Learning Fourier Sparse Set Functions

Peter Stobbe
Caltech

Andreas Krause
ETH Zürich

Abstract

Can we learn a sparse graph from observing the value of a few random cuts? This and more general problems can be reduced to the challenge of learning set functions known to have sparse Fourier support contained in some collection $\mathcal{P}$. We prove that if we choose $O(k \log^4 |\mathcal{P}|)$ sets uniformly at random, then with high probability, observing any k-sparse function on those sets is sufficient to recover that function exactly. We further show that other properties, such as symmetry or submodularity imply structure in the Fourier spectrum, which can be exploited to further reduce sample complexity. One interesting special case is that it suffices to observe $O(|E| \log^4 (|V|))$ values of a cut function to recover a graph. We demonstrate the effectiveness of our results on two real-world reconstruction problems: graph sketching and obtaining fast approximate surrogates to expensive submodular objective functions.

1 Introduction

Suppose we wish to sketch the evolution of a massive network: We are given a sequence of networks, where between each step, only few edges get added or removed. Can we compute a small number of statistics, which allow, in hindsight, to reconstruct which edges got added or removed, without storing the entire network? In this paper, we show that this is possible, by observing and storing a small number of random cuts and their values.

Formally, we consider the problem of learning a set function f (mapping subsets of some finite set V of size n to the real numbers) by observing its value on a few sets. Without observing any structure, we clearly need an exponential (in $|V|$) number of observations to approximate the function well over all sets, so we need an appropriate regularity condition. In this paper, we consider the situation where f is *smooth* in the sense of having a decaying Fourier (Hadamard-Walsh) spectrum. One natural example of this is the cut function of a (possibly directed) graph, or generalized additively independent (GAI) functions (Fishburn, 1967), that decompose into a sum of local terms.

By leveraging recent results from sparse recovery (Vershynin, 2010), we show that if the function is sparse in the Fourier domain, having at most k nonzero coefficients, and support contained in a known collection $\mathcal{P}$ of size p, then it is possible to efficiently recover the function exactly from very few samples. In particular, suppose we pick $O(k \log^4 (p))$ sets uniformly at random. Then with very high probability (over this random choice), observing the values of the function on these sets is sufficient to *exactly* reconstruct it.

Besides decaying Fourier spectrum, many set functions encountered in practice satisfy additional properties. In particular, we consider *submodular* functions, which form a natural discrete analogue of convex functions (Lovasz, 1983). Submodularity is satisfied by numerous set functions encountered in practice, such as the cut function in graphs (Schrijver, 2004), entropy (Kelmans & Kimelfeld, 1980), mutual information (Krause, Singh, & Guestrin, 2008) etc. The problem of learning submodular functions has received considerable attention recently (Goemans, Harvey, Iwata, & Mirrokni, 2009; Balcan & Harvey, 2011). However, approximating a submodular function by a factor better than $\sqrt{n}/\log n$ uniformly over all sets requires an *exponential number* of function evaluations, even if those can be adaptively chosen (Goemans et al., 2009). We show that submodularity implies certain structure in the Fourier domain, which can be exploited to reduce the number of required samples even further.

Besides allowing to *sketch the evolution of large graphs* by observing the value of a few random cuts, as mentioned above, our results show that practically relevant set functions, such as certain *valuation functions*, a fundamental concept in economics capturing substi-

Appearing in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

tutability of certain products, can be efficiently learned from few examples. Another natural application is in speeding up submodular optimization: Standard algorithms assume that the function f is presented by an *oracle*, which evaluates f on any set. In general, evaluating f can be very costly (requiring the solution of a large linear system, or perform large-scale simulations). In such a setting, if f is Fourier-sparse, we can approximate it compactly using a small number of random sets, and then optimize the compact representation instead.

In summary, our main contributions are:

- We show that it is possible to learn Fourier k-sparse set functions exactly using $O(k \log^4(p))$ random samples. This reconstruction is robust to noise.
- We show that properties such as symmetry and submodularity of f imply structure in the Fourier domain, which can be exploited to obtain further reduction in sample complexity.
- We demonstrate our algorithm on a problem of sketching the evolution of a graph, and on approximate submodular optimization.

2 Background

Throughout this paper, we refer to a finite ground set V of cardinality n. We will use the letters A, B, C for subsets of V, and the letters s, t for elements of V. Also we use the shorthand $A + s := A \cup \{s\}$ and $s + t := \{s, t\}$.

We consider real-valued set functions, i.e., functions mapping subsets of V to the reals, $f : 2^V \to \mathbb{R}$. Let $\mathcal{H}$ be the space of all such functions, with corresponding inner product: $\langle f, g \rangle := 2^{-n} \sum_{A \in 2^V} f(A)g(A)$. Suppose we are given the value of f on a number of m subsets $A_1, \ldots, A_m \subseteq V$. For now, let us assume these observations are noise free – we will relax this condition later. Under what conditions can we hope to recover $f \in \mathcal{H}$? Clearly, without any assumptions about f, we need an exponential number (in n) of samples in order to obtain exact reconstruction. However, if f is smooth in some way, we may hope to do better. Similar as for continuous functions, a natural smoothness condition is decaying Fourier spectrum.

The Fourier transform on set functions. Set functions can equivalently be represented as real-valued functions of boolean vectors, known as pseudoboolean functions. Just as the set of boolean vectors $\{0, 1\}^n$ forms the commutative group $\mathbb{Z}_2^n$ under addition modulo 2, the power set 2^V forms an equivalent group under the operation of symmetric set difference: $A \ominus B := (A \setminus B) \cup (B \setminus A)$. So the space $\mathcal{H}$ has a natural Fourier (also called Hadamard-Walsh) basis, and in our set function notation the corresponding Fourier basis vectors are:

$$\psi_B(A) := (-1)^{|A \cap B|}.$$

We denote the Fourier transform:

$$\hat{f}(B) := \langle f, \psi_B \rangle = 2^{-n} \sum_{A \in 2^V} f(A)(-1)^{|A \cap B|}.$$

Note that the sum in this definition has exponentially many terms, so it is not practical to evaluate directly. As with any orthonormal basis, we have a reconstruction formula: $f(A) = \sum_{B \in 2^V} \hat{f}(B)\psi_B(A)$.

