B) \in \mathcal{I}_i,\ \forall i \in \{1 \dots r\}. \tag{2}$$

Bisubmodular function maximization allows for richer problem structures than submodular max: i.e., simultaneously selecting and partitioning elements into two groups, where the value of adding an element to a set can depend on interactions between the two sets.

To illustrate the potential of bisubmodular maximization, we consider two distinct problems:

Coupled Sensor Placement (Section 5): One is given a set of n locations, V. There are two different kinds of sensors, which differ in cost and area covered. Given a total sensor budget, k, select sites to instrument with sensors of each type (A, B), to maximize the total area covered.

Coupled Feature Selection (Section 6): Given a regression model on n features, V, select a set of k features and partition them into disjoint sets A and B, such that a joint measure of feature quality $f : A \times B \to \mathbb{R}$ is maximized.

This paper is the first to (i) propose bisubmodular maximization to describe a richer class of value-of-information problems; (ii) define sufficient conditions under which bisubmodular max is efficiently solvable; (iii) provide an algorithm for such instances (algorithms for bisubmodular min exist [7]); (iv) prove, by construction, the existence of an embedding of a directed bisubmodular function into a submodular one. Ancillary to our study of bisubmodular maximization is a new result on the extension of submodular functions defined over matroids (Theorem 2).

Our key observation is that many bisubmodular

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22.

maximization problems can be reduced to matroid-constrained submodular maximization, for which efficient, constant-factor approximation algorithms already exist. We first introduce the concept of simple bisubmodularity (Section 3), which describes the biset analogue of submodularity. Directed bisubmodularity [23] allows for more complex interaction between the set arguments. The question of whether or not directed bisubmodular functions can be efficiently maximized was has been an open problem for over twenty years. Our reduction approach proves that a wide range of directed bisubmodular objectives can be efficiently maximized (Section 4).

2 Preliminaries

We first introduce basic concepts and notation.

Definition 1 (Submodularity)**.** *A set-valued function over ground set V, $f : 2^V \to \mathbb{R}$ is submodular if for any $A \subseteq A' \subseteq V \setminus v$,*

$$f(A + v) - f(A) \geq f(A' + v) - f(A'). \quad (3)$$

Equivalently, $\forall A, A' \subseteq V$,

$$f(A) + f(A') \geq f(A \cup A') + f(A \cap A'). \quad (4)$$

$+$ is used to denote adding an element to a set.

A set function $f(A, B)$ with two arguments $A \subseteq V$ and $B \subseteq V$ is a biset function. In some cases we assume the domain of f to be all ordered pairs of subsets of V:

$$2^{2V} \triangleq \{(A, B) : A \subseteq V,\ B \subseteq V\}.$$

In other cases we assume the domain of f to be ordered pairs of *disjoint* subsets of V:

$$3^V \triangleq \{(A, B) : A \subseteq V,\ B \subseteq V,\ A \cap B = \emptyset\}.$$

A biset function is normalized if $f(\emptyset, \emptyset) = 0$. In some cases we will also assume $f(A, B)$ is monotone.

Definition 2 (Monotonicity)**.** *A biset function f over 2^{2V} is monotone (monotone non-decreasing) if for any $s \in V$, $(A, B) \in 2^{2V}$:*

$$f(A + s, B) \geq f(A, B) \text{ and } f(A, B + s) \geq f(A, B).$$

A biset function f over 3^V is monotone (monotone non-decreasing) if for any $s \in V \setminus (A \cup B)$, $(A, B) \in 3^V$:

$$f(A + s, B) \geq f(A, B) \text{ and } f(A, B + s) \geq f(A, B).$$

Monotone submodular maximization is nontrivial only when constrained. We focus on matroid constraints:

Definition 3 (Matroid)**.** *A matroid $\mathcal{I}$ is a set of sets with three properties: (i) $\emptyset \in \mathcal{I}$, (ii) if $A \in \mathcal{I}$ and $B \subseteq A$ then $B \in \mathcal{I}$ (i.e., $\mathcal{I}$ is an independence system), and (iii) if $A \in \mathcal{I}$ and $B \in \mathcal{I}$ and $|B| > |A|$ then there is an $s \in B$ such that $s \notin A$ and $(A + s) \in \mathcal{I}$.*

3 Simple Bisubmodularity

A natural analogue to submodularity for biset functions is simple bisubmodularity:

Definition 4 (Simple Bisubmodularity)**.** *$f : 2^{2V} \to \mathbb{R}$ is simple bisubmodular iff for each $(A, B) \in 2^{2V}$, $(A', B') \in 2^{2V}$ with $A \subseteq A'$, $B \subseteq B'$ we have for $s \notin A'$ and $s \notin B'$:*

$$f(A + s, B) - f(A, B) \geq f(A' + s, B') - f(A', B'),$$
$$f(A, B + s) - f(A, B) \geq f(A', B' + s) - f(A', B').$$

Equivalently, $\forall (A, B), (A', B') \in 2^{2V}$,

$$f(A, B) + f(A', B') \geq f(A \cup A', B \cup B') + f(A \cap A', B \cap B')$$

Fixing one of the coordinates of $f(A, B)$ yields a submodular function in the free coordinate.

