A Walk Around the Torus: SFTs in Many Dimensions

Craig Gross
Department of Mathematics
Michigan State University
grosscra@msu.edu

***Abstract*—We present two ways of using efficient sparse Fourier transforms in one-dimension to produce sparse Fourier approximations in many dimensions. Both techniques rely on applying various simple transformations to the signal, mapping the many dimensions down to one via reconstructing rank-1 lattices, and then synchronizing the results from each transformation. Both results depend only linearly on the dimension and sparsity in runtime complexity with near-optimal error guarantees; they differ by trading off faster runtime for better error guarantees. These theoretical complexity and error bounds are also reflected in numerical experiments for function reconstruction and approximation.**

I. Introduction

The frequency data of a one-dimensional signal can be efficiently computed via the Fast Fourier Transform (FFT) in $\mathcal{O}(N \log N)$ time, where N is the bandwidth of the signal. However, for signals with few frequency components (e.g., s non-zero Fourier coefficients with $s \ll N$), Sparse Fourier Transforms (SFT) can avoid N-linear time/memory requirements, instead running with sub-linear $\mathcal{O}(s \operatorname{polylog}(s, N))$ time/memory [1]–[4]. In the multivariate case, to recover the frequency content of a d-dimensional signal, the traditional approach of applying an FFT in each dimension requires $\mathcal{O}\left(N^d \log^d(N)\right)$, i.e., exhibits the *curse of dimensionality*. Thus, multivariate SFTs become even more attractive if they can maintain sub-linear dependence on the bandwidth while also avoiding exponential blow up in d [5], [6].

Here, we outline two techniques for mapping fast univariate SFTs to fast multivariate SFTs. Both rely on *reconstructing rank-1 lattices*, which constrain the sampling set of a multivariate function so that the signal's samples correspond to those of a univariate function with identical Fourier coefficients [7]–[9]. Applying an SFT to this new univariate function recovers the desired coefficients along with one-dimensional frequencies that are in one-to-one correspondence to their desired d-dimensional counterparts. However, naïevely computing this correspondence incurs at worst the same N^d factor in the time/memory. By instead applying this same rank-1 lattice SFT technique to modifications of the d-dimensional signal, we can more quickly learn this frequency mapping by analyzing coefficient measurements from separate SFT runs.

II. Main Results

Rather than restrict ourselves to discrete Fourier approximation on the cube, we consider the exact Fourier coefficients c of a function $f \in C(\mathbb{T}^d)$ in a specified frequency space $\mathcal{I} \subset \mathbb{Z}^d$ defined as

$$c_{\mathbf{k}} := \int_{\mathbb{T}^d} f(\mathbf{x}) \mathrm{e}^{2\pi \mathrm{i} \mathbf{k}\cdot\mathbf{x}} \, d\mathbf{x} \quad \forall \mathbf{k} \in \mathbb{Z}^d.$$

Below, we will assume that $\mathcal{I}$ is contained within a bandwidth N cube, that is, $\mathcal{I} \subset (-\lceil N/2 \rceil, \lfloor N/2 \rfloor]^d$. A reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ for $\mathcal{I}$ has generating vector $\mathbf{z} \in \mathbb{Z}^d$ and length M such that all mapped frequencies $\mathbf{z}\cdot\mathbf{k} \bmod M$ with $\mathbf{k} \in \mathcal{I}$ are distinct. Since we consider Fourier-sparse or compressible signals, we define $c|_{\mathcal{I}}$ to be the restriction of c to only the frequencies in $\mathcal{I}$ and $(c|_{\mathcal{I}})_s^{\mathrm{opt}}$ to be the restriction of $c|_{\mathcal{I}}$ to its largest s entries.

A. Phase encoding

The first technique we discuss applies lattice SFTs to spatial shifts of the multivariate signal in successive dimensions. By shifting in one dimension at a time, the frequency components in that dimension are embedded as phase shifts in the SFT output and can therefore be used to recover each full d-dimensional frequency.

Theorem 2.1: Running the phase encoding algorithm (see Figure 1 for pseudocode) with input sparsity s on measurements of f corrupted by errors at most $e_\infty > 0$ in absolute magnitude will produce $b \in \mathbb{C}^{\mathbb{Z}^d}$ a $2s$-sparse approximation of c satisfying

$$\|b - c\|_2 \lesssim N \left[\frac{\|c|_{\mathcal{I}} - (c|_{\mathcal{I}})_s^{\mathrm{opt}}\|_1}{\sqrt{s}} + \sqrt{s}(\|c - c|_{\mathcal{I}}\|_1 + e_\infty) \right].$$

This error bound is achieved deterministically with runtime and sampling complexity $\mathcal{O}\left(ds^2 \operatorname{polylog}(|\mathcal{I}|, N)\right)$ and is also achieved with probability $1 - \sigma$ with runtime and sampling complexity $\mathcal{O}\left(ds \operatorname{polylog}(|\mathcal{I}|, N, \sigma^{-1})\right)$.