The Fourier support of a set function is the collection of subsets with nonzero Fourier coefficient: $\operatorname{Supp}[\hat{f}] := \{B \in 2^V : \hat{f}(B) \neq 0\}$. Given a collection of subsets $\mathcal{P} \subseteq 2^V$, let $\mathcal{H}_{\mathcal{P}} := \{f \in \mathcal{H} : \operatorname{Supp}[\hat{f}] \subseteq \mathcal{P}\}$ be the subspace with Fourier support contained in $\mathcal{P}$. We assume we have some a priori knowledge about the Fourier support which gives a natural choice for $\mathcal{P}$. We discuss this in further detail in Section 4, but for now assume $\mathcal{P}$ is some known collection of polynomial size. One illustrative example is the collection of sets of size d or less: $\mathcal{P}_d := \{B \subseteq V : |B| \leq d\}$. As it is particularly important, we denote this function space, consisting of all functions of order d or less, by the symbol $\mathcal{H}_d$. The number of free parameters is $p = \sum_{l=0}^{d} \binom{n}{l}$, which is not too large when $d = 2$.

Now with $\mathcal{P}$ fixed, suppose we restrict ourselves to $f \in \mathcal{H}_{\mathcal{P}}$. Can we recover f with a subexponential number of samples? In the next section, we show that if the Fourier support is small, then this is indeed possible, by leveraging recent results from sparse reconstruction.

3 Conditions for Recovery

Since a set function is uniquely determined by its Fourier transform, recovering a Fourier-sparse function can be thought of as recovery of a sparse vector in $\mathbb{R}^{2^n}$. For large n, even representing such vectors will be intractable. However, if we know that the Fourier support of a function is contained in $\mathcal{P}$, then instead we treat $\hat{f}$ as a sparse vector in $\mathbb{R}^p$. We will show that it is possible to uniquely recover *any* $f \in \mathcal{H}_{\mathcal{P}}$ with $|\operatorname{Supp}[\hat{f}]| \leq k$ by observing the values $f_{\mathcal{M}}$ (with high probability over the choice of measurement sets $\mathcal{M}$), provided that

$$m = O(k \log^4(p)).$$

Matrix vector notation. In the problems that we consider, we observe the function f evaluated on sets from a measurement collection $\mathcal{M} = \{A_i\}$ of size m.

We arrange these measurements in a vector $\boldsymbol{f}_{\mathcal{M}} \in \mathbb{R}^m$, where $\boldsymbol{f}_{\mathcal{M}}[i] := f(A_i)$ for $i = 1 \dots m$. Note the bold typeface used to distinguish vectors from set functions. Furthermore, we will assume that the Fourier support is contained in a known potential support collection $\mathcal{P} = \{B_j\}$ of size p. We denote $\hat{\boldsymbol{f}}_{\mathcal{P}} \in \mathbb{R}^p$ for the the corresponding vector of Fourier coefficients, where $\hat{\boldsymbol{f}}_{\mathcal{P}}[j] := \hat{f}(B_j)$ for $j = 1 \dots p$. Lastly, we denote $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}$ for the $m \times p$ matrix which relates the two vectors,

$$\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}[i,j] := \psi_{B_j}(A_i) = (-1)^{|A_i \cap B_j|}. \tag{3.1}$$

Then for $f \in \mathcal{H}_{\mathcal{P}}$ we have:

$$\boldsymbol{f}_{\mathcal{M}} = \boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}} \hat{\boldsymbol{f}}_{\mathcal{P}}. \tag{3.2}$$

So recovery of f is equivalent to recovery of a sparse vector from linear measurements.

Restricted Isometry. The problem of finding a k-sparse vector from an underdetermined linear system has received significant attention in the context of *compressive sensing* (Candes, Romberg, & Tao, 2006; Donoho, 2006). A sufficient condition for recovery is that the sensing matrix satisfies a key property, the *Restricted Isometry Property (RIP)*. In order to ensure that our measurement matrix $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}$ satisfies this property, we simply choose the measurement sets $\mathcal{M} = \{A_1, \dots, A_m\}$ uniformly at random. Then, as we will see below, results in random matrix theory imply that with high probability (for any fixed $\mathcal{P}$), the measurement matrix indeed satisfies RIP. This insight opens up a vast collection of tools from compressive sensing for the purpose of recovering set functions.

Define the kth restricted isometry constant δ_k for a matrix $\boldsymbol{\Phi}$ as:

$$\begin{aligned}\delta_k(\boldsymbol{\Phi}) := \min\{\delta : \forall \boldsymbol{x}, \mathrm{Supp}[\boldsymbol{x}] \leq k \\ (1-\delta)\|\boldsymbol{x}\|^2 \leq \|\boldsymbol{\Phi x}\|^2 \leq (1+\delta)\|\boldsymbol{x}\|^2\}\end{aligned} \tag{3.3}$$

So if $\delta_k(\boldsymbol{\Phi})$ is small, then $\boldsymbol{\Phi}$ acts approximately as an isometry on k-sparse vectors. An easy consequence of this definition is that the linear measurement vector $\boldsymbol{y} = \boldsymbol{\Phi x}_0$ uniquely determines any k-sparse $\boldsymbol{x}_0$ iff $\delta_{2k} < 1$. Furthermore, with a stronger assumption on the isometry constants, the original vector can be recovered by solving a convex optimization problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^p} \|\boldsymbol{x}\|_1, \quad \boldsymbol{\Phi x} = \boldsymbol{y} \tag{3.4}$$

Originally, Candes and Tao (2005) showed that Equation 3.4 recovers any k-sparse $\boldsymbol{x}_0$ if $\delta_{3k} + 3\delta_{4k} < 2$, but this condition has been weakened several times, most recently by Foucart (2010), who gives the condition $\delta_{2k}(\Phi) < 3/(4+\sqrt{6}) \approx .465$. Furthermore, as discussed below, this result can be generalized to noisy measurements.