3.1 Reduction to Submodular Maximization

The reduction uses a duplication of the ground set (c.f., [10, 2]). Define $\bar{V}$ to be an extended ground set formed by taking the disjoint union of 2 copies of the original ground set V. For an element $s \in V$, we use s_i where $i \in \{1, 2\}$ to refer to the ith copy in $\bar{V}$. Then $\bar{V} = V_1 \cup V_2$ where $V_i \triangleq \bigcup_i s_i$. For an element $s \in \bar{V}$ we use $\text{abs}(s)$ to refer to the corresponding original element in V (i.e. $\text{abs}(s_i) \triangleq s$). For a set $S \subseteq \bar{V}$ we similarly use $\text{abs}(S) \triangleq \{\text{abs}(s) : s \in S\}$.

Given any simple bisubmodular function f, define a single-argument set function g for $S \subseteq \bar{V}$:

$$g(S) \triangleq f(\text{abs}(S \cap V_1), \text{abs}(S \cap V_2)).$$

Note that there is a one-to-one mapping between $f(A, B)$ for $(A, B) \in 2^{2V}$ and $g(S)$ for $S \subseteq \bar{V}$ and therefore maximizing $g(S)$ is equivalent to maximizing $f(A, B)$. Furthermore,

Lemma 1. *$g(S)$ is submodular if $f(A, B)$ is a simple bisubmodular function.*

Note also that g is monotone when f is monotone. Based on this connection, maximizing a simple bisubmodular function f reduces to maximizing a normal submodular function g. We can then exploit constant-factor approximation algorithms for submodular function maximization.

Corollary 1. *If $f(A, B)$ is non-negative and simple bisubmodular, then there exists a constant-factor approximation algorithm for solving Equation 2 when the constraints are either: (i) non-existent; (ii) $|A| \leq k_1$, $|B| \leq k_2$; (iii) $|A| + |B| \leq k$; (iv) $A \cap B = \emptyset$ (v) any combination of the above.*

Proof. With no constraints, maximizing $f(A, B)$ corresponds to maximizing non-negative, submodular $g(S)$, solvable using Feige et al. [5]. The only question is how to represent the other constraints on f as matroid constraints on g. Cases (ii) and (iii) reduce to uniform matroids. In case (ii), $|S \cap V_1| \leq k_1$, $|S \cap V_2| \leq k_2$; in case (iii) $|S \cap V_1| + |S \cap V_2| \leq k$. For case (iv) the constraint is a partition matroid constraint $\forall v \in V$, $|S \cap \{v_1, v_2\}| \leq 1$. For any intersection of a constant number of matroid constraints we can use the algorithms of Lee et al. [18]. For the special case of monotone f (and therefore g) we can use the simple greedy algorithm of Fisher et al. [6] for a constant-factor approximation. For the special case of a single matroid constraint and monotone f and g, the algorithm of Calinescu et al. [2] gives an optimal approximation ratio of $1 - 1/e$. □

3.2 Coordinate-wise Maximization

Simple bisubmodular functions can also be maximized using a coordinate-wise procedure. Consider

$$\begin{aligned} \max_{A,B} \quad & f(A, B) \\ \text{subject to} \quad & (A, B) \in 2^{2V}, |A| \leq k_1, |B| \leq k_2. \end{aligned}$$

If f is simple then it suffices to solve the following pair of submodular optimizations:

$$\begin{aligned} A^* &= \operatorname*{argmax}_{A \subseteq V : |A| \leq k_1} f(A, \emptyset), \\ B^* &= \operatorname*{argmax}_{B \subseteq V : |B| \leq k_2} f(A^*, B), \end{aligned}$$

which due to Corollary 1, corresponds to the local greedy algorithm, which is approximately optimal [6].

Budget constraints of the form $|A| + |B| \leq k$ may be handled by converting the constraint into $|A| \leq k_1, |B| \leq k_2$, and then taking the best solution across all possible (k_1, k_2) division of the budget $k_1 + k_2 = k$. One possible approach to this would require $O(k)$ submodular optimizations. Budget constraints of the form $c_1|A| + c_2|B| \leq k$ with integer costs c_1, c_2 are handled in a similar fashion. The search-over-budget-divisions approach is still approximately optimal, since the domain under one of the budget divisions contains the optimal solution. However, searching over budget divisions requires $O(k^2)$ runs of the greedy algorithm.