B. Two-dimensional DFT technique

The second algorithm takes a "leave-one-out" approach when using a rank-1 lattice to map a d-variate signal to a univariate one. Instead, a rank-1 lattice maps $d-1$ dimensions down to one, leaving a two-dimensional signal. An SFT is performed in the lattice dimension, and an FFT in the "left-out" dimension. All frequency components in the FFT dimension can then be easily picked out, and the process is repeated d-times.

Theorem 2.2: Running the two-dimensional DFT algorithm (see Figure 2 for pseudocode) with input sparsity s on measurements of f corrupted by errors at most $e_\infty > 0$ in absolute magnitude will produce $b \in \mathbb{C}^{\mathbb{Z}^d}$ a $2s$-sparse approximation of c satisfying

$$\|b - c\|_2 \lesssim \frac{\|c|_{\mathcal{I}} - (c|_{\mathcal{I}})_s^{\mathrm{opt}}\|_1}{\sqrt{s}} + \sqrt{s}(\|c - c|_{\mathcal{I}}\|_1 + e_\infty).$$

This error bound is achieved deterministically with runtime and sampling complexity $\mathcal{O}\left(dN^2 s^2 \operatorname{polylog}(|\mathcal{I}|, N)\right)$ and is achieved with probability $1 - \sigma$ with runtime and sampling complexity $\mathcal{O}\left(dN^2 s \operatorname{polylog}(|\mathcal{I}|, N, \sigma^{-1})\right)$.

C. Numerics

We now present numerical results which demonstrate these sampling complexity and error estimates for both sparse and compressible signals. In particular, Figure 3 demonstrates the linear scaling in s of the sampling complexity of both randomized algorithms, and Figure 4 demonstrates the same for d. Figure 5 demonstrates the robustness of the phase encoding algorithm to noise and shows that for non-sparse but compressible signals we obtain near-optimal recovery.

Acknowledgments

This is joint work with Mark Iwen (Michigan State University), Lutz Kämmerer (TU Chemnitz), and Toni Volkmer (TU Chemnitz), and is based on the preprint [10]. This research was supported in part by NSF DMS 1912706.

Input: sparsity s; expansion parameter N; reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ for $\mathcal{I}$; d-variate signal f.
Output: $b \in \mathbb{C}^{\mathbb{Z}^d}$ a $2s$-sparse approximation to c.

$\mathbf{v} \leftarrow$ SFT applied to univariate function $t \mapsto f(t\mathbf{z})$
for $\ell = 1$ to d **do**
  $f^\ell(\mathbf{x}) := f(\mathbf{x} + \mathbf{e}_\ell/N)$ where $\mathbf{e}_\ell \in \mathbb{R}^d$ is the canonical basis vector in dimension ℓ.
  $\mathbf{v}^\ell \leftarrow$ SFT applied to $t \mapsto f^\ell(t\mathbf{z})$
end for
for all $\omega \in \mathrm{supp}(\mathbf{v})$ **do**
  for $\ell = 1$ to d **do**
    $k_\ell \leftarrow \mathrm{round}(N \arg(v^\ell_\omega / v_\omega)/2\pi)$
  end for
  $b_{\mathbf{k}} \leftarrow v_\omega$
end for

Fig. 1. Phase encoding algorithm

Input: sparsity s; expansion parameter N; rank-1 lattice $\Lambda(\mathbf{z}, M)$ which is reconstructing for $\mathcal{I}$ and $\{\mathbf{k} - k_\ell \mathbf{e}_\ell \mid \mathbf{k} \in \mathcal{I}\}$ for all $\ell = 1$ to d; d-variate signal f.
Output: $b \in \mathbb{C}^{\mathbb{Z}^d}$ a $2s$-sparse approximation to c.

$\mathbf{v} \leftarrow$ SFT applied to univariate function $t \mapsto f(t\mathbf{z})$
for $\ell = 1$ to d **do**
  for $j = 1$ to N **do**
    Row j of matrix $\mathbf{V}^\ell \leftarrow$ SFT applied to $t \mapsto f(j\mathbf{e}_\ell/N + t(\mathbf{z} - z_\ell \mathbf{e}_\ell))$
  end for
  $\mathbf{F}_N \mathbf{V}^\ell \leftarrow$ length N FFT applied to columns of $\mathbf{V}^\ell$
end for
for all $\omega \in \mathrm{supp}(\mathbf{v})$ **do**
  for $\ell = 1$ to d **do**
    $k_\ell \leftarrow h$, row of closest element $(\mathbf{F}_N \mathbf{V}^\ell)_{h,\omega'}$ to v_ω in absolute magnitude satisfying $hz_\ell + \omega' \equiv \omega \mod M$
  end for
  $b_{\mathbf{k}} \leftarrow v_\omega$
end for