Main Reconstruction Result As discussed above, RIP is a very powerful property, but it is not easy to check that any given matrix satisifies it. In fact, most constructions are based on choosing measurements with randomness and then calculating the likelihood of RIP. Perhaps the simplest such case is for random matrices with independent subgaussian entries. However, in our case, we are randomly sampling rows from an orthonormal matrix with bounded entries. Fortunately, as shown by Rudelson and Vershynin (2008) and Vershynin (2010), even in this setting, as long as $m = O(k \log^4(p))$, the expectation of the kth RIP constant is small. More recently, Rauhut (2010) demonstrated that RIP for such matrices holds with high probability. Our result below is essentially Theorem 4.4 of Rauhut (2010) as applied to our case of set functions.

Theorem 1 *For a fixed collection $\mathcal{P} = \{B_j\}_{j=1}^p \subset 2^V$, suppose a measurement collection $\mathcal{M} = \{A_i\}_{i=1}^m \subset 2^V$ is chosen by selecting the sets A_i uniformly at random. Define the matrix $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}} \in \mathbb{R}^{m \times p}$ as in Equation 3.1. Then there exist universal constants $C_1, C_2 > 0$ such that if $k \leq p/2$, and $m \geq \max(C_1 k \log^4(p), C_2 k \log(1/\delta))$, except for an event of probability no more than δ, the following holds for* all *$f \in \mathcal{H}_{\mathcal{P}}$:*

For any noise level $\eta \geq 0$ and any noisy vector of measurements $\boldsymbol{y} \in \mathbb{R}^m$ within that noise level: $\|\boldsymbol{y} - \boldsymbol{f}_{\mathcal{M}}\|_2 \leq \eta$, suppose $g \in \mathcal{H}_{\mathcal{P}}$ has Fourier transform vector $\hat{\boldsymbol{g}}_{\mathcal{P}} \in \mathbb{R}^p$ given by :

$$\hat{\boldsymbol{g}}_{\mathcal{P}} = \arg\min_{\boldsymbol{x} \in \mathbb{R}^p} \|\boldsymbol{x}\|_1, \ \|\boldsymbol{y} - \boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}} \boldsymbol{x}\|_2 \leq \eta. \tag{3.5}$$

Then the following bound holds for some universal constants C_3, C_4:

$$\|f - g\|_2 \leq \frac{C_3}{\sqrt{k}} \mu_k(\hat{\boldsymbol{f}}) + \frac{C_4}{\sqrt{m}} \eta, \tag{3.6}$$

where the quantity $\mu_k(\cdot)$ is defined as the ℓ_1 error of the best k-sparse approximation.

$$\mu_k(\boldsymbol{x}) := \min_{\mathrm{Supp}(\boldsymbol{z}) \leq k} \|\boldsymbol{x} - \boldsymbol{z}\|_1 \tag{3.7}$$

In particular, if $\hat{\boldsymbol{f}}_{\mathcal{P}}$ is k-sparse and $\eta = 0$, then $g = f$.

Therefore, we obtain a strong guarantee for efficiently (using convex optimization) learning Fourier-sparse set functions, robust against measurement noise. Note that, up to log factors, this matches lower bounds of Choi, Jung, and Kim (2011), who show that $\Omega(k \log n)$ measurements are necessary for recovery of a k sparse function in $\mathcal{H}_d$ with d fixed.

4 Classes of Set Functions

In general p is superlinear in n, so even though Eq. 3.4 is equivalent to a Linear Program, it will not necessarily lead to an efficient recovery algorithm. In the extreme case, if $\mathcal{P} = 2^V$, then even calculating a single matrix-vector product $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}^T \boldsymbol{y}$ is difficult. So even though the recovery guarantees of Theorem 1 apply to arbitrary collections $\mathcal{P}$, we need to make some further assumptions about our function to get a practical algorithm for recovery.

Symmetric functions. One natural structural property, obeyed by set functions commonly arising in practice, is *symmetry*. That is, for all sets A it holds that $f(A) = f(V \setminus A)$. Examples of functions satisfying this property are the cut function in undirected graphs, as well as the mutual information, both considered in our experiments (Section 6). It turns out that symmetry already implies interesting structure in the Fourier domain:

Proposition 2 *Let f be a symmetric set function. Then for all sets B of odd cardinality, it holds that $\hat{f}(B) = 0$.*

Therefore, symmetry already implies that we can restrict $\mathcal{P}$ only to sets of odd cardinality.

Low order functions. Let $\mathcal{H}_d := \{f \in \mathcal{H} : \forall |B| > d,\ \hat{f}(B) = 0\}$ be the subspace of dth order functions, so called because when written as pseudoboolean functions, they are order d polynomials. Equivalently, these are functions that can be decomposed as a sum of functions each of which depend on at most d elements: $f = \sum_{|B_i| \leq d} g_i(A \cap B_i)$.