4 Directed Bisubmodularity

Simple bisubmodular functions are related to a different class of functions called directed bisubmodular functions. Qi [23] posed the question of whether directed bisubmodular functions can be efficiently maximized. The previous section answers this question for simple bisubmodular functions.

In many situations, including those described in Sections 5 and 6, simple bisubmodularity is sufficient. The contributions of this section are primarily theoretical: we (i) connect simple to directed bisubmodularity, (ii) give a method for embedding a directed bisubmodular function into a submodular one; (iii) provide sufficient conditions under which directed bisubmodular maximization can be reduced to submodular maximization.

Definition 5 (Directed Bisubmodularity [24])**.** *Biset function $f : 3^V \to \mathbb{R}$ is directed bisubmodular iff*

$$\begin{aligned} f(A, B) + f(A', B') \geq f(A \cap A', B \cap B') + \\ f((A \cup A') \setminus (B \cup B'), (B \cup B') \setminus (A \cup A')). \end{aligned}$$

The set subtraction can be seen as enforcing the constraint that the two arguments of f remain disjoint.

To connect simple to directed bisubmodularity, we use a characterization of such functions by Ando et al. [1].

Definition 6. *$f : 3^V \to \mathcal{R}$ is said to be* ***submodular in every orthant*** *if for every partition of V given by A, B with $A \cup B = V$ and $A \cap B = \emptyset$ the function*

$$\hat{f}(S) \triangleq f(A \cap S, B \cap S)$$

is a submodular function.

Theorem 1 (Ando Conditions [1])**.** *A function f is directed bisubmodular iff*

1. *f is submodular in every orthant.*
2. *For any $(A, B) \in 3^V$, $s \in V \setminus (A \cup B)$ we have $f(A + s, B) + f(A, B + s) \geq 2f(A, B)$.*

The second condition is satisfied if f is monotone. It is not hard to show the following.

Proposition 1. *If f is simple bisubmodular then f restricted to 3^V is submodular in every orthant.*

Therefore we have this relationship between simple and directed bisubmodular functions.

Corollary 2. *If f is simple bisubmodular and monotone then f restricted to 3^V is directed bisubmodular.*

4.1 Embedding into a Submodular Function

At first glance, the techniques of Section 3.1 might appear to yield an approximation algorithm for directed bisubmodular maximization, but unfortunately they do not: f is only defined over 3^V while Section 3.1 made use of the fact that simple bisubmodular functions are defined over the larger set 2^{2V}. We prove, by construction, that any bisubmodular function can be embedded into a submodular one. However, for directed bisubmodular functions the resulting submodular function is not guaranteed to be monotone and non-negative, which precludes direct use of a submodular maximization oracle for directed bisubmodular maximization.

As before, define $\bar{V}$ to be an extended ground set formed by taking the disjoint union of 2 copies of the original ground set V. Let $g : S \subseteq \bar{V} \to \mathbb{R}$, where

$$g(S) \triangleq f(\text{abs}(S \cap V_1), \text{abs}(S \cap V_2)).$$

Since f is directed bisubmodular, it is only defined over disjoint pairs of subsets, and therefore g is not defined for each $S \subseteq \bar{V}$. For example, if both $s_1 \in V$ and $s_2 \in V$ for some $s \in V$, then $\text{abs}(S \cap V_1)$ and $\text{abs}(S \cap V_2)$ are not disjoint. When f is directed bisubmodular, there is no guarantee of a one-to-one mapping between the domains of f and g.

While g is not submodular over set system $2^{\bar{V}}$, it is submodular over the values in $2^{\bar{V}}$ where it is defined, which are the independent sets of a matroid.

Definition 7. *$g(S)$ is submodular over independence system $\mathcal{I}$ if for every $A \subseteq A' \subset (A' \cup s) \in \mathcal{I}$ we have*

$$g(A \cup s) - g(A) \geq g(A' \cup s) - g(A').$$

Equivalently, g is submodular over $\mathcal{I}$ if for any A, A' with $(A \cup A') \in \mathcal{I}$ we have

$$g(A) + g(A') \geq g(A \cup A') + g(A \cap A').$$

A directed bisubmodular f is defined over 3^V, so the values where g is defined form a partition matroid. A directed bisubmodular function is submodular in each orthant, so g is submodular over its partition matroid:

Corollary 3. *If $f(A, B)$ is directed bisubmodular then $g(S) \triangleq f(\text{abs}(S \cap V_1), \text{abs}(S \cap V_2))$ is both defined and submodular over the partition matroid*

$$\mathcal{I} = \{S : \forall_{s \in V} |\{s_1, s_2\} \cap S| \leq 1\}.$$

Seemingly, maximizing $f(A, B)$ is equivalent to maximizing g over a partition matroid, which ensures that the solution found is in 3^V. Fisher et al. [6] provides an algorithm for maximizing g subject to a matroid constraint, which requires only evaluating g for values in the matroid. The subtle flaw in this argument is that the proof of the approximation guarantee in Fisher et al. [6] involves evaluating g outside of the constraints.