Fig. 2. Two-dimensional DFT algorithm

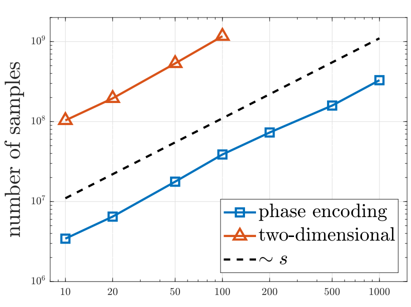

Fig. 3. Comparison in average sampling complexity as the sparsity increases over 100 runs for both randomized algorithms applied to signals with randomly chosen frequencies in a 10-dimensional hyperbolic cross. Both algorithms find the true support of the signal in at least 90% of the runs.

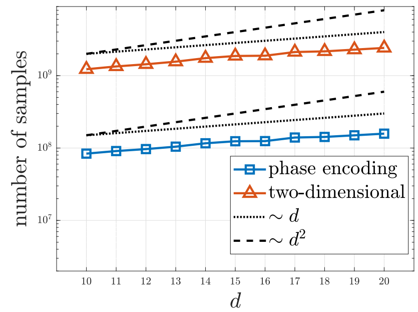

Fig. 4. Comparison in average sampling complexity as the dimension increases over 100 runs for both randomized algorithms applied to 100-sparse signals with randomly chosen frequencies in cuboids of cardinality $\sim 10^{12}$. Both algorithms find the true support of the signal in at least 90% of the runs.

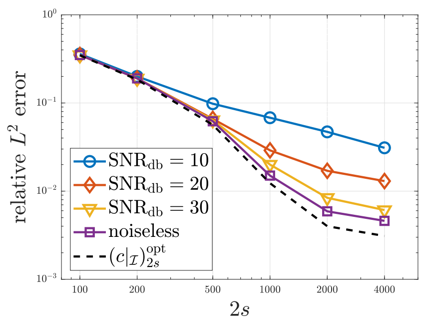

Fig. 5. Average relative L^2 error over 100 randomized phase encoding runs to approximate the 10-variate function $f(\mathbf{x}) := \prod_{\ell \in \{0,2,7\}} N_2(x_\ell) + \prod_{\ell \in \{1,4,5,9\}} N_4(x_\ell) + \prod_{\ell \in \{3,6,8\}} N_6(x_\ell)$, where $N_m : \mathbb{T} \to \mathbb{R}$ is the B-Spline of order m, $N_m(x) := C_m \sum_{k \in \mathbb{Z}} \mathrm{sinc}(\pi k/m)^m (-1)^k \mathrm{e}^{2\pi \mathrm{i} kx}$ with C_m chosen so that $\|N_m\|_{L^2} = 1$.

References

[1] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Simple and practical algorithm for sparse Fourier transform," in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 2012, pp. 1183–1194.

[2] M. A. Iwen, "Improved approximation guarantees for sublinear-time Fourier algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, pp. 57–82, 2013.

[3] M. Kapralov, *Sparse Fourier Transform in Any Constant Dimension with Nearly-Optimal Sample Complexity in Sublinear Time*. New York, NY, USA: Association for Computing Machinery, 2016, p. 264–277.

[4] L. Morotti, "Explicit universal sampling sets in finite vector spaces," *Applied and Computational Harmonic Analysis*, vol. 43, no. 2, pp. 354–369, 2017.

[5] D. Potts and T. Volkmer, "Sparse high-dimensional FFT based on rank-1 lattice sampling," *Applied and Computational Harmonic Analysis*, vol. 41, no. 3, pp. 713–748, 2016.

[6] B. Choi, A. Christlieb, and Y. Wang, "High-dimensional sparse Fourier algorithms," *Numerical Algorithms*, 2020.

[7] V. N. Temlyakov, "Reconstruction of periodic functions of several variables from the values at the nodes of number-theoretic nets," *Analysis Mathematica*, vol. 12, no. 4, pp. 287–305, 1986.

[8] L. Kämmerer, D. Potts, and T. Volkmer, "Approximation of multivariate periodic functions by trigonometric polynomials based on rank-1 lattice sampling," *Journal of Complexity*, vol. 31, no. 4, pp. 543–576, 2015.

[9] F. Y. Kuo, G. Migliorati, F. Nobile, and D. Nuyens, "Function integration, reconstruction and approximation using rank-1 lattices," *ArXiv e-prints*, 2020, arXiv:1908.01178.

[10] C. Gross, M. Iwen, L. Kämmerer, and T. Volkmer, "Sparse fourier transforms on rank-1 lattices for the rapid and low-memory approximation of functions of many variables," 2020, arXiv:2012.09889 [math.NA].