Many set functions f are low-order, or well-approximated by a low-order function. Recovery of an order 1 function is equivalent to classical compressed sensing with a Bernoulli measurement matrix[1]. Recovery of a symmetric order 2 function can be thought of recovering a graph from values of a cut function, a problem which received considerable interest, partly due to several problems arising in computational biology (Alon, Beigel, Kasif, Rudich, & Sudakov, 2004; Grebinski & Kucherov, 2000; Choi & Kim, 2010). We can see the correspondance as follows: given a weighted undirected graph $G = (V, E, w)$, define the symmetric cut function:

$$\phi_G(A) := \sum_{s \in A, t \in V \setminus A} w(s,t). \tag{4.1}$$

Then the Fourier transform can be computed explicity:

$$\hat{\phi}_G(B) = \begin{cases} \frac{1}{2}\sum_{s,t \in V} w(s,t), & B = \emptyset \\ -\frac{1}{2} w(s,t), & B = s + t \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

Hence there is a simple linear correspondence between weights of G and the 2nd order Fourier coefficients of ϕ_G. Clearly this correspondence works in reverse, i.e., given any symmetric 2nd order function f, there is a unique graph G such that $f(A) - f(\emptyset) = \phi_G(A)$. In the general case, functions in $\mathcal{H}_d$ can be thought of as cut functions of hypergraphs of degree d, as considered by Bshouty and Mazzawi (2010).

Submodular functions. Another structural property exhibited by many set functions of practical importance is *submodularity*, a natural discrete analogue of convexity (Lovasz, 1983). A set function is submodular if its 2nd order differences are everywhere nonpositive. We define the cone of submodular functions: $\mathcal{H}_- := \{f \in \mathcal{H} : s,t \in V \setminus A \Rightarrow f(A+s+t) - f(A+s) - f(A+t) + f(A) \leq 0\}$.

While submodularity does not immediately restrict the set $\mathcal{P}$ of candidate supports, it immediately implies dependence among the Fourier coefficients, (a subset of) which can be encoded as constraints in the convex program solved during recovery. In particular, submodularity can be *characterized* in the Fourier domain: f is submodular iff for all $|B| = 2$ and $A \subseteq V \setminus B$:

$$\hat{f}(B) + \sum_{\{C : B \subsetneq C\}} \hat{f}(C)\psi_C(A) \leq 0 \tag{4.3}$$

Checking submodularity is no easier in the Fourier domain; it still requires checking at least $2^{n-2}\binom{n}{2}$ inequalities. However, we can get a necessary condition for submodularity.

Proposition 3 *For all $f \in \mathcal{H}_-$, and $|B| = 2,\ B \subset C$,*

$$\hat{f}(B) + |\hat{f}(C)| \leq 0. \tag{4.4}$$

Proof. Given a particular C_1 such that $B \subset C_1$, let $\mathcal{Q} = \{A \in 2^V : \psi_{C_1}(A) = \operatorname{sign}(\hat{f}(C_1))\}$. Sum Equation 4.3 over all $A \in \mathcal{Q}$.

$$|\mathcal{Q}|\hat{f}(B) + \sum_{A \in \mathcal{Q}} \sum_{\{C : B \subsetneq C\}} \hat{f}(C)\psi_C(A) \leq 0 \tag{4.5}$$

$$|\mathcal{Q}|\hat{f}(B) + \sum_{\{C : B \subsetneq C\}} \hat{f}(C) \sum_{A \in \mathcal{Q}} \psi_C(A) \leq 0 \tag{4.6}$$

To further simplify, we use the following fact:

$$\sum_{A \in \mathcal{Q}} \psi_C(A) = \begin{cases} 0 & \text{if } B \subseteq C \text{ and } C \neq C_1 \\ |\mathcal{Q}| \operatorname{sign}(\hat{f}(C_1)) & \text{if } C = C_1 \end{cases}$$

Thus Equation 4.6 reduces to $|\mathcal{Q}|\hat{f}(B) + \hat{f}(C_1)|\mathcal{Q}|\operatorname{sign}(\hat{f}(C_1)) \leq 0$, which is then equivalent to Equation 4.4 as desired.

This has an immediate simple implication about the support of a submodular function:

[1] if we ignore the constant offset $f(\emptyset)$

Corollary 4 *For $f \in \mathcal{H}_-$, if $C \in \text{Supp}[\hat{f}]$, then $B \in \text{Supp}[\hat{f}]$ for all $B \subset C$ with $|B| = 2$.*

Besides providing some intution about the Fourier support of submodular functions, Equation 4.4 gives a relatively simple convex constraint that can be incorporated into our recovery program. In general, adding any valid convex constraint can never increase our recovery error (a simple consequnce of convexity), and in practice it often decreases it.

There is another such useful constraint for any function which is low order in addition to being submodular. We can fully characterize third order submodular functions in terms of $\binom{n}{2}$ inequalities.

Proposition 5 *For all $f \in \mathcal{H}_3$, then $f \in \mathcal{H}_-$ iff for all $|B| = 2$:*

$$\hat{f}(B) + \sum_{s \in V \setminus B} |\hat{f}(B+s)| \leq 0. \tag{4.7}$$

Proof. To show this is necessary for submodularity, apply Equation 4.3 to the set $A = \{s \in V \setminus B : \hat{f}(B+s) < 0\}$. Conversely it is sufficient because for any $A \subseteq V \setminus B$, we have $|\hat{f}(B+s)| \geq \hat{f}(B+s)\psi_C(A)$, which implies Equation 4.3.