4.1.1 Extending Submodular Functions

A function g defined on a subset of $\mathcal{S} \subset 2^{\bar{V}}$ is known as a *partial function.* The embedding of a non-negative directed bisubmodular function leads to a non-negative partial function g which is submodular over the independent sets of a matroid $I \in \mathcal{I}$. Is it possible to define an *extension* of g, g': a submodular function defined on $2^{\bar{V}}$, where $\forall I \in \mathcal{I}$, $g'(I) = g(I)$?

A recent result by Seshadri and Vondrák [25, Theorem 1.7] shows that, for functions g defined on arbitrary $\mathcal{S} \subset 2^{\bar{V}}$, an extension is not guaranteed to exist. But, every submodular partial function that comes up in this paper is one defined over a matroid; not an arbitrary set system. We refine the non-extendability result of Seshadri and Vondrák [25], by proving that a partial function over a matroid can always be extended:

Theorem 2 (Submodular Extension)**.** *For any function $g(S)$ which is submodular over independence system $\mathcal{I}$, there exists an extension $g'(S)$ which is submodular, and has $g'(S) = g(S)$ for $S \in \mathcal{I}$.*

While Theorem 2 is presented in the context of bisubmodularity, the result is of independent interest, especially in the context of testing whether a function is submodular.

However, the extension g' is not guaranteed to be non-negative and monotone. While a directed bisubmodular function can always be reduced a submodular one, efficient greedy algorithms cannot be directly used to maximize the resulting submodular function.

Lemma 2. *There exists $g(S)$ that is non-negative and submodular over a matroid for which no extension $g'(S)$ is non-negative and submodular.*

Lemma 3. *There exists $g(S)$ that is non-negative, monotone, and submodular over a matroid for which no extension $g'(S)$ is non-negative, monotone, and submodular.*

Even in light of Lemmas 2 and 3, we provide sufficient conditions under which directed bisubmodular maximization can be efficiently solved.

Corollary 4. *Let $f(A, B)$ be a monotone, non-negative, directed bisubmodular function. If there exists an extension $f'(A, B)$ on 2^{2V} which is monotone, non-negative, and simple bisubmodular with $f'(A, B) = f(A, B)$ for $(A, B) \in 3^V$, then there is a constant-factor approximation algorithm for maximizing f.*

Proof. Maximizing $f(A, B)$ reduces to maximizing $g(S)$ over a partition matroid. $g(S)$ is monotone, non-negative, and submodular over the partition matroid, but undefined elsewhere. The existence of f' implies there exists a non-negative, monotone, and submodular extension of $g(S)$, $g'(S)$, which is defined for all $S \subseteq V$. The existence of g' suffices to ensure Fisher et al. [6] yield a near-optimal solution. □

Note that while f' must exist for the corollary to hold, we do not need to be able to construct or evaluate f'.

Testing whether $g(S)$ has a monotone, non-negative extension is a linear programming problem. However, it is an open question as to whether the resulting (exponentially large) linear program is one which can be efficiently solved.

Theorem 3. *Given a matroid $\mathcal{I}$ and a submodular function g defined $\forall I \in \mathcal{I}$, testing whether g has an extension g' which is submodular, monotone, and non-negative is a linear programming feasibility problem.*

Proof. The proof is constructive. For $g'(S)$ to be submodular it is both necessary and sufficient to have for every S, $i \in (V \setminus (S+j))$

$$g'(S+i) - g'(S) \geq g'(S+i+j) - g'(S+j)$$

This is an alternate definition of submodularity [22]. Then a non-negative, submodular extension exists iff the following linear feasibility constraints are satisfied ($g_S = g(S)$, $g'_S = g'(S)$):

$$\begin{aligned} g'_S &= g_S \ \ \forall S \in \mathcal{I}, \\ g'_S &\geq 0 \ \ \forall S \notin \mathcal{I}, \\ g'_{S+i} - g'_S &\geq g'_{S+i+j} - g'_{S+j} \ \ \forall S \subset V, i \in (V \setminus (S+j)). \end{aligned}$$

If monotonicity is also required this can be encoded as additional constraints $g'_A \leq g'_B \ \forall A \subseteq B$. □

4.2 Coordinate-wise Maximization

If f is directed bisubmodular and we wish to maximize under the cardinality constraint $|A| \leq k_1$, $|B| \leq k_2$ for $k_1, k_2 \in \mathbb{Z}_+$, then an alternate approach, which makes no use of the theoretical results is Section 4.1, is to solve the following set of submodular optimizations:

$$\begin{aligned} A^* &= \operatorname*{argmax}_{A \subseteq V : |A| \leq k_1} f(A, \emptyset), \\ B^* &= \operatorname*{argmax}_{B \subseteq V : |B| \leq k_2} f(A^*, B \setminus A^*), \\ B^{**} &= \operatorname*{argmax}_{B \subseteq V : |B| \leq k_2} f(\emptyset, B), \\ A^{**} &= \operatorname*{argmax}_{A \subseteq V : |A| \leq k_1} f(A \setminus B^{**}, B^{**}), \end{aligned}$$

taking the better of (A^*, B^*) and (A^{**}, B^{**}). Budget constraints of the form $|A|+|B| \leq k$ can be handled using a search over budget divisions (Section 3.2). Unlike coordinate-wise maximization on a simple bisubmodular function, no near-optimality guarantee is provided.