5 Reconstruction Algorithms

In Section 3, we have shown that the problem of learning Fourier-sparse set functions can be reduced to the Compressed Sensing paradigm of recovery of a sparse vector from RIP measurements. This insight allows us to open up a cornucopia of algorithms that have been developed for this setup (Tropp & Wright, 2010). In particular, several greedy algorithms such as Orthogonal Matching Pursuit can explicitly take advantage of RIP to guarantee recovery, as shown by Tropp (2004). For our experiments, we take the approach of convex optimization. Rather than solving Eq. 3.4 exactly, we minimize the Lagrangian formulation so that we can apply an accelerated proximal method such as the one by Auslender and Teboulle (2006),

$$\min_{\boldsymbol{x} \in \mathbb{R}^p} \|\boldsymbol{x}\|_1 + \frac{1}{2\lambda} \|\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}} \boldsymbol{x} - \boldsymbol{y}\|^2. \tag{5.1}$$

In our experiments, we use the toolbox TFOCS (Becker, Candes, & Grant, 2011), which requires only that we supply a method to apply $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}$ and $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}^T$. In the case of 2nd order set functions, we do not need to store the entire matrix $m \times p$ matrix, and there is a formula that only requires $O(mn)$ storage. Let $\boldsymbol{\Psi}_{\mathcal{M},d} := \boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}_d}$ be the subsampled $m \times \binom{n}{d}$ Fourier matrix where the columns correspond to the sets of size d. So the matrix $\boldsymbol{\Psi}_{\mathcal{M},1} \in \mathbb{R}^{m \times n}$ is defined by $\boldsymbol{\Psi}_{\mathcal{M},1}[i,j] = \begin{cases} -1 & j \in A_i \\ 1 & j \notin A_i \end{cases}$. If the 2nd order Fourier coeffients from $\boldsymbol{x} \in \mathbb{R}^{\binom{n}{2}}$ are arranged in the off-diagonal elements of an $n \times n$ a matrix $\boldsymbol{X}$, then the elements of $\boldsymbol{\Psi}_{\mathcal{M},2}\boldsymbol{x}$ are the diagonal elements of $\boldsymbol{\Psi}_{\mathcal{M},1}\boldsymbol{X}\boldsymbol{\Psi}_{\mathcal{M},1}^T$, and the transpose operation is $\boldsymbol{\Psi}_{\mathcal{M},2}^T\boldsymbol{r} = \boldsymbol{\Psi}_{\mathcal{M},1}^T \operatorname{diag}(\boldsymbol{r})\boldsymbol{\Psi}_{\mathcal{M},1}$.

Exploiting structure in the Fourier domain. In Section 4, we have shown that submodularity implies constraints about the relative magnitudes of the Fourier coefficients. In addition to encoding this structure into the convex program to improve recovery, this structure can further be exploited to extend our technique to higher order functions (where the collection $\mathcal{P}$ can become intractably large). The key step in most sparse recovery algorithms is to find the largest magnitude elements of $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}^T\boldsymbol{r}$ given a residual vector $\boldsymbol{r}$. For example, first order methods applied to Eq. 5.1 such as the one we use are equivalent to iterative soft-thresholding. While we could find the largest magnitude elements of $\boldsymbol{\Psi}_{\mathcal{M},\mathcal{P}}^T\boldsymbol{r}$ by simply applying the full transformation and sorting, one can use submodularity to avoid having to compute the entire set of higher-order coefficients. For example, if the function is 3rd order and submodular, we can apply Equation 4.7, and note for $|B| = 3$,

$$|\hat{f}(B)| \leq \min_{s \in B} -\hat{f}(B-s) - \sum_{t \in V \setminus B} |\hat{f}(B-s+t)|$$

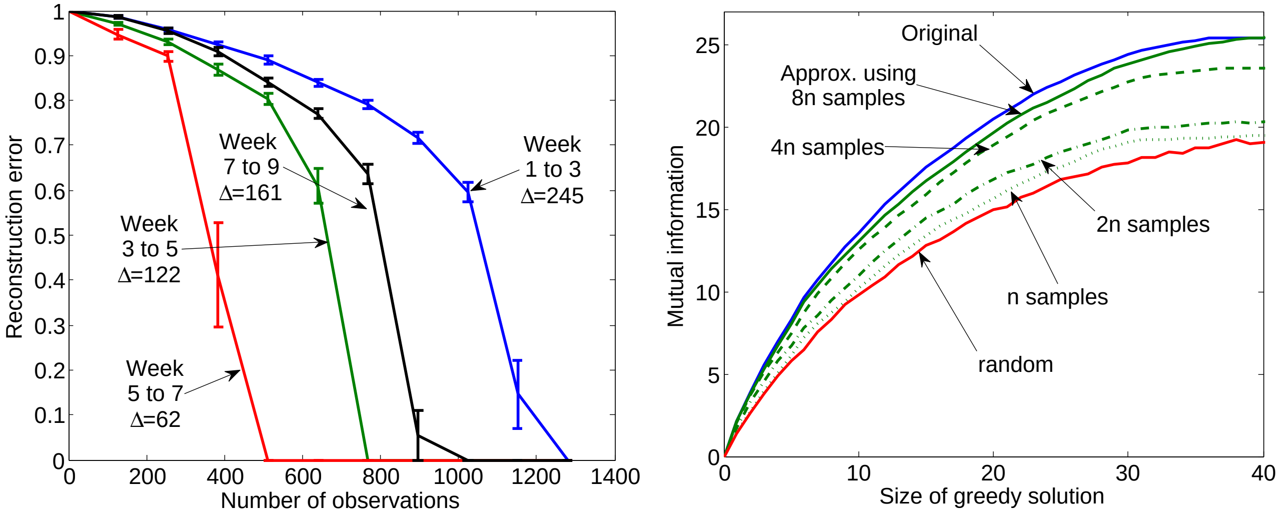
So these constraints can be used to speed up the identification of the largest magnitude coefficients, as we need only compute the 3rd order coefficients with sufficient slack. We leave a detailed investigation of this direction open for future work.

6 Applications and Experiments

We evaluate our approach towards learning set functions on two real-world data sets. We also use synthetic data to demonstrate our claim that enforcing submodularity through convex constraints can improve recovery of submodular functions.

Sketching graph evolution We consider the problem of reconstructing (differences between) graphs by observing random cuts. Suppose we are given a sequence of weighted undirected graphs $G_1 = (V_1, E_1, w_1), \ldots, G_T = (V, E_T, w_T)$ that, w.l.o.g., share the same set of vertices, but differ in the set of edges E_i and their weights w_i. Let $f_i(A) = \phi_{G_i}(A)$ be the the corresponding symmetric cut functions as defined in Eq. 4.1. Note that by (4.2), knowing f_i uniquely determines G_i.

To handle the case of directed graphs, we can define an undirected bipartite graph G' with enlarged ground



(a) *Graph sketching*

(b) *Approximate optimization*

Figure 1: Experimental results. (a) Graph sketching of transitions of the Autonomous Systems graph. We plot number of random cuts observed vs. reconstruction error (Combined Type I + Type II error). During different transitions, the number Δ of changing edges varies. Notice how approximately 8Δ random observations suffice for perfect reconstruction. (b) Approximate submodular maximization in environmental monitoring. We wish to choose sets of locations with maximum mutual information. We compare the greedy algorithm optimizing the true functions, Fourier-sparse reconstructions obtained from n, $2n$, $4n$ and $8n$ samples with random selection. Notice that $8n$ samples already provide performance very close to the true objective.

set $V \times \{1,2\}$. The weight of an edge $w_{(s,1),(s',2)}$ in G' corresponds to the weight of directed edge $w_{s,s'}$ in G, similarly $w_{(s',1),(s,2)}$ corresponds to the opposite direction $w_{s,s'}$. If we can observe directed cuts in G, we can infer undirected cuts in G'. From the reconstructed G' we can the recover G.

As we have observed in Section 5, cut functions are contained in $\mathcal{H}_2$, and there is one edge for each nonzero Fourier coefficient. We can thus use Corollary 1 to reconstruct the graph by observing $\mathcal{O}(|E^{(t)}| \log^4 n)$ values of random cuts. Note that while in practice, typically $|E| = \Omega(n)$, and for large graphs, we would require a proportionally large number of observations. If, however, we are interested in how a graph *changes* over time, and this change happens slowly, we use the fact that the *symmetric difference* $E^{(t)} \ominus E^{(t+1)}$ is sparse.

In our experiments, we take a sequence of five snapshots of the Autonomous Systems graph [2]. Our experiments are performed on the subgraph induced by the 128 nodes with largest degree. We first pick an increasing number of sets at random. We then sketch the graphs at different time steps by computing the cut values associated with those sets. Since the cut function is linear in the edge weights, the difference in cut values corresponds to the cuts in the symmetric graph differences. We can therefore reconstruct the difference in the edge sets by using the reconstruction algorithm described in Section 5. Note that the number of changing edges varies from 62 to 245. Figure 1(a) presents the reconstruction error (in terms of the fraction of edges correctly classified as changing or not changing). For all transitions, exact recovery is possible, using a number of samples that is approximately a factor of 8 larger than the number of changing edges. Also, we observe that consistently with results in compressive sensing, a sharp phase transition occurs between a regime in which the error is close to 100%, and the regime in which perfect reconstruction occurs.

Approximate submodular optimization Suppose a submodular function to be optimized is extremely expensive to evaluate, but can be approximated with our recontruction methods from random samples. Then one can evaluate the function on random samples to construct an approximation, and optimize the approximation. We test this approach on submodular function maximization, in an environmental monitoring application. We consider the problem of selecting a small number of most informative observations for the purpose of spatial prediction. We take temperature data from the NIMS sensor node (Harmon, Ambrose, Gilbert, Fisher, Stealey, & Kaiser, 2006) deployed at a lake near the University of California, Merced. The environment is discretized in a set V of $n = 86$ locations. We train a nonstationary Gaussian Process using data from a single scan of the lake by the NIMS sensor node, using a method described by Krause et al.

[2] downloaded from `http://snap.stanford.edu`

(2008). In order to quantify the informativeness of a set of locations $A \subseteq V$, we use the mutual information

$$f(A) = I(X_A; X_{V \setminus A}) = H(X_{V \setminus A}) - H(X_{V \setminus A} \mid X_A),$$

that quantifies the reduction of uncertainty in the unobserved locations $V \setminus A$ by taking into account the observations X_A at the selected observations. As shown by Krause et al. (2008), f is submodular and approximately monotonic (for small sets A), and therefore an efficient greedy algorithm, adding observations that maximally increase $f(A)$ until k observations have been selected produces a set A_G with near-maximal informativeness (Nemhauser, Wolsey, & Fisher, 1978).

Unfortunately, computing mutual information $f(A)$ for the case of Gaussian processes requires solving a linear system of n variables, which is very expensive for large n. We consider approximating f by a low-order function. We evaluate f on an increasing number of sets, chosen uniformly at random, and then use the algorithm described in Section 5 to approximate $f \in \mathcal{H}_2$. Notice that even though f not exactly sparse, it appears to be well-approximated by a order 2 function: The best order 2 approximation explains approximately 86 % of its variance. In order to determine how well suited the approximate function is for optimization, we run the greedy algorithm on the approximation, and compare the resulting sets with the (provably near-optimal) solutions obtained by running the greedy algorithm on the original (expensive to evaluate) function f. As baseline, we also compare against the performance of sets chosen uniformly at random. Figure 1(b) presents the results of the experiment, using approximations obtained from n, $2n$, $4n$ and $8n$ random function evaluations. Notice that n and $2n$ function evaluations not surprisingly lead to poor performance. However, even 4 samples per location lead to strong performance, and $8n$ samples leads to solutions almost as good as those obtained when working with the true objective. These results indicate that the proposed approximations can perform very well even though the assumption of exact sparsity in the Fourier domain is not met.