5 Coupled Sensor Placement

In this section, we generalize the sensor placement problem to allow for two different kinds of sensors. The goal is to cover a floor plan using sensors with a fixed sensing, or coverage, radius. Figure 1 depicts two of the layouts tested. The potential sensing locations V consist of points on an 50 by 50 grid of a floor plan.

Selected locations can be instrumented with sensors of type A or type B. Sensors cover a circular area, but walls in the environment block coverage. Type A sensors have a sensing radius of r_A units; type B sensors have a radius of r_B units. The costs to deploy a sensor of each type are $c_A, c_B \in \mathbb{Z}_+$. Each location can be instrumented with at most one sensor. The goal is to maximize the overall coverage $f(A, B)$ given a budget $k \in \mathbb{Z}_+$. In our experiments, $r_A = 20$ and $r_B = 10$, with deployment costs $c_A = 3$, $c_B = 1$. In terms of cost-per-area covered, large sensors are better, but less flexible—i.e., using larger sensors to cover narrow hallways would be wasteful.

The objective $f(A, B)$, the surface area covered, is monotone and simple bisubmodular over 2^{2V}. Since each location can hold at most one sensor, f is restricted to 3^V, and is therefore also directed bisubmodular. The optimization is

$$\begin{aligned} &\max_{A,B} && f(A, B) \\ &\text{subject to} && (A, B) \in 3^V, \ c_A|A| + c_B|B| \leq k. \end{aligned}$$

We compare three approaches: (i) only use small sensors, $B = \emptyset$; (ii) only use large sensors, $A = \emptyset$; (iii) allow a mix of both types of sensors. Approaches (i) and (ii) are instances of submodular maximization under a cardinality constraint; approach (iii) is solved using the reduction technique (Section 3.1).

The reduction yields a submodular optimization subject to a matroid and a knapsack constraint, which could be solved using Chekuri et al. [4]. However, this method is more complicated than the standard greedy algorithm. Instead, we convert the knapsack constraint into a partition matroid constraint by searching over divisions of the budget (e.g., if $k = 11$, the budget divisions include one large sensor and 10 small ones; or 2 large sensors, and 9 small ones; etc.). Within a budget division, we simultaneously maximize A and B, without resorting to coordinate maximization.

5.1 Results

Table 5.1 shows our results for the two example layouts and budgets of 15 and 30 (the first row shows results for layout 1 and a budget of 15). We report results in terms of percentage of coverage relative to the best method for that layout / budget combination (i.e., the rows of the table are normalized). Not surprisingly, the method using both kinds of sensors performs the best. This method in fact runs the other two methods

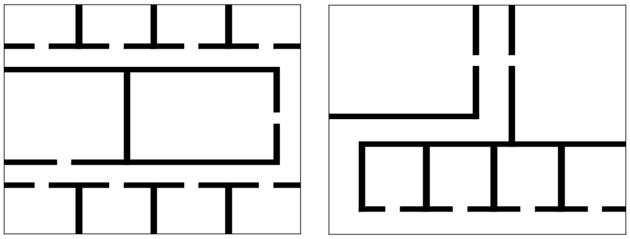

Figure 1: Layouts For Sensor Placement Experiments

Table 1: Percent Coverage (Relative to Best Method)

Problem	Small Sensors	Large Sensors	Both
1 / 15	89.27	82.52	100.00
1 / 30	97.35	96.29	100.00
2 / 15	91.20	86.92	100.00
2 / 30	95.91	88.07	100.00

as subroutines (using only large sensors is one possible allocation of the budget) and is therefore always at least as good as the other two methods. Figure 2 shows the results for layout 1 and a budget of 15. Sensor locations are shown in red, and covered area is shown in blue. Here the best placement of sensors uses small sensors to cover the small rooms and narrow hallways of the environment and large sensors for the larger rooms. The two type sensor method seems to perform significantly better in situations like this.

The differences in performance between the three approaches is necessarily dependent on the floor layout, the budget, and the relative cost/coverage of the sensor types. There are layouts where the value of using both sensors is less dramatic, or non-existent.