Synthetic Submodular Recovery We claimed in Section 4 that if a function is known to be submodular, then incorporating convex constraints implied by submodularity can improve the recovery of a function. We now describe some experiments on synthetic functions that demonstrate this claim empirically. We attempt the recovery a 3rd order submodular function by incorporating the constraints from Equation 4.7 into a convex recovery algorithm.

We take $n = 16$ and restrict to $\mathcal{H}_3$, therefore $p = 697$. By Proposition 5, we can check submodularity with 120 constraints. These numbers are small enough so that we can use a standard interior point method solver to get accurate solutions. We construct f by taking a function with i.i.d. Gaussian entries and then projecting it onto the cone $\mathcal{H}_- \cap \mathcal{H}_3$ to get a target synthetic function. The resulting projection is not exactly sparse, on average it has 200 ± 10 (out of 697 possible) nonzero Fourier coefficients. However, it is compressible, so we can expect a small error even without recovering the support exactly. Then, given random function samples, we reconstruct the target function by minimizing the Fourier ℓ_1 norm, but vary what sets of additional constraints we apply. First, we simply solve Equation 3.4 with no additional constraints. For our second way, we assume oracle access to the signs of the Fourier coefficients and we enforce the known signs of the coefficients. Lastly, we enforce the constraints of Equation 4.7. The results are plotted in Figure 2. Enforcing submodularity significantly improves the recovery. It gives a relative error of less than 10^{-3} with only 300 measurements, and it recovers the support exactly with about 350. Using the ℓ_1 norm alone requires about 450 measurements just to get a relative error or 10^{-3}. The method with oracle access to the signs of the coefficients has better performance than standard ℓ_1, but still not as good as the submodularity-enforcing method.

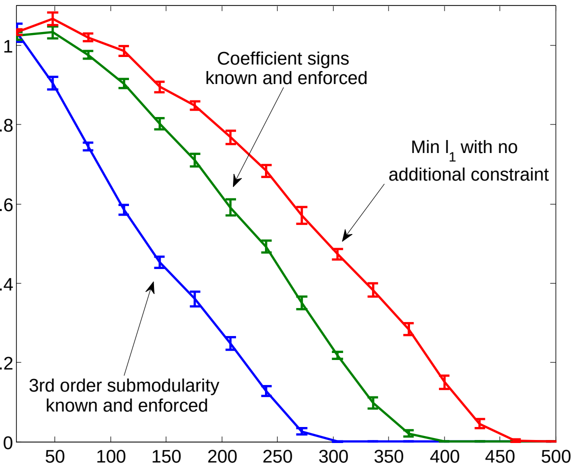


Figure 2: Empirical study of submodular constraints. Synthetic functions on a base set of 16 elements. Attempted recovery with differing types of constraints. The experiments were repeated with different random synthetic functions and different random measurments, and the mean relative error $\|f - g\|_2 / \|f\|_2$ is plotted vs. number of random measurements.

7 Related Work

Fourier analysis on the discrete cube $\{0,1\}^n$ The problem of learning Boolean and pseudoboolean functions has a long history with many special cases that have been studied, and the use of discrete Fourier analysis dates back to the work of Linial, Mansour, and Nisan (1993).

The specific problem of reconstructing graphs from few observations has received attention to important applications in bioinformatics. The literature distinguishes *additive queries* (computing weight of all edges in a subgraph), and less powerful *cross-additive* queries (computing the weight of edges between two sets of vertices). Cuts are a special case of the latter. The literature also distinguishes *adaptive queries* (that can choose observations based on past observations) and less powerful *nonadaptive queries* (that have to commit to all observations in advance). In general, non-adaptive algorithms only requiring cross-additive queries are preferred (as these make the fewest assumptions, can be parallelized, etc.). For graphs with n nodes and k edges, an information theoretic lower bound of $\Omega\big(\frac{k\log(n^2/k)}{\log k}\big)$ additive (possibly adaptive) queries is known. Mazzawi (2010) provides an adaptive polynomial time algorithm that attains this optimal complexity in $\log n$ nonadaptive rounds. To our knowledge, the only existing *non-adaptive* algorithms with linear dependence on k are non-constructive (i.e., not polynomial time) (Bshouty & Mazzawi, 2010). This approach also requires *additive* queries. To our knowledge, ours is the first efficient nonadaptive approach (and furthermore only requires cross-additive queries).

Learning of pseudo-boolean functions (and associated hypergraphs) has been considered by Choi et al. (2011), who provides an almost tight *adaptive* algorithm for computing the Fourier coefficients of k-bounded pseudoboolean functions. Bshouty and Mazzawi (2010) provide a non-adaptive, but also non-constructive approach, requiring additive queries.

Learning submodular functions Unfortunately, even without noise, there are *strong lower bounds*, limiting our expectations on learning *general* submodular functions. Without access to a data set of exponential size, it is not possible to approximate general submodular functions to a factor better than $\Omega(\sqrt{n}/\log n)$ (Goemans et al., 2009). On a more positive side, if the function is Lipschitz, and sets are sampled uniformly at random, then for any $\varepsilon > 0$, a $\mathcal{O}(\log \frac{1}{\varepsilon})$ approximation can be achieved on a fraction of at least $1-\varepsilon$ of all sets (Balcan & Harvey, 2011). However, optimization purposes, a guarantee that the approximation is of high quality on only a subset (even a large subset) of sets is problematic, since typically nothing can be inferred about the resulting minimizer. The problem of approximating a general submodular function by a simpler one for the purpose of efficient minimization is studied by Jegelka, Lin, and Bilmes (2011), who do not exploit the special structure of Fourier-sparse functions.