6 Coupled Feature Selection

We are given a Gaussian graphical model, depicted in Figure 3, with two variables to predict: C_1, C_2. Given a set of features V, the goal is to select and partition the features into two sets, A and B, such that C_1 is predicted using only features in A; and C_2 is predicted using only features in B. Communication constraints preclude transmitting features between nodes C_1 and C_2. However, local predictions (the value of C_1 and C_2) can be transmitted between nodes.

If we ignore the correlation between C_1 and C_2, then one criterion for feature selection is mutual information, which is submodular under the Naïve Bayes model: $I(A; C_1) = H(A) - \sum_i H(A_i \mid C_1)$. To exploit the correlation between tasks, we use the mutual information

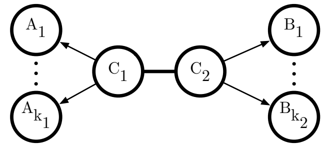

Figure 3: Coupled Feature Selection model.

of the underlying Gaussian graphical model:

$$\begin{aligned} f(A,B) &= I(A,B;C) = H(A,B) - H(A,B|C) \\ &= H(A,B) - \sum_i H(A_i|C_1) - \sum_j H(B_j|C_2), \end{aligned}$$

which we refer to as *biset mutual information*. Maximizing f is equivalent to choosing features for the two tasks that are maximally informative about both tasks. Using f as the coupled feature selection criterion yields a bisubmodular function maximization under the budget constraint $|A| + |B| \leq k$.

Without assumptions on the form of f, the problem is an instance of subset selection in a polytree directed graphical model, which is known to be $\mathbf{NP}^{\mathbf{PP}}$-complete [14]. However, we can show that when f is restricted to 3^V it is directed bisubmodular, under a relatively broad class of models.

Theorem 4. *Assume $A \cup B$ are mutually conditionally independent given C for any $A \subseteq V$ and $B \subseteq V$. $f(A, B) = H(A, B) - H(A, B \mid C)$ is directed bisubmodular, normalized, and monotone non-decreasing.*

Proof. Evaluate $f(\emptyset, \emptyset)$ to establish normalization. Monotonicity follows from the chain rule for mutual information. Let $R = A \cup B$ and $R' = A' \cup B'$ for $(A, B), (A', B') \in 2^{2V}$. $I(R, R'; C) - I(R; C) = I(R'; C \mid R) \geq 0$. Consider $f(A, B) = H(A, B) - H(A, B \mid C)$. By the conditional independence assumption, $H(A, B \mid C)$ is modular in its arguments. $H(S)$ where $S = A \cup B$ is submodular, so $H(S)$ is simple bisubmodular. The difference of a simple bisubmodular function and a modular one is simple bisubmodular. Restricting $f(A, B)$ to 3^V yields a directed bisubmodular function, by Corollary 1. □

Restricting A and B to be disjoint may make sense for some feature selection problems. For example, if C_1 and C_2 predict the weather for two different geographic regions and the features V correspond to different physical sensors, selecting a feature for both tasks would correspond to placing the same sensor in two different geographic regions, which is clearly impossible. A similar proof shows that when f is defined over 2^{2V} it is also simple bisubmodular.

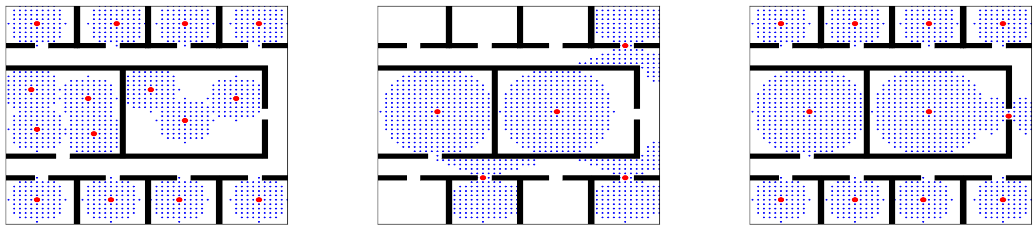

Figure 2: Left: Only Small Sensors, Middle: Only Large Sensors Right: Both

6.1 Results

We have proposed two algorithms for maximizing a simple or directed bisubmodular function: a slow, coordinate-wise maximization, and a fast reduction to submodular maximization. The reduction to submodular maximization is always faster, and it can also yield a result closer to the optimum in practice. In Section 5 the comparison was between the same algorithm on different ground sets (only small sensors, only large sensors, both type of sensors). In this section, the comparison is between two different algorithms on the same instance of bisubmodular max.

To illustrate, we generate random instances of Gaussian graphical models by randomly generating inverse covariance matrices which respect the structure in Figure 3. There are twenty features, half connected to C_1; half connected to C_2. A positive correlation is fixed in the potential on (C_1, C_2). All other parameters are drawn from $\mathcal{U}[-0.5, 0.5]$, with a rejection test to ensure that the resulting matrix is positive semi-definite.

Figure 4 compares the quality of the approximation produced by the two algorithms. The y-axis is scaled so that performance is measured as a percent of the unconstrained maximum of $f(A, B)$. The standard error bars reflect variation due to averaging results across randomly generated Gaussian graphical models. The faster reduction based algorithm achieves a result closer to the optimum,

Note that the fast algorithm is approximately optimal while the slower coordinate-wise algorithm is likely not. This is because the coordinate-wise algorithm has only been shown approximately optimal when maximizing over 2^{2V} while here we maximize over 3^V

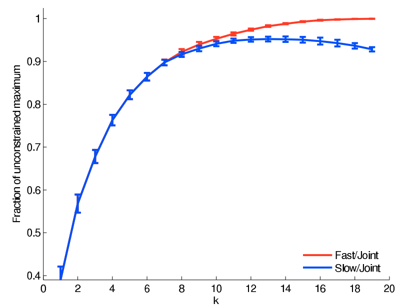

Figure 4: Coupled Feature Selection, comparison of the approximation quality of the fast reduction algorithm (red) vs. the slow coordinate-wise optimization (blue) across all budgets k. Error bars cover 2-std errors.

7 Related Work

Other related work has considered sensor placement problems involving more than one type of sensor [11, 8, 3, 19, 21]. However, the majority of this work doesn't make a connection to submodular function maximization. Note that Fusco and Gupta [8] even derive approximation guarantees for a greedy algorithm without connecting the problem to submodularity. Leskovec et al. [19] and Mutlu et al. [21] do make this connection; these authors pose the problem as a submodular maximization problem with a knapsack constraint. The knapsack constraint allows for different sensors to have different costs. Our work is distinct from this previous work, however, in that we pose our problems as optimization problems over two argument set functions (specifically bisubmodular set functions).

Other work has also considered applications of submodular maximization subject to partition matroid constraints [9, 17]. We note that Golovin et al. [9] in particular considers maximization algorithms which only evaulate $f(S)$ within the constraint set. However, this algorithm still requires that f is submodular everywhere (i.e. that you can reason about the value of $f(S)$ outside of the constraint set).

8 Conclusions

We believe that bisubmodularity is theoretically interesting, and potentially, a broadly useful approach to generalizing value-of-information problems. We have derived the first efficient algorithms for a wide range of bisubmodular maximizations—a requisite step in promulgating this class of problems in the machine learning community.

However, there are still theoretical and applied contributions to be made: (i) we have provided sufficient conditions under which directed bisubmodular functions can be efficiently maximized, but the necessary conditions remain unknown; (ii) building a catalogue of directed bisubmodular functions of interest to the machine learning community, especially ones that have no simple bisubmodular analogue. A topic of particular interest is developing an analogue of directed bisubmodularity for multiset functions, where the objective is a function of $r > 2$ set arguments.

Appendix

Proof of Lemma 1. We show that for any $A \subseteq B \subseteq \bar{V}$ and $s \notin B$,

$$g(A+s) - g(A) \geq g(B+s) - g(B)$$

The result follows directly from the definition of simple bisubmodularity. In particular, define

$$(A_1, A_2) \triangleq (\text{abs}(A \cap V_1), \text{abs}(A \cap V_2)),$$
$$(B_1, B_2) \triangleq (\text{abs}(B \cap V_1), \text{abs}(B \cap V_2)).$$

Then either

$$\begin{aligned} & g(A+s) - g(A) \\ &= f(A_1 + \text{abs}(s), A_2) - f(A_1, A_2) \\ &\geq f(B_1 + \text{abs}(s), B_2) - f(B_1, B_2) \\ &= g(B+s) - g(B) \end{aligned}$$

or

$$\begin{aligned} & g(A+s) - g(A) \\ &= f(A_1, A_2 + \text{abs}(s)) - f(A_1, A_2) \\ &\geq f(B_1, B_2 + \text{abs}(s)) - f(B_1, B_2) \\ &= g(B+s) - g(B) \end{aligned}$$

□

Proof of Theorem 2. For subsets of size $|S| \leq 1$ define $g'(S) = g(S)$ (we assume $\mathcal{I}$ contains all singletons [1]). For $|S| > 1$ we define $g'(S)$ recursively over subsets of increasing size. For any size $k > 2$ define $g'(S)$ for $|S| = k$ as follows: if $S \in \mathcal{I}$ then

$$g'(S) \triangleq g(S)$$

else

$$g'(S) \triangleq \min_{X \subset S, Y \subset S : S = X \cup Y} g'(X) + g'(Y) - g'(X \cap Y)$$

[1] If $\mathcal{I}$ doesn't contain all singletons, we can simply shrink the ground set by removing all singletons not in $\mathcal{I}$