Compressive sensing There has been vast interest in sparse reconstruction and compressive sensing (Candes & Wakin, 2008). But traditionally this has been motivated by sparsity of signals as a trigonometric polynomial or in the wavelet domain. However, we are unaware of any work directly applying these ideas to discrete cube. If work on sublinear Fourier transforms as of Gilbert, Guha, Indyk, Muthukrishnan, and Strauss (2002) can be thought of as applying the ideas from learning sparse boolean functions to sparse trigonmetric polynomials, then our work can be thought of as doing the reverse. Opening up a toolbox of new methods for this domain is the main contribution of this paper.

8 Conclusion

We have considered the problem of reconstructing set functions with decaying Fourier (Hadamard-Walsh) spectrum, from a small number of possibly noisy observations. By leveraging recent results from random matrices and sparse reconstruction, we have shown that standard algorithms can be used to obtain perfect reconstruction, with a number of samples that scales linearly with the support size of the Fourier spectrum. This insight allows us to open up a vast toolbox of modern optimization methods for learning set functions, which previously has been mostly the domain of purely theoretical investigation. For example, our results imply that standard ℓ_1 minimization can be used to reconstruct a sparse graph from observing the values of a number of random cuts, which (up to logarithmic factors) matches information-theoretic lower bounds by Mazzawi (2010). Furthermore, we show other properties, such as submodularity and symmetry, imply structure among the Fourier coefficients, that can be exploited to reduce sample complexity, as well as speed up reconstruction algorithms. We demonstrate the effectiveness of our approach on two applications, showing that we can indeed sketch changes in real-world networks by measuring random cuts, and that we can obtain useful approximations of expensive-to-compute set functions for the purpose of optimization.

References

Alon, N., Beigel, R., Kasif, S., Rudich, S., & Sudakov, B. (2004). Learning a Hidden Matching. *SIAM Journal on Computing*, *33*(2), 487.

Auslender, A., & Teboulle, M. (2006). Interior Gradient and Proximal Methods for Convex and Conic Optimization.. *SIAM Journal on Optimization*, *16*(3), 697–725.

Balcan, M., & Harvey, N. J. A. (2011). Learning Submodular Functions. In *Proc. STOC*.

Becker, S., Candes, E., & Grant, M. (2011). Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, *3*, 165–218.

Bshouty, N. H., & Mazzawi, H. (2010). Optimal query complexity for reconstructing hypergraphs. *Arxiv preprint arXiv:1001.0405*.

Candes, E., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, *52*(2), 489 – 509.

Candes, E., & Tao, T. (2005). Decoding by linear programming. *Information Theory, IEEE Transactions on*, *51*(12), 4203 – 4215.

Candes, E., & Wakin, M. (2008). An introduction to compressive sampling. *IEEE Signal Processing Magazine*, *25*(2), 21–30.

Choi, S.-S., Jung, K., & Kim, J. H. (2011). Almost Tight Upper Bound for Finding Fourier Coefficients of Bounded Pseudo-Boolean Functions. *Journal of Computer and System Sciences*, *77*(6), 1039–1053.

Choi, S.-S., & Kim, J. H. (2010). Optimal query complexity bounds for finding graphs.. *Artif. Intell.*, *174*(9-10), 551–569.

Donoho, D. (2006). Compressed sensing. *IEEE Trans. on Information Theory*, *52*(4), 1289–1306.

Fishburn, P. C. (1967). Interdependence and additivity in multivariate, unidimensional expected utility theory. *International Economic Review*, *8*, 335–342.

Foucart, S. (2010). A note on guaranteed sparse recovery via l1-minimization. *Applied and Computational Harmonic Analysis*, *29*(1), 97–103.

Gilbert, A. C., Guha, S., Indyk, P., Muthukrishnan, S., & Strauss, M. (2002). Near-optimal sparse fourier representations via sampling. In *Proc. STOC*.

Goemans, M., Harvey, N., Iwata, S., & Mirrokni, V. (2009). Approximating submodular functions everywhere. In *Proc. SODA*.

Grebinski, V., & Kucherov, G. (2000). Optimal Reconstruction of Graphs under the Additive Model.. *Algorithmica*, *28*(1), 104–124.

Harmon, T. C., Ambrose, R. F., Gilbert, R. M., Fisher, J. C., Stealey, M., & Kaiser, W. J. (2006). High Resolution River Hydraulic and Water Quality Characterization using Rapidly Deployable Networked Infomechanical Systems (NIMS RD). Tech. rep. 60, CENS.

Jegelka, S., Lin, H., & Bilmes, J. (2011). Fast approximate submodular minimization. In *Advances in Neural Information Processing Systems (NIPS)*.

Kelmans, A. K., & Kimelfeld, B. N. (1980). Multiplicative submodularity of a matrix's principal minor as a function of the set of its rows and some combinatorial applications. *Discrete Mathematics*, *44*(1), 113–116.

Krause, A., Singh, A., & Guestrin, C. (2008). Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research*, *9*, 235–284.

Linial, N., Mansour, Y., & Nisan, N. (1993). Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM (JACM)*, *40*(3), 607–620.

Lovasz, L. (1983). Submodular functions and convexity. *Mathematical Programming - State of the Art*, 235–257.

Mazzawi, H. (2010). Optimally reconstructing weighted graphs using queries. In *Proc. SODA*.

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, *14*(1), 265–294.

Rauhut, H. (2010). Compressive sensing and structured random matrices. In *Theoretical Foundations and Numerical Methods for Sparse Recovery*, pp. 1–94.

Rudelson, M., & Vershynin, R. (2008). On sparse reconstruction from Fourier and Gaussian measurements. *Communications on Pure and Applied Mathematics*, *61*(8), 1025–1045.

Schrijver, A. (2004). *Combinatorial Optimization*. Springer.

Tropp, J., & Wright, S. (2010). Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, *98*(6), 948–958.

Tropp, J. A. (2004). Greed is good: algorithmic results for sparse approximation.. *IEEE Transactions on Information Theory*, *50*(10), 2231–2242.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *Arxiv preprint arXiv:1011.3027*.