We show that $g'(S)$ is submodular. Consider any A, B with $A \neq B$ and therefore $A \subset A \cup B$ and $B \subset A \cup B$. If $(A \cup B) \in \mathcal{I}$ then we have that

$$\begin{aligned} g'(A) + g'(B) &= g(A) + g(B) \\ &\geq g(A \cup B) + g(A \cap B) \\ &= g'(A \cup B) + g'(A \cap B) \end{aligned}$$

using the submodularity of g over $\mathcal{I}$. If $(A \cup B) \notin \mathcal{I}$ then we have that

$$g'(A \cup B) \leq g'(A) + g'(B) - g'(A \cap B)$$

which implies the desired inequality. For any A, B with $A = B$ we have trivially

$$g'(A) + g'(B) = g'(A \cup B) + g'(A \cap B)$$

□

We show that it is not always possible to extend a function $g(S)$ that is submodular over a matroid such that non-negativity is preserved. We also show a similar result for monotonicity.

Proof of Lemma 2. Define $g(S) = k - |S|$. This is a submodular function (in fact modular), and for all S with $|S| \leq k$ (i.e. all S in a uniform matroid), $g(S)$ is non-negative. However, to extend $g(S)$ to S with $|S| > k$ we must necessarily use negative values. □

Proof of Lemma 3. Consider the following submodular $g(S)$ defined for all $|S| \leq 2$ with $S \subseteq \{1, 2, 3\}$.

$$\begin{array}{lll} g(\emptyset) = 0 & & \\ g(\{1\}) = 1 & g(\{2\}) = 1 & g(\{3\}) = 1 \\ g(\{1,2\}) = 1 & g(\{2,3\}) = 2 & g(\{1,3\}) = 1 \end{array}$$

An extension $g'(S)$ must assign a value to $g'(\{1, 2, 3\})$ We must have

$$g'(\{1,2,3\}) \geq g'(\{2,3\}) = 2$$

in order to maintain monotonicity. But we must also have

$$g'(\{1,2,3\}) \leq g'(\{1,2\}) + g'(\{1,3\}) - g'(\{1\}) = 1$$

in order to maintain submodularity. Clearly we can't have both. □

Acknowledgements

This material is based upon work supported by NSF under grant IIS-0535100, by NIH award R01 EB007057, and by an Intel, a Microsoft, and a Google research award. The opinions expressed in this work are those of the authors and do not necessarily reflect the views of the funders.

References

[1] K. Ando, S. Fujishige, and T. Naitoh. A characterization of bisubmodular functions. *Discrete Mathematics*, 148(1-3):299 – 303, 1996.

[2] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6), 2011.

[3] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *Computers, IEEE Transactions on*, 51(12):1448–1453, 2002.

[4] C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *FOCS*, 2010.

[5] U. Feige, V. S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. In *FOCS*, 2007.

[6] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions II. *Polyhedral combinatorics*, pages 73–87, 1978.

[7] S. Fujishige and S. Iwata. Bisubmodular function minimization. In *Integer Programming and Combinatorial Optimization*, 2001.

[8] G. Fusco and H. Gupta. Selection and orientation of directional sensors for coverage maximization. In *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009.

[9] D. Golovin, A. Krause, and M. Streeter. Online learning of assignments that maximize submodular functions. In *NIPS*, 2009.

[10] P. R. Goundan and A. S. Schulz. Revisiting the greedy approach to submodular set function maximization. Working paper, MIT, 2007.

[11] E. Hörster and R. Lienhart. On the optimal placement of multiple visual sensors. In *4th ACM International Workshop on Video Surveillance and Sensor Networks*, 2006.

[12] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[13] C. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Ops Research*, 43(4):684–691, 1995.

[14] A. Krause and C. Guestrin. Optimal nonmyopic value of information in graphical models - efficient algorithms and theoretical limits. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1339–1345, 2005.

[15] A. Krause, H. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *JMLR*, 2008.

[16] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

[17] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin. Simultaneous placement and scheduling of sensors. In *Information Processing in Sensor Networks*, 2009.

[18] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing nonmonotone submodular functions under matroid or knapsack constraints. *SIAM Journal on Discrete Mathematics*, 23(4), 2010.

[19] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

[20] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *NAACL/HLT*, 2010.

[21] B. Mutlu, A. Krause, J. Forlizzi, C. Guestrin, and J. Hodgins. Robust, low-cost, non-intrusive sensing and recognition of seated postures. In *UIST*, 2007.

[22] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming*, 14:265–294, 1978.

[23] L. Qi. Directed submodularity, ditroids and directed submodular flows. *Mathematical Programming*, 42:579–599, 1988.

[24] L. Qi. Bisubmodular functions. CORE Discussion Papers 1989001, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 1989.

[25] C. Seshadri and J. Vondrák. Is submodularity testable? In *Innovations in Computer Science (ICS)*, 2011.